Madrid Mobile Configuration and Navigation

Last updated: July 19, 2019
Some examples and graphics depicted herein are provided for illustration only. No real association or connection to ServiceNow products or services is intended or should be inferred.

If you have comments about this documentation, submit your feedback to:
docfeedback@servicenow.com
# Table of Contents

## Mobile navigation and configuration

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessing an instance on a mobile device</td>
<td>5</td>
</tr>
<tr>
<td>ServiceNow Agent mobile app UI</td>
<td>5</td>
</tr>
<tr>
<td>ServiceNow Classic mobile app UI</td>
<td>9</td>
</tr>
<tr>
<td>Mobile web UI</td>
<td>11</td>
</tr>
<tr>
<td>Tablet interface</td>
<td>12</td>
</tr>
<tr>
<td>Legacy mobile UI</td>
<td>13</td>
</tr>
<tr>
<td>Getting started with the mobile app</td>
<td>14</td>
</tr>
<tr>
<td>Device security for ServiceNow Mobile</td>
<td>20</td>
</tr>
<tr>
<td>Log in to an instance with the mobile app</td>
<td>28</td>
</tr>
<tr>
<td>Add comments and work notes to your records with your mobile app</td>
<td>28</td>
</tr>
<tr>
<td>Select favorite applications with the mobile app</td>
<td>30</td>
</tr>
<tr>
<td>Filtering your records in the mobile app</td>
<td>31</td>
</tr>
<tr>
<td>Manage notifications from your mobile device</td>
<td>35</td>
</tr>
<tr>
<td>Track your location from your mobile device</td>
<td>36</td>
</tr>
<tr>
<td>Updating records without an internet connection</td>
<td>37</td>
</tr>
<tr>
<td>Scan QR codes and barcodes from your mobile device</td>
<td>45</td>
</tr>
<tr>
<td>Previewing screens from your Apple iOS mobile device</td>
<td>46</td>
</tr>
<tr>
<td>Identifying icons in the mobile app</td>
<td>48</td>
</tr>
<tr>
<td>Comparing how the mobile app works with iOS and Android devices</td>
<td>50</td>
</tr>
<tr>
<td>ServiceNow mobile app configuration</td>
<td>56</td>
</tr>
<tr>
<td>Components of the mobile app</td>
<td>59</td>
</tr>
<tr>
<td>Create an application for a mobile device in Studio</td>
<td>62</td>
</tr>
<tr>
<td>Create a data item for a mobile application</td>
<td>64</td>
</tr>
<tr>
<td>Create an applet</td>
<td>68</td>
</tr>
<tr>
<td>Functions</td>
<td>91</td>
</tr>
<tr>
<td>Mobile parameter tutorials</td>
<td>116</td>
</tr>
<tr>
<td>Screen UI policies for the mobile app</td>
<td>127</td>
</tr>
<tr>
<td>Mobile UI styles</td>
<td>131</td>
</tr>
<tr>
<td>Mobile push notifications</td>
<td>133</td>
</tr>
<tr>
<td>Mobile offline mode</td>
<td>138</td>
</tr>
<tr>
<td>Mobile authentication</td>
<td>143</td>
</tr>
<tr>
<td>Mobile geolocation tracking configuration</td>
<td>145</td>
</tr>
<tr>
<td>Additional mobile app configuration options</td>
<td>145</td>
</tr>
<tr>
<td>Mobile app configuration in the platform</td>
<td>154</td>
</tr>
<tr>
<td>Roles and permissions for the mobile app</td>
<td>183</td>
</tr>
<tr>
<td>Localization on mobile devices</td>
<td>183</td>
</tr>
<tr>
<td>Domain separation in the Mobile application</td>
<td>184</td>
</tr>
<tr>
<td>Migrate from classic mobile to ServiceNow mobile</td>
<td>184</td>
</tr>
<tr>
<td>Supported devices for the ServiceNow Agent mobile app</td>
<td>186</td>
</tr>
<tr>
<td>Support for ServiceNow Classic</td>
<td>186</td>
</tr>
<tr>
<td>Enterprise mobility management (EMM)</td>
<td>188</td>
</tr>
<tr>
<td>AirWatch Mobile Device Management (MDM)</td>
<td>188</td>
</tr>
<tr>
<td>BlackBerry Unified Endpoint Management (UEM)</td>
<td>193</td>
</tr>
<tr>
<td>Citrix XenMobile</td>
<td>197</td>
</tr>
<tr>
<td>IBM Maas360 mobile device management (MDM)</td>
<td>198</td>
</tr>
<tr>
<td>Intune mobile device management (MDM)</td>
<td>200</td>
</tr>
<tr>
<td>Jamf Pro mobile device management (MDM)</td>
<td>204</td>
</tr>
<tr>
<td>MobileIron Mobile Device Management (MDM)</td>
<td>205</td>
</tr>
<tr>
<td>Approve the ServiceNow app for Google Play</td>
<td>209</td>
</tr>
</tbody>
</table>
Mobile navigation and configuration

Manage incidents, collaborate with your team, respond to approval requests, access the knowledge base, and receive push notifications all on the go with your mobile device.

Depending on your device, go to the iTunes Store or the Google Play store and search for ServiceNow to download the native mobile app.

Explore
- Upgrade to Madrid
- Domain separation in the Mobile application

Set up
- ServiceNow mobile app configuration
- Mobile app configuration in the platform

Use
- Supported devices for the ServiceNow Agent mobile app
- Getting started with the mobile app

Develop
- Developer training
- Developer documentation

Troubleshoot and get help
- Ask or answer questions in the community
- Search the HI knowledge base for known error articles
- Contact ServiceNow Support

Accessing an instance on a mobile device

Access an instance from anywhere using your mobile device. Connect using the mobile app or a web browser on a mobile device.

The updated mobile UI includes the new ServiceNow branding as well as an all new app.

The mobile browser experience is consistent with the ServiceNow Classic mobile app.

ServiceNow Agent mobile app UI

Access an instance from the new mobile app.

You need to be on an instance using Madrid or later to use the new mobile app.

In the new mobile app, you can access different applications from the mobile app homepage.
Applications in the mobile app
Each application contains folders, which contain applets. Applets represent different options within the app. For example, an applet can be a filtered list, a calendar, or a map.
Applets in the mobile app
ServiceNow Classic mobile app UI

The classic mobile app is similar for both Android and iOS 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.

**Note:** If your company uses multi factor authentication (MFA), when you sign in on the classic mobile app, append the MFA code to your password.
ServiceNow Madrid Mobile navigation and configuration

iOS app UI

Android app UI
Not supported

- Custom app configuration parameters
- Custom buttons
- Deep linking to the mobile app
- Formatters
- Form Templates
- Internal distribution
- Knowledge v2
- Connect Support
- UI Scripts
- UI macro variables in the mobile Service Catalog

Limited support

- List filtering: Several of the following fields 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 will not be able to edit any of the limited fields.
  - between
  - dates
  - tags
  - related fields

- Visualizations are specific to a device. If you set up a visualization in the app on your iPhone, then view the app on an iPad, the visualization may not be the same.

Mobile web UI

Access an instance using the browser on your mobile device.

The mobile web UI is similar to the ServiceNow Classic mobile app. Some features are optimized more for the ServiceNow Classic app and may not perform as well on a mobile browser.

Administrators can configure home pages by role so different users may see different mobile experiences depending on their role.

Mobile web unsupported features

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 mobile interface
- HTML fields
- CODE tags to render HTML in Journal Fields
- UI Scripts
- Legacy Chat
- Field styles
- Formatters
- Form Templates
- Timeline visualizations
Embedded lists
Assessments, surveys, and legacy surveys
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
Custom auto-complete scripts

Tablet interface
Use a tablet to access your instance either app or from a browser.

Use the native tablet app for an experience similar to the native mobile app. Applications or modules that you have favorited in a desktop instance appear as tiles on your mobile app homescreen.

The tablet web UI mimics the desktop experience in UI16. $tablet.do has been disabled in UI16 because the tablet interface is the same as the desktop.

Connect to an instance using the browser on your tablet for an experience similar to the standard desktop user interface.

Tablet features with limited support
- 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. Data presented by these tools is read-only when accessed through the tablet interface.
- Calendars: You can access calendar reports but scrolling around the calendar as you would on the desktop interface is not supported.
- Video and image attachment upload: Attach videos and images with both iOS 9 and Android. Other file type attachments are not allowed.

Unsupported tablet features
- Field watcher: Administrators must use the desktop version to access the Field Watcher.
- 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. 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.

- Slushbucket feature: Any lists, fields, or filters that use the slushbucket feature are unsupported on a tablet device. Slushbuckets are only supported in 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.

- Support 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.

Legacy mobile UI

Use the classic Dublin mobile web UI for an older ServiceNow mobile experience.

Administrators can revert to the classic legacy mobile web experience, by setting the glide.ui.m.helsinki_mobile_enabled property in the sys_properties table to false.
Getting started with the mobile app

With ServiceNow Agent, you can update records, coordinate with coworkers, track your location, and work without an internet connection — all from your mobile device. You can download
ServiceNow Agent for Apple iOS or Google Android from the iTunes Store or the Google Play store.

This 20-minute podcast provides an introduction to mobile capabilities, using native phone functionality, and using Mobile Studio and Virtual Agent.

Watch this three-minute video for an overview of Agent Mobile App capabilities.

**Application home page**

When you log in to your instance, you see the application home page where you can access any one of the applications that are available in your instance.
Folders and applets

Applications contain folders, and these folders contain applets. Each applet contains records that correspond to the applet function. The Open Incidents applet, for example, contains all active Incident records.
Details tab

When you open a record, you see the Details tab for that individual record. The Details tab displays information about the record.
**Activity Stream tab**

To view or post comments on an individual record, tap the speech bubble icon (💬) to access the **Activity Stream** tab.
Related Lists tab

To view records that are related to the record that you are currently viewing, tap the chain icon (🔗) to access the Related Lists tab.
Device security for ServiceNow Mobile

This document applies to the current ServiceNow app for iOS and Android for Madrid. This document may be subject to change for future mobile releases.

Components and architecture

The ServiceNow Agent app consists of the ServiceNow server instance and the native apps for iOS and Android. The app uses fully native code and is not a hybrid approach. The mobile client applications communicate over a wireless connection with the server and pull live data for the end user.
Component explanations

App for iOS

The ServiceNow app for iOS is a fully native iPhone and iPad application. The application can be pulled, dynamically configured, and distributed using MDM (more information available in EMM section). ServiceNow does not currently distribute the ipa file to customers.

App for Android

The ServiceNow app for Android is a fully native application for Android phone. It can be downloaded from the Google Play store directly by a user, or can be pulled, dynamically configured, and distributed using MDM (more information available in EMM section). ServiceNow does not currently distribute the apk file to customers.

Identity and access management

User authentication for ServiceNow Mobile

The mobile app supports platform authentication using OAuth 2.0. Authentication mechanisms include multi provider SSO, MFA, LDAP, Local DB, and Digest.

AppAuth authentication

The ServiceNow mobile app uses a new authentication methodology called AppAuth. AppAuth uses an external mobile browser to log the user in. The following steps detail login using AppAuth.

1. The client creates and records a secret named the code_verifier, and derives a transformed version \( t(code\_verifier) \) (referred to as the code_challenge). This code_challenge is sent in the OAuth 2.0 Authorization Request along with the transformation method \( t_m \).

2. The Authorization Endpoint responds as usual but records \( t(code\_verifier) \) and the transformation method.

3. The client then sends the authorization code in the Access Token Request as usual but includes the code_verifier secret generated in the previous steps.

4. The authorization server transforms code_verifier and compares it to \( t(code\_verifier) \) from the previous steps. Access is denied if they are not equal.
Abstract protocol flow

Multi Provider SSO

The multi providers SSO plugin (com.snc.integration.sso.multi.installer) provides SAML authentication support. The login process (AppAuth) uses this plugin to redirect the user to the IDP (SAML provider) login page when using SAML. For more information on configuring multi provider SSO, see [Single sign on for the ServiceNow Classic mobile app](#).

Multifactor authentication

Users can access the instance via Multifactor Authentication using the MFA plugin (com.snc.integration.multifactor.authentication). For more information on MFA configuration, see [mobile multifactor authentication](#).

LDAP

Use LDAP authentication to access using LDAP credentials. The user sees the same login page as the local login (DB based) but the back end to the LDAP server deletes the authentication. For more information on LDAP configuration, see [LDAP integration and authentication](#).

Local DB

The user name and password in the user record in the instance database.
Not officially supported

- Kerberos
- Certificate-based authentication (AppAuth’s external browser may solve for some certificate based mechanisms)

Storage/Keychain
When you sign in to the app on your mobile device, the app uses your credentials to negotiate an OAuth Token with the instance. The iOS Keychain stores the token and Android uses KeyStore. The keychain encryption is AES 128 in Galois/Counter Mode (GCM).

The mobile app never stores the user password.

The mobile app does store the Client ID which is necessary for getting the OAuth token as part of the authentication flow.

Access and Refresh Tokens
The mobile application uses access and refresh tokens to determine valid user sessions.

At first login, a user is given an access token and a refresh token. These tokens are valid for a configurable amount of time. When the user opens the mobile app, the client checks to see if the access token is valid. If valid, the user is able to continue with the session. If not valid, the client then checks if the refresh token is valid. If valid, the refresh token is used to fetch a new valid access token for the user, and the session can continue. If the refresh token is not valid, the user must reauthenticate.

User termination
When an administrator deletes or removes a user from the system, the Access Token is no longer valid and any operation logs the user out.

Mobile data flow
Data flow follows standard secure OAuth practices. A description of the flows can be found below.

Retrieval
The following describes how data is retrieved from the ServiceNow mobile app.

Read data
When a user requests to view information on the mobile app, the following steps occur.
1. The mobile app sends a request to access data from the instance. The request includes the token and any relevant data field needed for the request.
2. The instance receives the request and checks if the Token is valid.
3. If the token is valid, the request is directed to the relevant API to fetch the information.
4. The information is returned to the mobile app.

Downloading documents
When a user requests to download documents from the app, the following steps occur.
1. The mobile app sends a request to access the document. The request includes the Token.
2. The instance receives the request and checks if the Token is valid.
3. If valid, the document becomes available to view or take further actions on the device.

Write-backs
The following describes how data is written back from the ServiceNow mobile app.

Updating fields
When a user updates a field in the mobile app, the following steps occur.
1. The mobile app sends the Token and the action metadata, for example the ID, or the field to be updated, to the instance.
2. The instance directs the action based on the relevant API.
3. The instance completes the action and sends a response to the mobile app.
4. Based on the response, the mobile app reflects the field changes and action availability in the UI.

Attaching documents
When attaching documents, the following steps occur.
1. The mobile app asks the user to attach a document, for example, an image.
2. The mobile app sends the document and Token to the instance.
3. The instance places the document based on the relevant API.
4. The instance sends a response back to the mobile app.

Internal mobile app distribution
Internal distribution of the ServiceNow Mobile app is supported through all major EMM vendors.
Customers are able to pull the app for iOS or Android from the Apple App store and Google Play respectively, dynamically configure the apps to point to the correct ServiceNow instance, and distribute using the EMM hub. This way, the MDM can fully manage the app as part of a customer portfolio.

Note: ServiceNow does not currently distribute the ipa/apk files, or any other unpublished app to customers as it breaches the Apple Enterprise Developer License Agreement.

Mobile app distribution providers:
- AirWatch Mobile Device Management (MDM)
- BlackBerry Unified Endpoint Management (UEM)
- Intune mobile device management (MDM)
- IBM MaaS360 mobile device management (MDM)
- MobileIron Mobile Device Management (MDM)

Data security for ServiceNow Mobile
The ServiceNow Mobile app uses SSL/TLS for Over-the-Air communication encryption. The OAuth authorization endpoints are HTTPS.
Data at rest

Application preference data such as favorites, home screen, and the mobile navigator items are stored and cached locally on the device. The mobile app does not store record data such as incidents, problems, etc. on the device unless the organization has specifically enabled offline syncing for Field Services. The record data is encrypted with AES 128.

Stored Information

- Databases
  - User defined instances
  - Favorite application IDs
  - Push Notifications
  - Geolocation updates
  - Offline data
- Preferences
  - sys_id, display name, username, and initials of the current user
  - URL and name of the current instance
  - Last activity timestamp
  - Encrypted PIN code
  - Offline cache warning period
- Server Properties
  - LOCATION_PROXIMITY
  - IS_PIN_CODE_REQUIRED
  - IS_BLURRED_IN_BACKGROUND
  - IS_BLOCK_ATTACHMENT_SHARING
  - LOCATION_TRACKED
  - IS_CLEARING_CLIPBOARD_IN_BACKGROUND
  - IS_HIDE_APPLICATIONS_SCREEN_IMAGE
  - IS_OFFLINE_ENABLED
  - LOCATION_FREQUENCY
  - key_analytics_initial_app_launch flag
- Information stored in the system Account Manager
  - Login date
  - Instance URL
  - Access Token
  - Refresh Token

Data in motion

Data in motion is over a secure SSL/TLS channel and encrypted with HTTPS.
Offline access and data cache configuration

As an admin, you can choose specific Applets and Actions to be enabled offline from within the Studio. On the mobile device, in Settings your users can select offline and choose to “cache data”. The offline flows designated by the admin are downloaded and cached to the device. Offline cached data is encrypted using native encryption and expires at a specified period of time. The default is 48 hours or when user signs out of the mobile app.

Offline data is protected by local-auth and the app PIN that can be enforced.

Push notifications

Administrators create push notifications and users are able to receive them.

Cloud

For more information on the push notification system including process, configuration, and architecture, see Push notification system. Administrators can configure push notification delays using scheduled jobs. To view an example included with the base system, navigate to System Scheduler > Scheduled jobs, then search for a job with the name Push. 5 seconds is the minimum time allowed for the push delay.

Mobile security practices

Mobile security practices include mobile-specific system properties, attachment control, password reinforcement, security patching, and controlling shared data.

Security controls

Configure security controls to restrict copy/paste, enforce PIN, or block attachment functionality.

Restrict copy/paste

Copy/paste restrictions are defined in the system properties (sys_properties) table. There are two applicable security properties.

- glide.ui.sg.clear_pasteboard_when_backgrounded: Clears the copy/paste clipboard when the ServiceNow app enters the background. For more information on clearing the clipboard, see Configure clear clipboard

Require an app PIN

Require users to enter a six digit PIN each time they sign in from their mobile device. Requiring an app PIN is controlled by the glide.sg.require_mobile_application_pin system property. For more information on requiring an app PIN, see Require an app PIN for the mobile app.

Disabling attachments on a mobile device

Use an ACL to block specific access on mobile. Use the isMobile method to check if a request comes from a mobile device. For example, you could add an ACL for the attachment (sys_attachment) table where the read and write scripted ACLs includes the following check. You
can also add this code to any existing ACLs you have for the attachment table. If have multiple attachment ACLs, all of the need to have **Admin override** option unchecked.

```java
if( gs.isMobile() ){
    answer = false;
}
```

**Note:** You need elevated privileges to create ACLs.

## Enable the blur app option

Blur the mobile app when not in focus on a mobile device using the following system property in the system properties (sys_properties) table.

- `glide.ui.sg.blur_ui_when_backgrounded`

For more information, see [Configure the blur app option](#).

## Penetration testing

ServiceNow engages a third party to perform penetration testing of the mobile app. This typically happens annually but sometimes occurs more frequently. The results of these tests are available to customers on CORE. Customers can test the mobile application in conjunction with a pen test of their instance per the process outlined in [KB0538598](#).

**Security patching**

In the event a security patch is needed, the mobile development team aligns with standard SDLC properties in order to patch.

**User data collection**

The mobile app does not specifically collect any user data.

Any user transactions or usage within the app is tracked on the ServiceNow instance just as it is on the web. For user credentials, after a user logs in, the mobile app negotiates an OAuth Token that is stored in the Apple Keychain or the Android Keystore. User credentials are never saved. If the user opts in, the following information is collected:

- Location
- Access to camera
- Notifications

**Shared data**

The mobile app communicates with a third party software for app crash reporting. No customer information is shared.

- **iOS:**
  - Uses Crashlytics for crash reporting.
- **Android:**
  - Uses Hockey App for crash reporting.

**Incident reporting**

Mobile app issues should be reported through the standard support channels. You can report incidents by contacting ServiceNow support.
Log in to an instance with the mobile app

Read and edit records from any location by using the mobile app to log in to an instance.

Role required: none

If you downloaded the mobile app from an enterprise mobility management (EMM) suite, enter your user credentials as prompted. If you did not download the mobile app from an EMM suite, follow the subsequent steps.

1. Tap the mobile app (📲) on your device, and then tap the plus button (➕) to open an instance.
2. In the instance address field, enter the instance address in one of two ways:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual input</td>
<td>Type the instance address in the instance address field. Note: You do not need to include service-now.com at the end of the instance name.</td>
</tr>
<tr>
<td>Scan QR code</td>
<td>Tap the QR icon (🔍), and then scan the QR code that was provided by your administrator.</td>
</tr>
</tbody>
</table>

3. Tap Login.
4. In the window that appears to remind you that you are connecting your mobile device to the server, tap Continue.
5. When you see the prompt to connect to the server, tap Allow.

Add comments and work notes to your records with your mobile app

Submit work notes and comments to a record with your mobile app by posting on the Activity Stream for that record.

Role required: none

1. Navigate to the Activity Stream tab of a record, and tap the Activity Stream icon (💬).
2. To post to the Activity Stream, tap the plus icon () in the record title bar.

You can post any of the following items to the Activity Stream:

- Work notes
- Additional comments
- Photos
- Videos
- Files

**Note:** A work note appears to employees only, but an additional comment appears to employees and customers.
Select favorite applications with the mobile app

Mark an application as a favorite on the application home page so that you can quickly find it later.

Role required: none

1. Navigate to the application home page.
2. Mark applications as favorites by tapping the star icons (⭐) next to each application.
3. To hide unmarked applications, tap the star icon at the top, right corner of the screen.
   Tap the star icon again to show all applications on the application home page.
Filtering your records in the mobile app

Filter screens to display only the records that are most relevant to you.
When you tap the filter icon ( filtro icon), you can filter your records. Filtering records in the mobile app works like filtering with a condition builder on the Now Platform.

The intersection Venn diagram ( diseño Venn) enables you to create AND conditions between different fields. The following filter, for example, queries your list for all Incident records in which the state is In Progress and the caller is Beth Anglin.

```
state = 'In Progress' AND caller = 'Beth Anglin'
```

The equivalent condition builder statement appears as follows.
The union Venn diagram enables you to create OR conditions between different fields. For example, if your filter looks as follows, the list displays Incident records in which the state is Resolved or in which the caller is Fred Luddy.
The equivalent condition builder statement appears as follows.

Similar to when you select **Run** in the condition builder, tap **Done** in the mobile filter to apply your query conditions to the list of records.
To remove all filter conditions, tap Clear.

Manage notifications from your mobile device

Receive push notifications for the records that are most relevant to you by changing the notification settings on your mobile device and in the mobile app.

Role required: none

Push notifications appear on your device lock screen when they first arrive. You can tap on a push notification to open the corresponding record. A badge also appears on the mobile app icon when you receive notifications.

Push notifications are sent to you when:

- A record is assigned to you.
- A record is assigned to your group.
- A record that you are assigned to or are following is changed.

**Note:** The notifications setting on your mobile device takes precedence over the notifications setting on the mobile app. If you enable notifications in the mobile app but do not allow push notifications on your mobile device, you cannot receive push notifications.

1. Navigate to the **Notifications** tab by tapping the bell icon (🔔).
2. Tap the **Enable Notifications** switch to enable or disable push notifications.

3. Clear notifications in one of the following ways:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear individual notifications</td>
<td>Swipe left on the individual notifications and tap <strong>Clear</strong>.</td>
</tr>
<tr>
<td>Clear all notifications</td>
<td>Tap <strong>Clear All</strong>.</td>
</tr>
</tbody>
</table>

Track your location from your mobile device

Turn on location tracking from your mobile device and in the mobile app so that you can keep a record of where you were when you last worked on an instance. Location tracking continues even when there is no internet connection.

Role required: none
The location setting on your mobile device takes precedence over the location setting in the mobile app. To enable the mobile app to track your location, you must first grant permission on your mobile device.

1. Navigate to the Settings tab by tapping the gear icon (⚙️), and then open Location Tracking.
2. Tap the Location Tracking switch.
3. In the Track For field, select how long you want the mobile app to monitor your location.
4. View the expiration time for location tracking on the Settings tab.

5. To end location tracking before the expiration time, navigate to Settings > Location Tracking, and then tap the Location Tracking switch.

Updating records without an internet connection

Access and submit actions to records even if you do not have an internet connection.
You have to plan ahead when you use offline mode. If you know that you will be working in an area with no internet access, download what you want to work on ahead of time while you are still connected to the internet.

When you are in offline mode, the changes that you make to your records are logged in your outbox. Your outbox tracks all the actions that you made on your cached records. After your device has internet access, you can sync your device with the instance. Your changes update to the instance and your local cache reflects the changes that you made to the instance when you were offline.

**Enable offline mode**

To enable offline mode, tap the gear icon (⚙️) to access the **Settings** tab. Open **Offline Mode** and then toggle on **Offline Mode**.

If you have not already downloaded the offline cache, you see a dialog box that asks you to download it. Tap **Download and Go Offline**.
Navigate the mobile app in offline mode

When you are in offline mode, a banner that reads "Offline Mode" appears across the top of all screens.
Depending on how your administrator configures the mobile app, you are unable to submit certain actions while you are in offline mode. These actions are grayed out on the user interface.
When you submit an action while you are in offline mode, the change gets marked with a yellow border. Changes remain marked until your device syncs to the server.
Disable offline mode and sync outbox

To return online, in the mobile app, navigate to Settings > Offline Mode. On the offline mode screen, toggle off Offline Mode.
A dialog box asks you to sync the changes in your outbox. To push all the changes that you made in offline mode to the instance, tap **Go Online & Sync**. After the sync completes, you are back online and your offline cache is deleted.

**Cache expiration**

Your administrator configures a default length of time after which your offline cache expires. When a cache expires, you lose all the data that you saved to the cache. If you do not sync the cache to an instance before the cache expires, none of your changes show on the instance. Warning messages appear periodically to remind you to sync your cache before it expires. To avoid losing your data due to a cache expiration, always sync your cache before and after going offline.
Resolve sync errors

Problematic changes that you made in offline mode do not sync to the instance. They remain in the outbox until they are resolved.

You cannot sync changes that contradict changes that were made by other users while you were offline. For example, you may receive an error message if you try to sync changes to a record that another user closed while you were working in offline mode.

To view the errors in your cached changes, navigate to **Settings > Offline Mode > Outbox**. Error messages indicate where errors occurred in your cached records while you were offline. From the application home page, navigate to the indicated records and resolve these errors.
After you resolve an error, return to the outbox to delete the error messages. In the outbox, swipe left on the error message and then tap **Delete**.

**Scan QR codes and barcodes from your mobile device**

Log in to an instance or keep track of your assets with a quick scan from your mobile device.

Role required: admin

1. Tap the scan icon (.scanicon) to open your device camera.
2. Tap the bolt icon (.flashicon) to enable or disable flash.
3. Point your device camera toward a QR code or barcode.

The camera automatically takes a photo when a QR code or barcode appears on the screen. A box surrounds the selected code image and the code text displays near the top of the screen.
4. To manually edit the code text, tap the text that appears near the top of the screen.

5. To retake the photo, tap the redo icon (↺).

Previewing screens from your Apple iOS mobile device

Use Apple 3D Touch capabilities to quickly navigate to screens and finish your work. 3D Touch is supported for any device with Apple iOS 9 or later. 3D Touch is not available on Apple iPads or Google Android mobile devices.

Enable 3D Touch on your iPhone

To use the preview feature on the mobile app, you must have 3D Touch enabled on your device.
To enable 3D Touch, navigate to the Settings app on your iOS device and then select **Accessibility**. On the Accessibility screen, toggle on **3D Touch**.

**Preview an item in the mobile app**

To preview a screen, press firmly on the item that you want to look at. A window appears providing you with a preview of the next screen.

Let go to stop previewing.

You can also access your favorite applications from your device home screen by using 3D Touch.
To open a favorite from your home screen, tap and hold the mobile app icon (NOW), and then tap one of the application names that appear. For more information on how to designate applications as favorites, go to Select favorite applications with the mobile app.

**Identifying icons in the mobile app**

Identify the correct icon to use in the mobile app so that you can complete your tasks in the instance.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Corresponding action</th>
</tr>
</thead>
<tbody>
<tr>
<td>🏡</td>
<td>Access the application home page.</td>
</tr>
<tr>
<td>🔔</td>
<td>Access the Notifications tab.</td>
</tr>
<tr>
<td>Icon</td>
<td>Corresponding action</td>
</tr>
<tr>
<td>------</td>
<td>----------------------</td>
</tr>
<tr>
<td><img src="icon1.png" alt="Icon" /></td>
<td>Access the <strong>Settings</strong> tab.</td>
</tr>
<tr>
<td><img src="icon2.png" alt="Icon" /></td>
<td>Add an instance, or add a post to the Activity Stream.</td>
</tr>
<tr>
<td><img src="icon3.png" alt="Icon" /></td>
<td>Scan a QR code or barcode.</td>
</tr>
<tr>
<td><img src="icon4.png" alt="Icon" /></td>
<td>Turn on or off flash.</td>
</tr>
<tr>
<td><img src="icon5.png" alt="Icon" /></td>
<td>Retake a photo.</td>
</tr>
<tr>
<td><img src="icon6.png" alt="Icon" /></td>
<td>Show information about a corresponding item.</td>
</tr>
<tr>
<td><img src="icon7.png" alt="Icon" /></td>
<td>Select and filter favorite applications on the application home page.</td>
</tr>
<tr>
<td><img src="icon8.png" alt="Icon" /></td>
<td>Submit an action to a list or record.</td>
</tr>
<tr>
<td><img src="icon9.png" alt="Icon" /></td>
<td>Access record details.</td>
</tr>
<tr>
<td><img src="icon10.png" alt="Icon" /></td>
<td>Access the <strong>Activity Stream</strong> tab for a record.</td>
</tr>
<tr>
<td><img src="icon11.png" alt="Icon" /></td>
<td>Access the <strong>Related Lists</strong> tab for a record.</td>
</tr>
<tr>
<td><img src="icon12.png" alt="Icon" /></td>
<td>Apply a filter to a list of records.</td>
</tr>
<tr>
<td><img src="icon13.png" alt="Icon" /></td>
<td>Apply an AND condition statement to a list of records.</td>
</tr>
<tr>
<td><img src="icon14.png" alt="Icon" /></td>
<td>Apply an OR condition statement to a list of records.</td>
</tr>
<tr>
<td><img src="icon15.png" alt="Icon" /></td>
<td>Return to the current day on the calendar screen.</td>
</tr>
<tr>
<td><img src="icon16.png" alt="Icon" /></td>
<td>Select a date.</td>
</tr>
<tr>
<td><img src="icon17.png" alt="Icon" /></td>
<td>Call or send a text message.</td>
</tr>
<tr>
<td><img src="icon18.png" alt="Icon" /></td>
<td>Find directions to a GPS location.</td>
</tr>
<tr>
<td><img src="icon19.png" alt="Icon" /></td>
<td>Open the record listed above the current record.</td>
</tr>
<tr>
<td><img src="icon20.png" alt="Icon" /></td>
<td>Open the record listed below the current record.</td>
</tr>
<tr>
<td><img src="icon21.png" alt="Icon" /></td>
<td>Navigate to the previous screen.</td>
</tr>
</tbody>
</table>
Comparing how the mobile app works with iOS and Android devices

Whether you downloaded the mobile app for Apple iOS or Google Android, ServiceNow Agent caters to your mobile device.

**Activating system properties in the app**

In the mobile app, activate the system properties that you add or edit in the Now Platform.

<table>
<thead>
<tr>
<th>App</th>
<th>Activation method</th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS</td>
<td>Log out of the instance from your mobile app and then log in again.</td>
</tr>
<tr>
<td>Android</td>
<td>Go to the applications home page and swipe down to refresh. You do not need to log out of the instance.</td>
</tr>
</tbody>
</table>

**Navigating through records**

The iOS version of the mobile app allows you to open other records from the details screen of one record. The Android version does not include this function.
<table>
<thead>
<tr>
<th>App</th>
<th>Navigation method</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS</td>
<td>Tap the up and down arrows on the details screen to open adjacent records on the same list.</td>
<td></td>
</tr>
</tbody>
</table>
Blurring the app when not in focus

If you configure the mobile app to *blur the screen when backgrounded*, you can expect the configuration to behave differently between iOS and Android.

<table>
<thead>
<tr>
<th>App</th>
<th>Blur behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS</td>
<td>When you move the mobile app to the background, the screen is blurred.</td>
</tr>
<tr>
<td>Android</td>
<td>When you move the mobile app to the background, the screen is white and screenshots are disabled.</td>
</tr>
</tbody>
</table>
Disabling the app from pasting text

If you configure the mobile app to clear the pasteboard when you navigate out of the mobile app, you can expect the configuration to behave differently between iOS and Android.

<table>
<thead>
<tr>
<th>App</th>
<th>Paste behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS</td>
<td>The pasteboard is cleaned when you navigate out of the mobile app.</td>
</tr>
<tr>
<td>Android</td>
<td>Copying text in the mobile app pastes the string &quot;Removed&quot; outside the app.</td>
</tr>
</tbody>
</table>

Using the internal web browser

When you open a link with an internal browser, the actions that you can perform in the internal browser differ between iOS and Android.
<table>
<thead>
<tr>
<th>App</th>
<th>Browser features</th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS</td>
<td>Click <code>&lt;</code> or <code>&gt;</code> to navigate pages and click <code>X</code> to close the internal browser.</td>
</tr>
</tbody>
</table>
Using 3D Touch

3D Touch is supported for any device with Apple iOS 9 or later. 3D Touch is not available on Apple iPads or Google Android mobile devices. For more information on 3D Touch in the mobile app, see [Previewing screens from your Apple iOS mobile device](#).
### ServiceNow mobile app configuration

The ServiceNow mobile app is built with a mobile-first design. You can configure the ServiceNow mobile app so your users can access an instance on a tablet or smartphone.

<table>
<thead>
<tr>
<th>App</th>
<th>Preview method</th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS</td>
<td>To preview a screen, press firmly on the item that you want to look at.</td>
</tr>
<tr>
<td>Android</td>
<td>3D Touch is not available.</td>
</tr>
</tbody>
</table>
Requirements

Configuration for the mobile application takes place in Studio. The mobile web experience is consistent with the ServiceNow Classic app. Use the native app to test your configuration. Depending on your device, go to the iTunes store or the Google Play store and search for ServiceNow to download the native mobile app.

You must be on Madrid or later to configure the new mobile app.

Use the following instructions to get started on building a mobile app.

What to do

The mobile app is built on a hierarchy. You can have several applications for the mobile app and limit user access by role so only some users can see each application. Each application contains folders to help separate applets. For example, you could have an ITSM application, which contains folders for My Incidents and My Group Incidents.

Applets are miniature applications that contain various screens with more information. Data items determine the information that appears in each applet. For example, an applet for My High Priority Incidents would have a data item that pulls information from the system to create a list of incidents that are only assigned to that user and have a priority of Critical or High.

Functions in an applet determine how the user interacts with the app. You can add functions that navigate to another screen, open the phone option on your mobile device, or edit a record.
Role required: admin

1. **Create an application in Studio.**
   Applications are the high-level containers for information in the mobile app. For example, you could create an ITSM application that contained applets specific to an ITSM user. Applications appear on the main app page and you can limit user access by role.

2. **Create a data item.**
   Data items represent blocks of information from a table in the platform. Configure data items to determine what information appears in the mobile app. Each data item gets associated with an applet.

3. **Create an applet.**
   Applets are smaller applications that contain specific information and screens. For example, for an ITSM application, you could have an applet for Open Incidents. Depending on the
information you want to display, you can choose from different screen templates, such as a list, a map, or a URL page.

4. **Add functions to the mobile app.**

Functions determine how the user navigates through the app. Configure functions that allow users to navigate to a new screen, make a call or send an email from a field, or make changes to a record.

**Next steps**

Configure push notifications, offline support, and location tracking.

**Components of the mobile app**

The following list of terms provides overview information on mobile-specific terminology.

<table>
<thead>
<tr>
<th>Glossary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Term</strong></td>
</tr>
<tr>
<td>Action function</td>
</tr>
<tr>
<td>Action item</td>
</tr>
<tr>
<td>Activity stream screen</td>
</tr>
<tr>
<td>Applet</td>
</tr>
<tr>
<td>Application</td>
</tr>
<tr>
<td>Term</td>
</tr>
<tr>
<td>----------------------</td>
</tr>
</tbody>
</table>
| Functions            | Functions are activities that users can do in the app. They adhere to include UI conditions and can be state conditional. There are three types of functions:  
  · Action function  
  · Navigation functions  
  · Smart Button functions |
<p>| Data item            | A data item is a dataset correlated with a table in an instance. Associate data items with applets so that the applets can transform the dataset into human-readable information. |
| Detail view          | The Detail view is the drill down from the first layer list, map, or other template. It is presented as a list and the fields are configurable.        |
| Folders              | Optionally group applets into folders. You can always adjust the order of applets in each folder, but the folders will be displayed alphabetically in the app. |
| Item view            | The item view determines the way that fields appear in the app. When you select an applet preview for an app, you are actually selecting the item view. Item views use JSON to determine how fields are configured in the app. |
| Navigation function  | Navigation functions are a type of button that allows users to move from one screen to another. For example, a user can tap a field, such as a user name, on a form and move to another form that contains more information for that field, such as the user profile. Navigation functions can be configured with or without parameters. |
| List view (master)   | The list view is first layer or level of the data visualization. This can be a card view, a map, a calendar or any supported template. The list view components can be configured by the Studio user. |
| Map applet           | Map applets can be used to display configurable pins on a map with phone, GPS navigation, and additional details functionality. The data item configured for a map applet must have an address or GPS coordinates for map to work. |
| Studio               | The mobile Studio is the mobile builder where ServiceNow admins can configure mobile applications for users.                                  |</p>
<table>
<thead>
<tr>
<th><strong>Term</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Offline</td>
<td>App creators/admins will designate specific flows for users that will be available to “download” for offline. Users will have to enable these flows to be offline while they have connectivity. Actions submitted while offline will be sent to an Outbox queue that will be resynced after connectivity is reestablished. Not all online functionality will be available offline. Going offline is intended for users who spend minimal time offline, it is not optimized for long periods of time offline.</td>
</tr>
<tr>
<td>Parameters</td>
<td>Use parameters to create a variable or a placeholder that waits for input from either the user or the database. Create parameters for data items, action functions, or navigation functions. For tutorials on how to use parameters, see Mobile parameter tutorials.</td>
</tr>
<tr>
<td>Carried parameters</td>
<td>A carried parameter is a parameter that gets carried through more than one screen.</td>
</tr>
<tr>
<td>Related list screen</td>
<td>Related lists are one of the screen options you can apply to certain applets. Like records, they display lists that apply to specific records. You can configure related lists to navigate to new screens using a navigation function.</td>
</tr>
</tbody>
</table>
| Screen | Screens are different types of UIs the user is presented with in the app. Applets are made up of several screens, and the screens that are available depend on the applet. Screens have configurable components, conditional formatting, sorting order, and filters. Current available screens:  
  - List (Master)  
  - Details  
  - Activity Stream  
  - Related Lists  
  - Map  
  - URL Template  
  - Calendar  
  - Grouped list |
| Smart Button function | Functions that can be smart buttons such as Email, Phone (call/text), Map Directions, and URL. Smart button functions open a native application on your phone, such as a map, or your phone keypad. |
| UI Policy | UI Policies manage the dynamic visibility of parameters. Changing fields or parameters will adhere to governed UI policies that have been set up in the instance. |
Create an application for a mobile device in Studio

While you can create a mobile application in the ServiceNow platform, most application configuration takes place in the Studio. Use the steps in this topic to set up the Studio and to get you started on creating a mobile application.

The Studio is an IDE-like interface that allows application developers to work on custom applications in one centralized location. All configuration for the mobile application in this section take place in Studio (System Applications > Studio). For more general information on using Studio, see ServiceNow Studio.

Role required: admin

Watch this one-minute video for an overview of mobile app creation.

1. Navigate to System Applications > Studio.
2. In the Studio, click Create Application or select an existing application from the list.
   The application you create here is a new scoped application for you to build your app in. Scoped applications help restrict data and application files to just this one application. For more information on scoped applications, see Application scope.
3. In the Application, click Create Application File.
4. From the Create Application File window, click Mobile Development, then click Create.
5. In the Name field, create a name for the application.
   The application you create here is the mobile app. The name you add is the name of the mobile app and appears on the home screen of the mobile app. For example, ITSM.
6. In the Description field, add information that will help you identify the application later.
   For example, if you had several ITSM applications, you could use the short description to differentiate.
   Users can see the description you add in the mobile app by tapping the information icon.
7. Click Save.
8. To deactivate the application, clear the Active toggle switch.

9. To limit user access the application in the mobile app, click the user icon ( ).
   a) In the permissions window, use the slushbucket to add roles that can see the application in the mobile app.
   b) Click **Save**.

To test any changes that you make in the mobile app, return to the home screen with the list of applications and refresh the screen.

Create a data item for a mobile application

Data items are data sets that represent a table from the platform. Data items determine the information that appears on a screen. Associate data items with specific applets to determine the purpose of each applet.

Role required: admin

1. In Studio, navigate to **Mobile Development > Data Items**, and click the popout icon ( ).
2. In the Data Items tab, click **Create a new data item**.
3. Complete the following fields as needed.

### Data item fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A title for the data item. You can have multiple data items with the same name. Make sure that this name is unique so that you can find the screen you are looking for when associating it with a folder.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional additional details about the data item</td>
</tr>
<tr>
<td>Table</td>
<td>The name of the table you want the data item to pull information from.</td>
</tr>
<tr>
<td>Condition type</td>
<td>Determine whether the condition for the data item is declarative or scripted. Use a declarative condition to create conditions for the data item using the condition builder. Use a scripted conditions to create a scripted data item.</td>
</tr>
<tr>
<td>Add sort</td>
<td>Adds fields to the form that allow you to configure how to sort the list. In the Add Sort window, select the field you want the list to be sorted by. For example, Caller. Then select a to z or z to a to determine which order the list goes in.</td>
</tr>
<tr>
<td>Query Condition</td>
<td>Set of conditions for the data item to conform to. You can create conditions using the condition builder.</td>
</tr>
</tbody>
</table>
ServiceNow Madrid Mobile navigation and configuration

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sorted by</td>
<td>This field only appears if you click Add sort. It is automatically populated with the information you added in the Add Sort window.</td>
</tr>
<tr>
<td>Direction</td>
<td>This field only appears if you click Add sort. It is automatically populated with the information you added in the Add Sort window.</td>
</tr>
</tbody>
</table>

4. Optional: If you want the application to query the database for more information before loading the data, add parameters to the data item.

5. Optional: To configure a data item for a Grouped By applet, in the Group Function section, from the Group By list, select the field you want to group items by. The available fields are based on the table you selected.

6. Click Save.

Associate a data item with an applet.

Configure a data item with parameters

Data items are data sets from a table in the platform. Configure a parameterized data item to filter and view just the relevant data according to the selected parameters.

Role required: admin

Use the included examples to create a data item that allows users to open an incident list filtered by priority. For more detailed tutorial on how to create a data item with parameters, see Tutorial: Configure a data item with parameters.

1. In Studio, in your mobile app, navigate to Mobile Development > Data Items.
2. Click Create a new data item or open an existing data item you want to add parameters to.
3. Complete the Name, table, and condition fields as needed.
   For more information on creating a data item, see Create a data item for a mobile application.

   For example, create a data item for open incidents.

4. Click Save.

5. In the Parameter Definition section, click the add icon (+).

6. In the Item Parameter window, in the Name field, type a name for the parameter. Parameter names correlate most often with fields on a form.
   For example, type Priority as the parameter name.

7. From the Type list, select the type of parameter. The type determines how the user interacts with the mobile UI. For example, a type of Decimal or Integer tells the mobile device to open a numbers-only keypad. Select from the following types.
   - String: Uses a full keyboard for input. Use the String type for list parameters, such as priority or state, or for reference fields, such as assigned to or caller.
   - Integer: Opens a numbers-only keypad
   - Decimal: Opens a numbers-only keypad
   - Boolean: Opens a true or false selection option
   - DateTime: Opens a calendar with an exact time selector
   - Date: Opens a calendar
8. Click **Save**.

9. In the data item form, in the Query condition section, add a query condition for your parameter. The condition field should match the parameter you are querying the database for. For example, if you are creating a data item to query the Priority field, create a condition for **Priority is <priority>**.
Open Incidents

Data Item

**Properties**

- **Name**: Open incidents
- **Table**: Incident

**Description**

Query condition:

- **Active**: is true
- **State**: is one of: New, In Progress, On Hold, Resolved
- **Priority**: is string

**Parameter Definition**

**Parameters**

- **Name**: Priority
- **Type**: string
10. Click **Save**.

After you create a data item, you can assign it to an applet. Data items with parameters require additional configuration in the applet. For more information on assigning a data item with parameters to an applet, see *Assign a data item with parameters to an applet*.

Create an applet

Applets define the template for screens that appear in the mobile app. Create an applet in the application to contain a specific type of screen, for example, a list or a map.

Applets are templates for a series of screens. Depending on the template you select, different options are available. You can also configure individual screens in the platform, but the platform does not have the precautions that Studio has to prevent you from creating something that breaks the app. The following steps use Studio for configuration.

Role required: admin

1. In **Studio**, navigate to the application you want to create an applet for, then click the pop-out icon ( ) that appears when you point to the folder name.
2. Optional: Create a folder to contain the applet.
   a) On the application page, click **Create folder**.
   b) In the blank field that appears, type a name for the folder. For example, ITSM. The folder name appears at the top of a group of applets.
   c) Click **Save**.
3. In the applet list, click **Create new applet**.
4. In the Applet name field, type a name for the applet. The applet name appears on a tile in the application on a mobile device.
5. From the list of icons, select an icon to appear in the applet.
6. Optional: In the Description field, type a description for the applet.
7. Select an applet template. Depending on the applet you select, different screen configuration options appear.
8. From the screen preview section, select or clear the check boxes next to the screens you want to appear as part of your applet. For example, include an activity stream and related list screens as part of your applet by selecting those check boxes. Different screens are available depending on the applet template you choose.
9. Click **New**.
10. Optional: If you create folders for your application, you can move the applets to different folders by dragging them into that section.

Configuration for the applet varies depending on the template you select. For more information on how to configure a specific applet type, navigate to the topic specific to that applet.

### Assign a data item with parameters to an applet

When you associate a parametrized data item with a screen, additional fields appear in the screen configuration that you must complete for the parameter to apply.

You should have already configured a data item that has parameters.

Role required: admin

1. In Studio, in your mobile application, navigate to **Mobile Development > Applications > [Name of your application]**.
2. Click **Create a new applet** or open an existing applet.
3. In the Data Item field, select the parameterized data item you created.
4. In the Parameter setting section that appears, under **UI Parameters > User Input**, click the row that contains the parameter label and type.
5. In the User Input Parameter Definition window, complete the following fields as needed. Some fields only appear when you select a specific input type.
   - Use the fields on this form to determine how the user interacts with the UI.
User Input Parameter Definition fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for the UI parameter. You can have multiple UI parameters with the same name so make sure you choose something you can easily discover later. The name you select appears in the mobile app. For this example, use Priority as the name.</td>
</tr>
<tr>
<td>Input type</td>
<td>The UI used to complete the variable. For example, if you have a parameter for the Assigned to field, select List so that users have a list of users in that field to search for. Choose from one of the following options.</td>
</tr>
<tr>
<td></td>
<td>- Text: Provides a simple text field. This option works best for fields that require free text, such as work notes or resolution details. Text is the default type.</td>
</tr>
<tr>
<td></td>
<td>- List: Opens a list for the user to select from. This option works well for reference fields that require specific information.</td>
</tr>
<tr>
<td></td>
<td>- SearchList: Provides a search bar so that users can search in a list.</td>
</tr>
<tr>
<td></td>
<td>- QR/Barcode: Provides the option to search by QR code or barcode.</td>
</tr>
<tr>
<td>Table name</td>
<td>This field only appears if you select List as the input type. The table for the field you want to create a UI parameter for. For example, Incident.</td>
</tr>
<tr>
<td>Field name</td>
<td>This field only appears if you select List as the input type. The field name you want to create for the UI parameter. For example, Priority.</td>
</tr>
<tr>
<td>Input style</td>
<td>How the user interacts from the UI. Choose from inline or popup. For this example, choose Inline.</td>
</tr>
<tr>
<td>Default value type</td>
<td>The value that appears by default in the UI field. Select one of the following options.</td>
</tr>
<tr>
<td></td>
<td>- None: There is no default text. This works well for a list input type.</td>
</tr>
<tr>
<td></td>
<td>- Manual: An additional field appears for you to enter a default term. For example, Search for a field. The manual default works well for search or text input types.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>Whether or not the user is required to enter information for that field. For this example, leave this check box cleared.</td>
</tr>
<tr>
<td>Placeholder</td>
<td>Text that appears below the field type. This option does not appear if you have a default value selected.</td>
</tr>
</tbody>
</table>

6. In the Screen UI Parameter Mapping section, make sure the following fields are completed with the correct values.
   - Data Item Parameter: The name of the parameterized data item you created. For example, Priority.
   - UI Parameter: The name of the UI parameter in the next section of the Parameter settings section. For example, Priority.

7. In the User Input Parameter Definition, click Save.
8. Click Save.
9. Optional: If you want the field on the mobile screen to be automatically populated with a value, configure the autofill parameters.
   a) In the Autofill Parameters section, click the add icon (+).
b) In the Name field, enter a name for the autofill parameter.

c) In the Input type field, select from the following options.
- GPS location: Automatically inputs the location of the mobile device.
- Date: Automatically inputs the current date for the mobile device.
- User: Automatically inputs the currently logged in user.

d) Click Save.

10. Complete any additional applet fields as needed. For more information on creating an applet, see Create an applet.

11. Click Save.

Configure a list applet

The list screen provides a list view of information. From the applet, users can access a detail view of each item and depending on the configuration, an activity stream screen and a related lists screen.

The followings steps provide a more detailed view of how to create a list applet.

1. To create an applet, follow the instructions in Create an applet, then select the List applet template.

   You can make changes to the applet name, description, icon, and whether or not the applet is available in offline mode after you create the applet by clicking the Properties button in the applet tab header.

2. From the primary screen tab of the configure screens page, in the data item field, select a data item from the list. If the list is disabled, no data items have been created. To create a data item, click the plus icon next to the list. Use the new tab that opens to create the data item.

   For more information on creating a data item, see Create a data item for a mobile application.

3. Optional: Click Change list item to select the header pattern you want to use for the list view and the details screen view.

   Note: If you change the list item or the data item, any existing header configuration is deleted.

   If the pattern you choose includes an image field indicated by a circle, make sure you select an image type field from the slushbucket. For example, select the Caller > Avatar field. Select the image field, or any other dot-walked fields, before you select the parent field.
4. In the field configuration section, use the slushbucket to add fields to the list preview, the details screen header, and the details screen body. You can add the same fields from the primary header to the details header by clicking **Replicate from primary**.

Use the pattern mapping preview to see how the fields appear. You must add at least one field to the header fields for the primary screen and the details screen.

5. To determine whether or not an applet displays in the application folder, in the header of the applet tab, clear or select the Display option.

You can configure some applets just for users to navigate to. In which case, you would not want the applet to appear on the main application screen. For example, a user could navigate to a list of callers from the Caller field on a form. The list of callers is a standalone applet that you only want users to access from the Caller field.

6. To make any changes to the applet, such as the name, description, icon, and whether or not the applet is available in offline mode, click the **Properties** button in the applet tab header.

7. Click **Save**.
Inc list

INC0000016  1 - Critical
Rain is leaking on main DNS Server
State: In Progress
Assigned to: ITIL User

INC0000019  2 - High
Can't launch 64-bit Windows 7 virtual machine
State: In Progress
Assigned to: Bud Richman

INC0000020  5 - Planning
I need a replacement iPhone, please
State: In Progress
Assigned to: ITIL User

INC0000025  1 - Critical
Need to add more memory to laptop
State: In Progress
Configure a map screen

Use map screens to display location-based information on a map. When you create a data item for a map screen, make sure you use information that can be plotted on a map.

Role required: admin

1. To create an applet, follow the instructions in Create an applet, then select the Map applet template.
2. Optional: Clear or select additional screens for the map applet. For example, you can add an activity stream screen, a related list screen, or a details screen.
3. From the primary screen tab of the configure screens page, in the data item field, select a data item from the list. If the list is disabled, no data items have been created. To create a data item, click the plus icon next to the list (+).

For more information on creating a data item, see Create a data item for a mobile application.

4. In the field configuration section, use the slushbucket to add fields to the primary screen, the details screen header, and the details screen body.

Use the pattern mapping preview to see how the fields appear. You must add at least one field to the header fields for the primary screen and the details screen.

Map screens require specific field types to be present in order to display information correctly. The E5 location requires a reference to an item in the cmn_location table, such as an address.
or a city. The phone icon in the E6 location is not required but is looking to a reference to a phone number.

Map applet primary
Configure a URL screen

URL screens open a URL from within the ServiceNow app. You can configure relative URLs to open pages within the ServiceNow platform.

Role required: admin

1. In Studio, navigate to the application you want to create a URL applet for.
2. Click Create an applet.
3. Select the URL applet template.
4. Complete the following fields as needed.

**URL screen fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of your screen. This name appears as a tile in the mobile application.</td>
</tr>
<tr>
<td>Icon</td>
<td>An image for the tile on the application homepage.</td>
</tr>
<tr>
<td>Hidden</td>
<td>Whether or not the screen is accessible from the application homepage.</td>
</tr>
<tr>
<td>Description</td>
<td>Additional information about the screen.</td>
</tr>
<tr>
<td>URL</td>
<td>The URL you want the send the user to.</td>
</tr>
<tr>
<td>Open in external browser</td>
<td>Whether or not the new page opens within the ServiceNow app or in an external browser application on the mobile device.</td>
</tr>
<tr>
<td>Relative URL</td>
<td>Whether or not the URL uses a long address including the https:// or a short URL appended to the instance name. For example, if you wanted to send users to a knowledge base within an instance, you could type $knowledge as the URL. When relative URL is select, the system automatically appends whatever you type in this field to the instance URL so the screen routes to, for example, https://&lt;instance_name&gt;.service-now.com/$knowledge. You can use the relative URL option to route users to the mobile Service Portal. You can use the relative URL to link users directly to a query by including the query URL. For example, if you search for server in a knowledge base, query=server is included in the URL. The relative URL to link a user to a knowledge search for server would be $knowledge?query=server.</td>
</tr>
</tbody>
</table>

5. Click Save.
Configure a grouped list screen

Grouped lists provide an easily navigable view of a list of items grouped by a particular field.

Before you can create a grouped list applet, you need to create a data item with the group function defined. For more information on configuring a data item, see Create a data item for a mobile application. After creating the data item for the grouped by applet, leave the data item tab open so that you can access it later.

Role required: admin

1. To create an applet, follow the instructions in Create an applet, then select the Grouped List applet template.
2. Click New.
3. In the data item field, select the data item you created that has the grouped function defined.
4. In the Item View field, click the search icon ( ) to open a list of item views.
5. In the Item View window, click New.
6. In the Name field, enter a name for the item view.
7. In the Description field, include additional information about the item view.
8. In the Item View JSON field, paste the following JSON to set up an item view.

```json
{
    "Type": "ViewGroup",
    "Margin": {
        "Top": 15,
        "Bottom": 14
    },
    "Orientation": "Horizontal",
    "Alignment": "Center",
    "Distribution": "Auto",
    "Children": [
        {
            "Type": "ViewGroup",
            "Orientation": "Vertical",
            "Alignment": "Left",
            "Distribution": "Auto",
            "Children": [
                {
                    "Type": "Text",
                    "Text": "PLACEHOLDER",
                    "CellId": "sys_group_by",
                    "TextColor": "#293e40",
                    "TextAlignment": "Left",
                    "MaxLines": 2,
                    "Font": {
                        "Weight": "regular",
                        "Size": 16
                    }
                }
            ]
        },
        {
            "Type": "ViewGroup",
            "Margin": {
                "Left": 16,
                "Right": 16
            }
        }
    ]
}
```
9. In the Dependency fields field, type `sys_group_by, sys_count`.
10. In the Data Item field, select the data item you created for the grouped list.
11. In the Item View window, click **Submit**.
12. In the Grouped List screen, click **Submit**.
My Group Incidents

Grouped by category

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>1</td>
</tr>
<tr>
<td>Inquiry / Help</td>
<td>3</td>
</tr>
<tr>
<td>Network</td>
<td>1</td>
</tr>
<tr>
<td>Software</td>
<td>5</td>
</tr>
</tbody>
</table>
Configure a calendar screen

Use the calendar applet to display scheduled tasks in a calendar view.

Role required: admin

1. To create an applet, follow the instructions in Create an applet, then select the List applet template.
2. Click New.
3. In the calendar screen record, click Submit.
4. In the Calendar Data Sources related list, click New.
5. From the Calendar Data Source tab, complete the following fields as needed.

### Calendar data source fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screen</td>
<td>The screen you just created for the calendar.</td>
</tr>
<tr>
<td>Application</td>
<td>This field is automatically populated with the current application.</td>
</tr>
<tr>
<td>Data Source Label</td>
<td>The name you want to appear as the source of information in the calendar.</td>
</tr>
<tr>
<td>Data Item</td>
<td>The name of the data item you created for the calendar screen. For more information on creating a data item, see Create a data item for a mobile application.</td>
</tr>
<tr>
<td>Start Date Field</td>
<td>The date the calendar uses as the start date for each item in the table. You can only choose from date fields.</td>
</tr>
<tr>
<td>End Date Field</td>
<td>The date the calendar uses as the end date for each item in the table. You can only choose from date fields.</td>
</tr>
</tbody>
</table>

6. Click Submit.
7. Navigate back to the calendar screen.
8. In the Master Item related list, click New.
9. Complete the Master Item fields as needed.

### Master Item fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for the master item</td>
</tr>
<tr>
<td>Screen</td>
<td>The screen you are configuring the master item for. Select the screen you created for the calendar.</td>
</tr>
<tr>
<td>Table</td>
<td>The table the master item is pulling data from. This field is automatically populated with the table you configured for the data item.</td>
</tr>
<tr>
<td>Condition</td>
<td>Any conditions you want to apply to the table.</td>
</tr>
</tbody>
</table>

© 2019 ServiceNow, Inc. All rights reserved.

ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item view</td>
<td>Determines how the information displays on the calendar screen. For more information on configuring the item view for a calendar, see Configure the calendar screen item view.</td>
</tr>
</tbody>
</table>

10. In the master item tab, click **Submit**.
Calendar screen
Configure the calendar screen item view

For the calendar screen you need to configure an item view to determine how information displays on the screen.

You can only access the item view from a field on the calendar master item record. Therefore, you should be creating a calendar screen if you are following these steps.

1. Navigate to a calendar screen.
2. From the master item related list, open the master item you created for the calendar.
3. In the Item View field, click the search icon ( ) to open a list of item views.
4. In the Item View window, click New.
5. In the Name field, enter a name for the item view.
6. In the Short Description field, include a description of the item view.
7. In the Item View JSON field, define the item view by pasting the following JSON. Update the <field_name> placeholders to match with the fields listed in the Dependency Fields field.

```json
{
    "Type": "ViewGroup",
    "Margin": {
        "Top": 16,
        "Bottom": 16
    },
    "Orientation": "Horizontal",
    "Alignment": "Center",
    "Distribution": "Auto",
    "Children": [
        {
            "Type": "ViewGroup",
            "Orientation": "Vertical",
            "Alignment": "Left",
            "Distribution": "Auto",
            "Children": [
                {
                    "Type": "Text",
                    "Margin": {
                        "Top": 3
                    },
                    "CellId": "<field_name>",
                    "TextColor": "#293e40",
                    "TextAlignment": "Left",
                    "MaxLines": 1,
                    "Font": {
                        "Weight": "regular",
                        "Size": 12
                    },
                    "Width": 60,
                    "Text": "PLACEHOLDER"
                },
                {
                    "Type": "Text",
                    "Width": 60,
                    "Margin": {
                        "Top": 9
                    },
                    "Text": "PLACEHOLDER",
                    "CellId": "<field_name>",
                    "TextColor": "#293e40"
                }
            ]
        }
    ]
}
```

© 2019 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.
8. In the Dependency fields field, in a comma separated list, type the names of fields included in the JSON. For example, expected_start, estimated_end, short_description, number. The dependency fields should match the <field_name> values in the JSON.

9. Optional: Change the color of the dot that appears in the calendar for each data source by changing the hex color associated with the BackgroundColor.

10. In the Data Item field, add the data item you created for the calendar screen.

11. In the item view window, click Submit.

Configure an employee directory screen

Configure an employee directory applet to provide a list of employees in the app.

Role required: admin

1. To create an applet, follow the instructions in Create an applet, then select the Employee Directory applet template.

2. From the data item list, select a data item. If the list is disabled, no data items have been created. To create a data item, click the plus icon next to the list (+).

   For more information on creating a data item, see Create a data item for a mobile application. For the employee directory applet, make sure you create a data item with a list of users, for example, from the Users (sys_user) table.

3. Use the slushbucket to add fields to the header. Use an image field where the item preview shows an image (E1).

   ![Employee list item](image)

   Employee list item

4. From the Details tab, click Replicate from primary to add the same fields to the detailed view for the employee profile.
5. In the Body section, add fields such as phone number or email. You can associate smart buttons with fields in the body section that allow users to open other native apps on a mobile device, such as an email client.

6. Click **Save**.

Configure a related list screen

Related list screens are included with certain applets, for example, the list applet. Configure the lists that appear as related lists.

Related lists behave the same way in a mobile device as they do on the platform. Only certain related lists are available depending on the table you are in. Before you set up an applet with a related list, determine which related lists are available.

The related list screen uses the data item you create for the applet to determine which related lists are available.

You need to create an applet for users to navigate to from the related list. Make sure the tables for the data items you create for each applet have a relationship with each other that will allow you to create the related list.

Create a list applet for a related list to navigate to

When you create a related list screen for an applet, you need to have an applet for the related list to navigate to.

1. Follow the steps for **creating a data item with parameters** to create a data item for the list applet.
   
   The parameter you create for the data item should query the database for the main field for the list. For example, for Task SLAs, create a parameter for the Task field.
2. **Create a list applet** for the related list to navigate to.
   Make sure you toggle the Display switch to off so that users can only access the list from the related list, instead of the application homepage.

3. Click **Save**.
Create an applet with a related list

After you create a list applet for the related list to navigate to, you can create an applet that includes a related list.

1. Create a data item for the applet.
2. Navigate to **Application > Module**.
3. Create an applet with a template that includes the related list screen. For example, List, Map, or Employee Directory.

![Screen preview](image)

For more information on creating an applet, see [Create an applet](#).

4. In the applet configuration screen, open the Related List tab.
5. Click the add icon.
6. In the Relationship list, search for the related list you want to add.
   
   The relationship field uses the table you selected for the data item to determine which relationships are available. You cannot add a relationship that does not exist, for example, you cannot add a Roles related list to an Incident.

   For a list of related lists available to a particular table, in the platform navigate to a form on the table you want to create a related list for. Then from the context menu click **Configure related lists**. You can only associate related lists shown in the slushbucket.

7. In the Destination field, select the applet you created previously to navigate to from the related list. For example, **ITSM > Task SLAs**.
### Screen Configurations

#### Related List

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Destination Applet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task SLA→Task</td>
<td>Task SLAs</td>
</tr>
</tbody>
</table>

#### Related list configuration

8. Click **Save** to save the applet.
Functions

Use functions in the ServiceNow Agent mobile app to define what your users can do with your applets. Functions let you perform actions like approving tasks, updating records, sending emails, or making a phone call from within the app.
Function types

You can use three different types of functions to configure actions in the mobile app.

Actions
Functions that let your users change data, such as assigning a task or adding a comment to a record. Action functions require a write-back action item to operate.

Navigations
Functions that let your users move to a new screen. These functions let your users open a record from a list or navigate from an employee user profile screen to a manager user profile screen.

Smart buttons
Functions that let your users perform another action, like sending an email, making a phone call, pulling up a location, or navigating directly to a URL.
Function behavior

Top menu functions display when your user taps the button in the upper right corner of the app screen. Your user can only see the top menu button when you configure functions to use the top menu.

You can configure multiple functions for the top menu. These functions appear as a list when your user taps the button.
Functions that your users enable by tapping an item in a list and swiping to the left or right. This swipe action reveals the swipe functions.

You can specify which direction to swipe when you configure your function. You can have a different set of functions available, depending on which direction a user swipes.

**Assigned Incidents**

- **In Progress**
  - **My desk phone stopped working**
    - Caller: Bertie Luby
    - Priority: 3 - Moderate

- **In Progress**
  - **The USB port on my PC stopped working**
    - Caller: Armando Papik
    - Priority: 5 - Planning
Field functions in the details screen that your users tap to activate. You can only assign a single field function to a field in a details screen.

Use a field function with a navigation function to direct a user to a related record, or use a smart button function to send a text or email to a contact.

**Function conditions**

Use the condition section of a function record to determine when that function is available. A condition can be declarative or scripted. Declarative conditions use a condition builder to create a condition. Scripted conditions use a script, which must evaluate to true to make the function available.
Condition fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition Type</td>
<td>• <strong>Declarative</strong>: Function availability is based on a condition that you create with the condition builder.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Script</strong>: Function availability is based on a script.</td>
</tr>
<tr>
<td>Table</td>
<td>• If you select <strong>Declarative</strong> in the <strong>Condition Type</strong> field, this field appears as a condition builder. For more information on using the condition builder, see <a href="#">condition builder</a>.</td>
</tr>
<tr>
<td></td>
<td>• If you select <strong>Scripted</strong> in the <strong>Condition Type</strong> field, this field appears as a text area. Enter a script in this field. The function is available when this script evaluates to true.</td>
</tr>
<tr>
<td>Condition</td>
<td>Field that appears as a condition builder</td>
</tr>
<tr>
<td>Roles</td>
<td>Roles that are required to use this function.</td>
</tr>
</tbody>
</table>

This example is from the **Accept** function for Work Order Task records.

```javascript
current.state == 16 && current.assigned_to == gs.getUserID()
```

The function is available for records in the **Assigned** (16) state, and the record is assigned to the current user. If the function has a value in the **Roles** field, the role requirements are applied in addition to this scripted condition.

Configure a navigation function

Configure how your users move from one place to another in ServiceNow Agent, such as navigating to a record from a field on another record. To configure a navigation function, you must first create the navigation and then associate it with an action in the app.

Before you create a navigation function, you should have at least two applets; otherwise, you won’t have anything to navigate to.

Role required: admin

1. In Studio, navigate to **Mobile Development > Functions > Navigation**.
2. Click the pop-out icon (clientes) to open the Navigations list.
3. Click **Create a new navigation**.
4. In the Properties section, in the **Name** field, type a name for the navigation.
   Because you can reuse navigations, use a name that you can easily identify.
5. In the **Destination** field, select the application and then the applet that you want to navigate to. For example, **ITSM > Open incidents**.
Depending on the screen that you select, you may need to include parameter settings.

6. In the **Description** field, type a description for your navigation function.

7. Use the Available Offline toggle switch to determine whether the navigation function is available offline.
   You must mark the applet and the application that contains the navigation function as available offline for the navigation function to work offline. For more information on offline mode, see [Mobile offline mode](#).

8. Optional: In the Parameter Setting section, update the redirection parameter fields.
   The Parameter settings section only appears if you add a destination applet that has a parameterized data item.
   a) Click the value in the parameter name field to open the parameter settings.
   b) Complete the following fields as needed. Some of the fields vary depending on the value that you select for the **Type** field.
### Redirection parameter fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination UI parameter</td>
<td>Field that is automatically filled with the name of the parameter that you created for the data item. If the parameter type is User Input, make sure that the destination UI parameter name is user friendly.</td>
</tr>
<tr>
<td>Type</td>
<td>Source of information for the parameter. Select from the following options:</td>
</tr>
<tr>
<td></td>
<td>• Field: The parameter gets the information from a field on a table.</td>
</tr>
<tr>
<td></td>
<td>• Constant: The parameter is fulfilled by a constant value.</td>
</tr>
<tr>
<td></td>
<td>• Source UI parameter</td>
</tr>
<tr>
<td>Source field</td>
<td>Field that appears if field is selected as the redirection parameter type. Select the field that you want to use as the source for the parameter. Available fields are determined by the table that is defined in the data item.</td>
</tr>
<tr>
<td>Constant value</td>
<td>Field that appears if Constant is selected as the redirection parameter type. Enter the value that you want to always appear for the data item parameter.</td>
</tr>
<tr>
<td>Source UI parameter</td>
<td>Field that appears if Source UI Parameter is selected as the redirection parameter type. From the list, search for the UI parameter for the source screen.</td>
</tr>
</tbody>
</table>

9. Optional: In the Advanced Configurations section, determine the display conditions and the roles required for the navigation function.
   a) To determine the conditions that are required for the navigation function to display, in the Display Conditions section, select a table from the list.
   b) In the condition builder, select filter conditions to control whether the navigation function appears. For example, if you want users to be able to navigate to a problem from an incident form, you can restrict the navigation function to only appear if the problem field is not empty.
   c) To define the roles that can use the navigation function, in the Roles Permission section, move roles to the Selected roles list.

10. Click **Save**.

After you create a navigation function, you need to associate it with a specific location in the mobile app. You can add a navigation function to a top menu, a swipe action, or to a specific field. For more information on associating the navigation function to a location, see Associate a function with a location in the app.
Configuration for smart buttons

Configure smart buttons so that ServiceNow Agent can interact with native applications on a mobile device, such as a phone, map, or email applications.

Smart button types

Your users can use smart buttons to quickly perform actions that they specify outside the app. With a smart button, your users can navigate to a location on a map, send a text message or email to a contact, or open a URL in a browser. Your users can select the following options.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>Address that opens the GPS location mobile menu. Users can select which map app to use to open the address.</td>
</tr>
<tr>
<td>Email</td>
<td>Email that is created with the email address and subject that is automatically populated by the button properties.</td>
</tr>
<tr>
<td>Phone</td>
<td>List of phone number options, such as a text message, call, or saved phone number.</td>
</tr>
<tr>
<td>URL</td>
<td>Field with a web address that you can open in a web browser such as add a URL to open the company’s website.</td>
</tr>
</tbody>
</table>

Smart button properties

Use the smart button properties to define the action that a smart button performs.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for the smart button.</td>
</tr>
</tbody>
</table>
| Type       | Type of smart button:  
|            | • Address  
|            | • Email  
|            | • Phone  
|            | • URL |
| Description | Description for the smart button. |
| Context    | Source of the address, email, and phone number used by your smart button.  
|            | • Record: Select a field on a table that contains the needed information.  
<p>|            | • Global: Manually enter a static value that contains the information. |
| Relative Url | Relative URL. This property appears when the Url Type is selected. |</p>
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email address</td>
<td>Email address that is used by this smart button. This property appears when the Email type is selected and the Global context is selected.</td>
</tr>
<tr>
<td>Phone number</td>
<td>Phone number that is used by this smart button. This property appears when the Phone type is selected and the Global context is selected.</td>
</tr>
<tr>
<td>Table</td>
<td>Table where the address, email, and phone number that is used by your smart button is located.</td>
</tr>
<tr>
<td>Field</td>
<td>Field where the address, email, and phone number that is used by your smart button is located. This field must be on the table that you selected in the Table property.</td>
</tr>
<tr>
<td>URL label</td>
<td>Visible text of the URL. This property appears when the URL type is selected and Relative Url is selected.</td>
</tr>
<tr>
<td>Instance URL</td>
<td>URL of the instance that the relative link uses. This property appears when the URL type is selected and Relative Url is selected.</td>
</tr>
<tr>
<td>Relative link</td>
<td>Relative link within the instance that is selected in the Instance URL property. This property appears when the URL type is selected and Relative Url is selected.</td>
</tr>
<tr>
<td>Email subject</td>
<td>Subject for the email message. This property appears when the Email Type is selected.</td>
</tr>
<tr>
<td>Email content</td>
<td>Content of the email message. This property appears when the Email Type is selected.</td>
</tr>
</tbody>
</table>
| Phone type       | Type of phone for the recipient:  
  - Cellular  
  - Landline  
  This property appears when the Phone Type is selected.                                                                                                                                                |
| SMS              | Content of the SMS message sent. This property appears when the Phone Type is selected and the Phone type is set to Cellular.                                                                                  |
| Available Offline| Option to display the smart button in offline mode.                                                                                                                                                         |

**Smart button advanced configurations**

Use advanced configurations to control when your button appears, based on conditions or roles. You may, for example, want to hide an email button for records that have no email address or display a URL link only to your users.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Conditions</td>
<td>Conditions under which the smart button appears.</td>
</tr>
</tbody>
</table>
Configure a smart button function

Configure smart buttons so that your users can quickly perform another action, such as sending an email, making a phone call, pulling up a location, or navigating directly to a URL.

Role required: admin

1. In Studio, navigate to **Mobile Development > Functions > Smart Buttons**.
2. Click the pop-out icon (확대) to open the Smart buttons list.
3. Click **Create a new smart button**.
4. In the Properties section, in the **Name** field, type a name for the smart button. Because you can reuse smart buttons, use a name that you can easily identify.
5. In the **Type** field, select a type for the smart button. Your users can select the following options.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>GPS location mobile menu. You can select which map app to use to open the address.</td>
</tr>
</tbody>
</table>
| Email | Email with the email address that is automatically populated in the To: line. If you select this option, these additional fields appear:  
  - Email subject: Automatically populates the subject line with any text that you enter.  
  - Email content: Automatically populates the content of the email with any text that you enter. |
| Phone | List of phone number options, such as text message, call, or saved phone number. If you select this option, these additional fields appear:  
  - Phone type: Whether the phone number is a cell phone or a landline. If you select a **Landline**, the SMS field disappears.  
  - SMS: Automatically populates a text message with any text that you enter. |
| URL | Web address that opens in a web browser. An example is a URL that is added to open the company’s website. |

6. In the **Table** field, select the table that you want to add a smart button to.
7. In the **Field** field, select the field that you want to add a smart button to. Make sure that your selected field is appropriate for the type of smart button that you are creating. For example, if you are creating a smart button with the type Phone, select a field that includes a phone number.
8. Use the Available Offline toggle switch to determine if the smart button function is available offline.

You must mark the applet and the application that contains the smart button function as available offline for the smart button function to work offline. For more information on offline mode, see **Mobile offline mode**.
9. Optional: Under Advanced Configuration, in the Display Conditions section, create conditions to limit when the smart button appears. For example, if you are creating a Phone type of smart button, you could create a condition to make sure that the phone number field is not empty.

10. Optional: In the Role Permissions section, add roles to limit who the smart button can appear for.

After you create a smart button function, you need to associate it with a specific location in the mobile app. You can add a smart button function to a top menu, a swipe action, or to a specific field. For more information on associating the smart button function to a location, see Associate a function with a location in the app.

Configure a smart button using a parametrized URL

Use a parametrized URL to include record-specific information in your smart buttons.

Role required: admin

Parameters improve the functionality of smart buttons. In this task, you configure a smart button to link to a list of knowledge articles. The button uses the short description of the current incident as that search criteria for the knowledge article list.

Watch this two-minute video to learn how to find a relative link in your instance.

1. Navigate to System Applications > Studio and open your mobile app.
2. Navigate to Mobile Development > Functions > Smart Buttons and click the pop-out icon to open the Smart buttons list.
3. Click New.
4. Fill in the smart button properties as shown in the table.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>KB with Short Description</td>
</tr>
<tr>
<td>Type</td>
<td>URL</td>
</tr>
<tr>
<td>Relative URL</td>
<td>Checked</td>
</tr>
<tr>
<td>Table</td>
<td>Incident (incident)</td>
</tr>
<tr>
<td>URL Label</td>
<td>Show KB</td>
</tr>
<tr>
<td>Instance URL</td>
<td>Instance URL. This value is automatically populated.</td>
</tr>
<tr>
<td>Relative Link</td>
<td>sp?id=search&amp;spa=1&amp;t=kb&amp;q={{short_description}}</td>
</tr>
</tbody>
</table>

5. Click Save.
6. Associate your new button to your app. For details on this process, see Associate a function with a location in the app.

When you tap this button, you see a list of knowledge base articles that use the short description for the incident as a search term.

Create an action item

Create an action item to associate with the action function. You use action items to define what the action function is and how it works.

Most action items use parameters.
The mobile app does not allow users to do anything that the platform does not allow. For example, if you use access control lists (ACLs) to prevent a user from closing an incident without adding a resolution code and notes, the user cannot close an incident in the app without adding a resolution code and notes. When you create actions, you need to keep this in mind so that you can add the correct parameters.

Role required: admin

1. In Studio, navigate to **Mobile Development > Action Items**.
2. Click the pop-out icon (⋮) to open the Actions items list.
3. Click **Create a new action item**.
4. On the form, fill in the fields.

### Action item form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the action item. You can have multiple action items with the same name. Make sure that you choose a name that is easily identifiable.</td>
</tr>
<tr>
<td>Description</td>
<td>More information to help you identify the action item.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of action item. Choose from the following:</td>
</tr>
<tr>
<td></td>
<td>· New</td>
</tr>
<tr>
<td></td>
<td>· Update</td>
</tr>
<tr>
<td></td>
<td>· Delete</td>
</tr>
<tr>
<td></td>
<td>· Script</td>
</tr>
<tr>
<td></td>
<td>Different fields appear on the action item form depending on the type of action that you select.</td>
</tr>
<tr>
<td>Table</td>
<td>Table that the action item applies to, such as an Incident (incident) table.</td>
</tr>
<tr>
<td>Execution Script</td>
<td>Script that is executed by the action. This field only appears if you select <strong>Script</strong> as the type.</td>
</tr>
<tr>
<td>Use current record as condition</td>
<td>Separate set of query conditions for the action item. If selected, the <strong>Query conditions</strong> field is disabled. For update or delete actions, you must define the record that you are updating or deleting by providing a sys_ID. Marking <strong>Use current record as condition</strong> as true allows you to do this without creating a parameter.</td>
</tr>
<tr>
<td>Query Condition</td>
<td>Filter conditions that apply to the action item.</td>
</tr>
<tr>
<td>Set field values</td>
<td>Field values for an action. For example, if you want to create an action that updates an incident with a state of Resolved, you can the field values for State = Resolved. You can also create parameterized items to pass into the field value.</td>
</tr>
</tbody>
</table>
5. Click **Submit**.

The following example uses a script to assign a task to the current user, using the SMTask object. The first `if` statement checks to see that the input is a valid `wm_task` record and ends the script if it is not. The second `if` statement contains code that will assign the task to the current user, if the user has permission, as determined by the `canAssignToSelf` method. This action was done as a script rather than an update so that these checks could be included.

```javascript
(function WriteBackAction(input) {
    var smTask = new global.SMTask();
    var wotGR = new GlideRecord("wm_task");
    if (!wotGR.get(input.sys_id)) {
        gs.error("wot_assign_to_me write-back action - failed to find work order task");
        gs.addErrorMessage(gs.getMessage("Task assignment failed."));
        return;
    }
    if (smTask.canAssignToSelf(wotGR))
        smTask.assignToMe(gs.getUserID(), input.sys_id);
    else
        gs.addErrorMessage(gs.getMessage("Not a valid task assignment."));
})(input);
```

If you use parameters for the action item, you can call them in the script. The call in the script needs to match the parameter name exactly. For example, if the parameter name is `wb_wot_reject_work_note`, you can call it in the script using `gr.work_notes = input.wb_wot_reject_work_note;`.

Associate the action item with an **action function**.

Create an action item with parameters

Create an action item with parameters to define the changes that are being made to an action and how the changes get made. Parameters determine the information that you pass into the action so that you make changes to the correct record and to enforce required fields as needed.

1. In Studio, navigate to **Mobile Development > Action Items**.
2. Click **Create a new action item** or select an existing action item to add parameters to.
   For more information on creating an action item, see **Create an action item**.
3. In the Item Parameters related list, click **New**.
4. In the **Name** field, enter a name for the action item.
   You can have multiple parameters with the same name, so choose a name that you can easily identify later.
5. From the Type list, select the type of parameter. The type determines how the user interacts with the mobile UI. For example, a type of Decimal or Integer tells the mobile device to open a numbers-only keypad. Select from all of the following types:
   - **String**: Uses a full keyboard for input. Use the String type for list parameters, such as priority or state, or for reference fields, such as assigned to or caller.
   - **Integer**: Uses a numbers-only keypad for input
   - **Decimal**: Uses a numbers-only keypad for input
   - **Boolean**: Uses a true or false selection option
   - **DateTime**: Uses a calendar with an exact time selector
- **Date**: Uses a calendar

6. In the **Item Parameter** tab, click **Save**.

7. In the **Action Item** tab, click the contextual reference value icon ( ), to add the item parameter that you just created as a condition in the condition builder for the action item.

8. Click **Update**.

---

**Create an action function**

Create an action function to let your users change something in the database. For example, if your users want to make an update or add a comment to a record, you must create action function first.

Create an action item.

**Role required**: admin

1. In Studio, navigate to **Mobile Development > Functions > Actions**.

2. Click the pop-out icon ( ) to open the Actions list.

3. Click **Create a new action**.

4. In the Properties section, in the **Name** field, type a name for the action.

   Because you can reuse actions, use a name that you can easily identify later.

5. On the form, fill in the fields.

**Action function form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for the button. You can have multiple buttons with the same name. Make sure that you choose a name that is easily identifiable.</td>
</tr>
<tr>
<td>Label</td>
<td>Not used. The label for the button is defined in the button instance record.</td>
</tr>
<tr>
<td>Description</td>
<td>Additional information about the button.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of button. This field automatically populates with a write-back action as the kind of button.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Context</td>
<td>Area that you want to apply the action to. Select from one of the following:</td>
</tr>
<tr>
<td></td>
<td>• Record: Apply an action at the record level. For example, use this option to create a button that changes the state of a record. You can set the context to Record for actions on a particular field, a particular record, or swipe actions. If the action function includes a UI parameter with the Input source field set to Auto fill, you must specify the table in the action function Table field.</td>
</tr>
<tr>
<td></td>
<td>• Global: Apply an action at the global or list level. For example, use this option to add a button that creates a record. You can only set the context to Global for actions in the list menu or to swipe actions. Do not use Global for actions that are in the context of a record or actions that include a UI parameter with the Input source field set to Auto fill.</td>
</tr>
<tr>
<td></td>
<td>For more information on button locations, see <a href="#">Associate a function with a location in the app</a>.</td>
</tr>
<tr>
<td>Table</td>
<td>List that the button applies to. For example, Incident.</td>
</tr>
<tr>
<td>Condition</td>
<td>Conditions that must be met in order to use the button. For example, you can prevent users from resolving an incident that is in a state of closed, resolved, or canceled. For more information on using the condition builder, see <a href="#">condition builder</a>.</td>
</tr>
<tr>
<td>Write-back action item</td>
<td>Associated action item with the action. Select a write-back action item that you created from the list to associate with the action. For more information on action items, see <a href="#">Create an action item</a>.</td>
</tr>
<tr>
<td>Show confirmation message</td>
<td>Confirmation messages that appear to verify the user wants to continue with the action.</td>
</tr>
<tr>
<td>Success message</td>
<td>Text for a success confirmation message. Use curly braces to create a dynamic success message. For example, {{\text{number}}} has successfully updated.</td>
</tr>
<tr>
<td>Failure message</td>
<td>Text for a failure confirmation message. Use curly braces to create a dynamic failure message. For example {{\text{number}}} could not be updated.</td>
</tr>
<tr>
<td>Jump to screen after successful write-back action execution</td>
<td>User that gets redirected to a screen after completing the action.</td>
</tr>
<tr>
<td>Destination screen</td>
<td>If Jump to screen after successful write-back action execution is selected, select a screen to direct the user to after they complete the action.</td>
</tr>
</tbody>
</table>
After you create an action function, you need to associate it with a specific location in the mobile app. You can add an action function to a top menu, a swipe action, or to a specific field. For more information on associating the action function to a location, see Associate a function with a location in the app.

Create an action function with parameters

Configure the UI parameters for the action function that have parameters.

The UI parameter fields do not appear on the action function unless you associate an action item with parameters to the action function.

1. In Studio, navigate to Mobile Development > Functions > Actions.
2. Complete the action function fields as needed. In the action item field, include an action item with parameters.
   For more information on completing the fields for the action function, see Create an action function.
3. In the UI Parameters related list, click New.
4. In the form, complete the following fields as needed.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for the UI parameter. You can have multiple UI parameters with the same name so make sure that you choose something you can easily identify later. If the parameter requires user input, make sure that the name you add here is user friendly.</td>
</tr>
<tr>
<td>Parameter type</td>
<td>Type of parameter. Choose Screen or Button.</td>
</tr>
<tr>
<td>Button</td>
<td>Button that you associate with this parameter. This field appears when the Parameter type field is set to Button.</td>
</tr>
<tr>
<td>Screen</td>
<td>Screen that you associate with this parameter. This field appears when the Parameter type field is set to Screen.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>Whether the user is required to enter information for that field.</td>
</tr>
<tr>
<td>Order</td>
<td></td>
</tr>
<tr>
<td>Input source</td>
<td>Source for the parameter input. Select User Input or Auto fill.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Input type            | UI that is used to complete the variable. The available options in this field depend on what the user enters in the **Input source** field. Select one of the following options:  
  - Text: Provides a simple text field. This option works best for fields that require free text, such as work notes or resolution details.  
  - List: Opens a list for the user to select from. This option works well for reference fields that require specific information.  
  - SearchList: Provides a search bar so that users can search in a list. Select a reference field on a table for users to search on.  
  - QR/Barcode: Provides the option to search by QRC or barcode.  
  - GPS Location: Auto-fills with the user’s location when the action is used.  
  - Date: Auto-fills with the date and time stamp of when the action is used.  
  - Constant: Auto-fills with a constant value. The **Constant value** field appears on the form when the user selects this option.  
  - Source field: Auto-fills from a specified field. The **Button parent table** and **Source field** fields appear on the form when the user selects this option.  
  - User: Auto-fills with the mobile user’s user record.  
<p>| | |
|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Input style           | How the user interacts in the UI. Choose from inline or popup.                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Constant value        | Static value to use for this parameter. This field appears if you use the <strong>Constant</strong> input type.                                                                                                                                                                                                                                                                                                                                                                                          |
| Placeholder           | Text that appears below the field type. This option does not appear if a default value is selected.                                                                                                                                                                                                                                                                                                                                                                                         |
| Table name            | Name of the table that you want to pull information from. This field appears if you use the List or the SearchList input type. Select a table that correlates with the action item. For example, if you create an action item that uses the Incident (incident) table, select the Incident (incident) table for the UI parameter as well.                                                                                                                             |
| Field name            | Name of the field that you want to pull information from. This field appears if you use the List or the SearchList input type. Select a reference field from the table that you selected. For example, select the Assign to field.                                                                                                                                                                                                 |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value type</td>
<td>Value that appears by default in the UI field. Select one of the following options.</td>
</tr>
<tr>
<td></td>
<td>· None: There is no default text. This option works well for a list input type.</td>
</tr>
<tr>
<td></td>
<td>· Manual: A field appears for you to enter a default term. For example, Search for a field. The manual default works well for search or text input types.</td>
</tr>
<tr>
<td></td>
<td>· Source field: Pulls in information from the field selected in the Field name section. A table name and field are required if you select this option.</td>
</tr>
<tr>
<td>Button parent table</td>
<td>Parent table for the source field. This field appears if you use the Source field input type.</td>
</tr>
<tr>
<td>Source field</td>
<td>Field that is used for the source field. This field appears if you use the Source field input type.</td>
</tr>
<tr>
<td>Carried</td>
<td>Whether this parameter a carried parameter. Use carried parameters to move information between different screens and actions.</td>
</tr>
</tbody>
</table>

5. Click **Save**.
6. From the Action function, in the Action parameter mappings related list, click **New**.
   a) In the **Button** field, if the field is not completed already, enter the name of the action function.
   b) In the **Item Parameter** tab, search for the item parameter that you created for the action item. For example, Assignee.
   c) In the **UI Parameter** field, search for the name of the UI parameter that you created.
   d) Click **Save**.

7. From the action function, click **Update**.

Associate the function with a specific applet. For more information on how to associate the action, see Associate a function with a location in the app.

**Associate a function with a location in the app**

Associate each function that you create for an app with a specific location to make it accessible to your users. You can associate most functions with a top menu, a swipe, or a specific field.

**Top menu functions**

You can create a top menu for either a primary or a details screen.

1. Navigate to an applet that you want to add a navigation function to.
2. From a primary or details screen, in the screen configuration section, switch to the **Functions** tab.
3. In the Top Menu Functions section, click the Add icon (+).

4. In the **Label** field, type a name for the navigation function. This name appears in the top menu in the app.

5. In the **Function** field, click the type of function that you want to add to the top menu. Select the name of the function that you created.

6. Click **Done**.
Assigned Incidents

In Progress

The USB port

Caller  Bud Richman

Priority  5 - Planning

Description

Reassign

Resolve

Add Comments

Edit

Caller

Bud Richman

Business service

Configuration item

*RETH_IRM
Swipe functions

Swipe functions only apply to primary screens that display a list. You cannot, for example, create a swipe function for a map screen.

1. Navigate to a list-type applet that you want to add a swipe function to.
2. From the primary screen, in the screen configuration section, switch to the Functions tab.
3. In the Swipe Functions section, click the Add icon ( + ).
4. In the Swipe Functions definitions window, in the Label field, type a name for the swipe function. This name appears when you swipe an item in a list.
5. In the Function field, click the type of function you want to add to the top menu, then select the name of the function you created.
6. In the Swipe Direction field, select Left or Right, depending on which direction you want the swipe to appear.
7. Click Done.
Assigned Incidents

In Progress

My desk phone stopped working

Caller  Bertie Luby
Priority  3 - Moderate

Resolve  Reassign  Add Comments

New
Field functions

Field functions only apply to items on the details screen in the body displayed area.

1. Navigate to an applet that you want to add a field function to.
2. From the details screen, make sure there are elements in the body display area that you can add field functions to.
3. Click the Functions tab.
4. In the Field Functions section, click the Add icon (+).
5. In the Field field, select a field from the list. Only fields that are configured for the body display area appear in the list.
6. In the Function field, click the type of function that you want to add to the top menu. Select the name of the function that you created.
7. Click Done.
<table>
<thead>
<tr>
<th>Configuration item</th>
<th>*BETH-IBM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Hardware</td>
</tr>
<tr>
<td>Subcategory</td>
<td></td>
</tr>
<tr>
<td>Impact</td>
<td>3 - Low</td>
</tr>
<tr>
<td>Urgency</td>
<td>3 - Low</td>
</tr>
<tr>
<td>Assignment group</td>
<td></td>
</tr>
</tbody>
</table>
Mobile parameter tutorials

Use parameters to create a variable or a placeholder that waits for input from either the user or the database. You can add parameters to a data item or an action item.

When you add parameters to a data item, the parameter looks for additional information before opening a screen. For example, you can create a data item that lets users filter incidents by priority. The data item parameter holds a place in the Priority field. When the user opens the screen, they can select the priority. For more information on configuring a data item with parameters, see Configure a data item with parameters.

When you add a parameter to an action function, that parameter looks for information from the user or the database before updating a record. For example, you can create an action function that allows a user to update the assigned to field from a swipe action. The action function parameter holds the space of the assigned to field. When a user uses the swipe action, they are prompted to select an assignee. For more information on creating an action function with parameters, see Create an action function with parameters.

In these tutorials, you learn how to create action and data items with parameters and associate them with their respective functions:

- Tutorial: Create an applet with a parameterized data item
- Tutorial: Create a navigation function with a parameterized data item
- Tutorial: Configure an action function with a parameterized action item

Tutorial: Configure a data item with parameters

Learn how to create an applet that lets users filter by priority before they can view a list. Use this tutorial as a guideline to help you understand how data items with parameters work in ServiceNow Agent.

In this tutorial, you learn how to:

- Create a data item with parameters.
- Configure the applet to use the parameterized data item.

Before you begin, navigate to Studio (System Applications > Studio) and create an application for your mobile app. For more information on setting up Studio for your mobile app, see Create an application for a mobile device in Studio.

Create a data item with parameters

Create a data item for a list of incidents.

To do this tutorial, you should be in Studio and have an application set up for ITSM. For more information on creating an application in Studio, see Create an application for a mobile device in Studio.

Role required: admin

1. In Studio, navigate to Mobile Development > Data Items.
2. Click the pop-out icon ( ) to open the data items list.
3. Click Create a new data item.
4. Enter this information in the listed fields:
   - Name: Open incidents
   - Table: Incident
ServiceNow Madrid Mobile navigation and configuration

- Description: List of open incidents with parameters

5. In the Query condition, add these conditions:
   - Active is true
   - State is New, In progress, or On hold

6. Click Save.

7. In the Parameter Definition section, click the add icon (+).

8. In the Name field, enter Priority as the parameter name.
   You can have multiple parameters with the same name, so choose a name that you can easily identify later.

9. In the Type list, select String.

10. In the Item Parameter tab, click Submit.

11. In the Data Item tab, from the condition builder, click And.

12. From the Choose field list, select Priority.

13. In the same row as the Priority field in the condition builder, click the contextual reference value icon (○).

14. Add the item parameter that you just created.

15. Click Save.

After you have completed the preceding steps, your query conditions should appears as shown in the image.

![Query conditions for a data item with parameters](image)

Assign the data item to an applet

After you create a data item with parameters, assign it to an applet.

1. In Studio, navigate to Mobile Development > Application > ITSM.
   If you do not have the ITSM application, create an application.
2. Click the pop-out icon to open the ITSM application.
3. Click Create an applet.
4. From the Create an applet window, in the Name field, type Open incidents.
5. In the Description field, type Open incidents sorted by priority.
6. Select List applet template.
7. Click New.
8. In the Open incidents applet tab, from the Data Item list, select Open incidents.
9. In the Field Configuration field, move these fields to the Selected Fields list:
   - Number
   - Priority
   - Short description
   - Assigned to
   - Assigned to > Avatar
   - State
10. From the Details tab of the open incidents applet, above the Selected Fields list, click Replicate from primary.
11. Click Save.

A parameter settings section appears under the data item.

12. In the Parameter setting, in the Screen UI Parameter Mapping section, make sure that these fields have the correct values:
   - Data Item Parameter: The name of the parameterized data item that you created, which in this case is Priority.
   - UI Parameter: The name of the UI parameter in the next section of the Parameter settings section, which should be Priority.

13. In the User input parameters section, click the first row in the table to open the User input parameter definition window.
14. Enter this information in the listed fields:
   - Name: Priority (this field should already have the name included)
   - Input type: List
   - Table name: Incident
   - Field name: Priority
   - Input style: Inline
15. Click Save.

When your user opens the applet, they should be prompted to select a priority level before opening the list.

Tutorial: Configure an action with parameters

Learn how to create a swipe action that allows a user to update the Assigned to field from a ServiceNow Agent list. Use this tutorial as a guideline to help you understand how actions with parameters work in the mobile app.

In this tutorial, you learn how to:
- Create an Assign to action item with parameters.
- Create an action function to change the incident assignee.
- Associate the action function with an incident list applet.
Create an action item with parameters

Create an **Assign to** action item with parameters.

To do this tutorial, you should be in Studio and have an application set up for ITSM. For more information on creating an application in Studio, see [Create an application for a mobile device in Studio](#).

Role required: admin

1. In Studio, navigate to **Mobile Development > Action items**.
2. Click the pop-out icon ( explored area ) to open the Action items list.
3. Click **Create a new action item**.
4. Enter this information in the listed fields:
   - Name: Assign to
   - Type: Update
   - Table: Incident
   - Use current record as condition: Selected
5. Click **Submit**.
6. In the Item Parameters related list, click **New**.
7. In the **Name** field, enter **Assignee** as the parameter name.
8. In the Type list, select **String**.
9. In the **Item Parameter** tab, click **Submit**.
10. In the **Action Item** tab, from the **Set field values** condition builder, search for **Assigned to**.
11. In the same row as the **Assign to** field in the condition builder, click the contextual reference value icon ( explored area ).
12. Add the item parameter that you just created.

13. Click **Update**.

Create an action function

After you create the action item with parameters, configure an action function to associate the action item with.

1. In Studio, navigate to **Mobile Development > Functions > Actions**.
2. Click the pop-out icon ( explored area ) to open the Action items list.
3. Click **Create a new action**.
4. On the form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Assign</td>
</tr>
<tr>
<td>Context</td>
<td>Record</td>
</tr>
</tbody>
</table>
5. Click **Submit**.
6. In the UI parameter related list, click **New**.
7. Enter this information in the listed fields:
   - Name: **Assign**
   - Input type: **List**
   - Table name: **Incident**
   - Field name: **Assigned to**
   - Input style: **Popup**
   - Default value type: **None**
8. Click **Submit**.
9. From the **Assign Action** function, in the Action parameter mappings related list, click **New**.
   a) In the **Button** field, if the field is not completed already, search for **Assign**.
   b) In the **Item Parameter** tab, search for **Assignee**.
   c) In the **UI parameter** field, search for **Assign**.
   d) Click **Submit**.
10. From the **Assign action** function, click **Update**.

**Assign the action function to an applet**

After you create an action function with a parameterized action item, assign the action item to an applet.

1. Create a data item for open incidents.
   a) In Studio, navigate to **Mobile Development > Data Items**.
   b) Click the pop-out icon ( ) to open the data items list.
   c) In the **Data Items** tab, click **Create a new data item**.
   d) In the **Name** field, type **Open incidents**.
   e) In the **Table list**, search for **Incident (incident)**.
   f) In the **Query condition section**, use the `<State><is one of> filter, and select the **New**, **In progress**, and **On hold** states.
   g) Click **Save**.

2. Create an applet for open incidents.
   a) In Studio, navigate to **Mobile Development > Application > ITSM**. If you do not have an ITSM application, create an application.
   b) Click the pop-out icon ( ) to open the ITSM application in a tab.
   c) Click **Create an applet**.
   d) In the **Name** field, enter **Open Incidents**.
e) Select **List** under **Choose the screen(s) template**

f) Click **New**.

g) From the Data Item list, select **Open incidents**.

h) In the **Field Configuration** field, move these fields to the Selected Fields list:
   - Number
   - Priority
   - Short description
   - Assigned to
   - Assigned to > Avatar
   - State

   You may need to dot-walk and add **Assigned to > Avatar** before adding the Assigned to field to the list.

i) From the **Details** tab of the open incidents applet, above the **Selected Fields** list, click **Replicate from primary**.

j) Click **Save**.

3. Click the pop-out icon (겠습니다) to open the Actions list.

4. Add the action function to a swipe action for the Open incidents list.
   a) From the tab for the Open incidents applet, click **Functions**.
   
   b) In the Swipe Functions section, click the Add icon (➕).
   
   c) In the **Label** field, type **Assign**.
   
   d) In the **Function** field, select **Action > Assign**.
   
   e) Click **Done**.

---

**Tutorial: Configure a navigation function with parameters**

Learn how to create a navigation from a field on a mobile form. Use this tutorial as a guideline to help you understand how navigation functions with parameters work in ServiceNow Agent.

In this tutorial, you learn how to:
- Create a data item with parameters
- Create an applet to navigate to
- Create a navigation function
- Associate the navigation function with an applet

Before you begin, navigate to Studio (**System Applications > Studio**) and create an application for your mobile app. For more information on setting up Studio for your mobile app, see [Create an application for a mobile device in Studio](#).

Create a data item with parameters

Create a data item for a list of users.

Role required: admin

1. In Studio, navigate to **Mobile Development > Data Items**.
2. Click the pop-out icon ( ) to open the data items list.
3. Click **Create a new data item**.
4. On the form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>List of callers</td>
</tr>
<tr>
<td>Table</td>
<td>User (sys_user)</td>
</tr>
<tr>
<td>Description</td>
<td>List of users</td>
</tr>
</tbody>
</table>

5. In the Query condition, add these conditions:
   - Active is true
6. Click **Save**.
7. In the Parameter Definition section, click the add icon ( ).
8. In the **Name** field, type `caller_sys_ID` as the parameter name.
   You can have multiple parameters with the same name, so select a name that you can easily identify later.
9. In the Type list, select **String**.
10. In the **Item Parameter** tab, click **Submit**.
11. In the **Data Item** tab, from the condition builder, click **And** to add a new row to the query.
12. From the Choose field list, select **Sys ID**.
13. In the same row as the **Sys ID** field in the condition builder, click the contextual reference value icon ( ).
14. Add the item parameter that you just created.
15. Click **Save**.

The completed data item should appear as it does in the following image.
Create an applet to navigate to

Before you create a navigation function, you need to have an applet to navigate to.

1. In Studio, navigate to Mobile Development > Application > ITSM.
2. Click the pop-out icon ( ) to open the ITSM application.
3. Click Create an applet.
4. From the Create an applet window, in the Name field, type List of callers.
5. In the Description field, type List of callers.
6. Select the List applet template.
7. Click New.
8. In the List of callers applet tab, from the Data Item list, select List of callers.
9. Above the Item preview example, click Change List Item.
10. In the List Item Patterns window, select the item preview in the last row of the second column.

11. Click Done.
12. In the Field Configuration field, move these fields to the Selected Fields list:
   - Department
   - Name
   - Avatar
   - Title

13. Switch to the Details tab of the list of callers applet.
14. Above the Selected Fields list, click Replicate from primary.
15. In the applet header, clear the Display toggle.

When you turn off Display, the applet does not appear on the application main page. Users can only access it from the field that you configure the navigation for in another applet.

16. Click Save.

A parameter setting section appears under the data item.

17. In the Parameter setting, in the Screen UI Parameter Mapping section, make sure that the following fields are completed with the correct values:
• Data Item Parameter: The name of the parameterized data item that you created, which in this case is caller_sys_ID.
• UI Parameter: The name of the UI parameter in the next section of the Parameter settings section. The name should be caller_sys_ID.

18. In the User input parameters section, click the first row in the table to open the User input parameter definition window.
19. Verify that the following fields are completed:
   • Name: caller_sys_ID
   • Input type: Text

20. Click Save.
21. In the List of callers applet, click Save.

Configure a navigation function with parameters
After you create an applet to navigate to, create a navigation function.
1. In Studio, navigate to Mobile Development > Functions > Navigations.
2. Click the pop-out icon ( ) to open the navigation function list.
3. Complete the fields as described in Configure a navigation function.
4. Click Save.
5. In the Parameter Setting section, click the first row in the table.
6. On the form, fill in the fields.

Redirection parameter fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination UI parameter</td>
<td>Field that is automatically filled with the name of the parameter created for the data item.</td>
</tr>
<tr>
<td>Type</td>
<td>Source for the information in the parameter. Choose from:</td>
</tr>
<tr>
<td></td>
<td>• Field: The parameter is filled with whatever information is in the field.</td>
</tr>
<tr>
<td></td>
<td>• Constant: A constant value. If you select this option, type a constant value in the field that appears.</td>
</tr>
<tr>
<td></td>
<td>• Source UI parameter: The source for the UI parameter. If you select this option, choose a UI parameter from the list.</td>
</tr>
<tr>
<td>Source field</td>
<td>Field that you want the user to navigate from. For example, the Callers field on the Incident (incident) table.</td>
</tr>
</tbody>
</table>

7. Click Save.
8. In the main navigation tab, click Save.

Associate the navigation function with an applet
After you create a navigation function, you need to assign it to an applet.
You can use any applet for these steps, including one from a previous tutorial. The first steps include instructions on how to create an applet. Make sure that the **Caller** field is included as part of the body on the details screen of the app. Otherwise, you won’t have a field to assign the navigation to.

1. Create a data item for open incidents.
   a) In Studio, navigate to **Mobile Development > Data Items**.
   b) Click the pop-out icon ( ) to open the data items list.
   c) In the **Data Items** tab, click **Create a new data item**.
   d) In the **Name** field, type **Open incidents**.
   e) In the **Table list**, search for **Incident**.
   f) In the **Query condition section**, use the `<State><is one of>` filter, and then select the **New**, **In progress**, and **On hold** fields.
   g) Click **Save**.

2. Create an applet for open incidents to assign the navigation function to.
   a) In Studio, navigate to **Mobile Development > Application > ITSM**.
   b) Click the pop-out icon ( ) to open the ITSM application.
   c) Click **Create an applet**.
   d) From the **Data Item** list, select **Open incidents**.
   e) Move these fields to the **List header list**.
      - Number
      - Priority
      - Short description
      - Assigned to
      - Assigned to > Avatar
      - State

      Add the **Assigned to > Avatar** field first and move it to the correct position before adding the **Assigned to** field.
   f) From the **Details** tab of the open incidents applet, above the **Selected Fields** list, click **Replicate from primary**.
   g) Click **Save**.

3. From the **Details** tab of the applet, in the **Field Configuration** field, select **Body**.
4. In the **Field Configuration** field, move the **Caller** field to the **Selected Fields** list.
5. Click **Save**.
6. In the **Details** tab, open the Functions section.
7. Under Field functions, click the **Add** icon ( ).
8. From the Field Functions window, in the Field list, select **Caller**.
   If you do not see a field here, you may need to save the applet again, or make sure that you added one to the body field of the applet.
9. In the Function list, select **Navigation > Jump to caller**.
10. Click **Done**.
Screen UI policies for the mobile app

Use screen UI policies to control which field is mandatory or visible in ServiceNow Agent. With screen UI policies, you can improve screen readability, which helps your users find the information that they need to do their tasks more efficiently.

Screen UI policies are similar to the UI policies that are used on forms in the instance, but screen UI policies are designed for the mobile app. Policies contain a set of conditions that you can use to determine when the policy applies. When a screen UI policy is triggered by a condition that you defined for the policy, all UI policy rules that are associated with that policy are applied.

Differences from UI policies for forms

Screen UI policies are similar to the UI policies that are used on forms with the following exceptions:

- Screen UI policies cannot control whether a field is read-only.
- Screen UI policy conditions cannot be scripted.
- Reference and date/time field types cannot be used in screen UI policy conditions.

Create a screen UI policy for the mobile app

Create screen UI policies to dynamically change which fields are mandatory or visible on screens in the ServiceNow Agent mobile app.

Role required: ui_policy_admin or admin

A screen UI policy condition evaluates all fields on a table even if they are not visible on the screen.

1. Navigate to System Applications > Studio.
2. In the Select Application pop-up window, select your mobile app.
3. In the Application Explorer pane, point to Mobile Development > UI Policies and click the pop-out icon.
4. On the Screen UI Policies list, click New.
5. On the form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the screen UI policy.</td>
</tr>
<tr>
<td>Short description</td>
<td>Short summary of the screen UI policy.</td>
</tr>
<tr>
<td>Order</td>
<td>Processing sequence, from the lowest to the highest number. If two policies conflict, the screen UI policy with the higher number executes. For inherited UI policies, the extended (child) table UI policies are executed first. Next, the base table UI policies are executed from the lowest to the highest specified value.</td>
</tr>
<tr>
<td>Table</td>
<td>Table for the screen that you want to modify,</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Condition</td>
<td>Conditions that must be fulfilled to apply this screen UI policy. Conditions are built with the condition builder. For details about this tool, see <a href="#">Condition builder</a>. The instance only rechecks if a user manually changes a field on a screen. If a UI action or context menu action makes a change, or a user makes a change from a list, the instance does not evaluate these changes.</td>
</tr>
<tr>
<td>Reverse if false</td>
<td>Option for specifying that the screen UI policy action should be reversed when the conditions of its policy evaluate as false. When the conditions are true, actions are taken and when they change back to false, the actions are reversed (undone).</td>
</tr>
<tr>
<td>Active</td>
<td>Option for enabling the screen UI policy.</td>
</tr>
</tbody>
</table>

**Use a screen UI policy when the state field is not On Hold**

This example shows how you can configure a screen UI policy to apply on screens that use data from the Incident (incident) table. You can build a condition that checks the values of fields on this table; in this case, the **State** field. When **(State) is not** (On Hold), the UI policy applies the associated UI policy rules. When you select the **Reverse if false** option, a field made invisible when the state field is not **On Hold** is made visible when the state is **On Hold**.
Create UI policy rules in the related list. The screen UI policy applies these rules when the data on a screen matches the policy conditions. For details on creating these rules, see [Create a UI policy rule](#).

Create a UI policy rule

Create UI policy rules to control whether the fields that your users see are visible or mandatory with the ServiceNow Agent mobile app.

Role required: ui_policy_admin or admin

1. Navigate to **System Applications > Studio**.
2. In the Select Application pop-up window, select your mobile application.
3. In the Application Explorer pane, point to **Mobile Development > UI Policies** and click the pop-out icon that appears.
4. Select the UI policy that you want to create rules for.
5. In the UI Policy Rules related list, click **New**.

© 2019 ServiceNow, Inc. All rights reserved.

ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.
6. On the form, fill in the fields.

**UI Policy Rule form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI policy</td>
<td>Screen UI policy that this rule is associated with. The field is automatically populated when you create rules from the UI policy rules related list on a screen UI policy form.</td>
</tr>
<tr>
<td>Field name</td>
<td>Name of the field to apply the rule to.</td>
</tr>
<tr>
<td>Visible</td>
<td>Options that change the visibility of a field on a screen. If the field is not available on the screen, you cannot configure a rule to make it appear.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>Options that change whether the field is mandatory.</td>
</tr>
</tbody>
</table>

- **Leave Alone**: Rule that has no effect on the visibility of this field.
- **True**: Rule that makes this field visible.
- **False**: Rule that hides this field on the screen.

**Use a UI policy rule to hide a field**

This example shows how you can control the visibility of the On hold reason field. On hold reason is selected in the Field name field, and the Visible field is set to False. If the screen UI policy conditions are met, the On hold reason field is hidden on the screen.
Mobile UI styles

UI styles let you change the color of fields in ServiceNow Agent.

Use UI styles to make some fields stand out. For example, you can configure different colors for the **Priority** field, depending on the status. Use brighter colors to make higher priority items stand out.

Select colors that are high contrast so that users can still easily make out the text.

Create mobile UI styles

Create UI styles for applet fields in ServiceNow Agent to improve readability and highlight important information.

Create an app and applets before you configure UI styles.
Role required: admin

1. Navigate to System Applications > Studio.
2. From the Select Application window, select your mobile app.
3. Next to UI Styles, click the pop-out icon ( ) that appears when you point to the folder name.
4. Click New.
5. On the form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for the UI style. Select a name that is easily identifiable. You will use this name later when you apply the style to a specific field in an applet.</td>
</tr>
<tr>
<td>Table</td>
<td>Name of the table that contains the field that you want to add a style to.</td>
</tr>
<tr>
<td>Condition</td>
<td>Conditions in which you want the field style to appear. For example, you can configure the State field to appear blue, but only when State is Assigned.</td>
</tr>
<tr>
<td>Field name</td>
<td>Name of the field that you want to add the style to. Only the fields that are available on the table that you selected appear in this list.</td>
</tr>
<tr>
<td>Style</td>
<td>Colors that you want on the field. Make sure that the contrast between the background color and the font color is enough so that users can still see the value.</td>
</tr>
<tr>
<td>Active</td>
<td>Whether or not the UI style is active.</td>
</tr>
</tbody>
</table>

6. Click Submit.
7. Repeat this process for each condition that you want the UI style to appear for.

Add the field style to the field in an applet.

Add a UI style to an applet

Add a UI style to a field to change that field's appearance.

1. In Studio, navigate to Mobile Development > Applications, and then click the expand arrow icon ( ) to see the list of applets for your app.
2. Click Create an applet or open an existing applet that you want to add UI styles to.
   The data item for the applet should match the table that you selected for the UI style.
3. From the screen configuration page for the applet, add the field that you want to add styles to, in the primary screen or details screen header.
   Only certain field locations are in the header support UI styles. Make sure that you add the field to a position that allows UI styles. If the field is not in a position that allows UI styles, the UI Style Configuration section does not display any options.
   For example, in the following header preview, only the field positioned at E1 allows UI styles.
4. In the **UI Style Configuration** field, move the styles that you want to appear for the field to the Selected Styles list.

5. Click **Save**.

**Mobile push notifications**

Set up push notifications in ServiceNow Agent, so that your users can see important messages about record assignments, approvals, and changes to their assigned records.

To set up push notifications for the mobile app, you must perform the following steps:

- In Studio, create an app and an applet to apply a push notification to.
- In the platform, configure the push notification message and the push notification.

**Note:** Push notifications are not supported on on-premise instances.

This image shows how push notifications move from your instance to the mobile devices of your users.
ServiceNow datacenter

Admin creates and app and app

ServiceNow instances

ServiceNow feedback and push proxies

Push notifications structure

Admin creates/updates push messages and notifications for SN mobile app
Set up push notifications for the mobile app

Configure push notifications for ServiceNow Agent mobile app in the System Notification section of the platform. Setting up a push notification includes setting up the message and setting up the notification.

Role required: admin

1. Get the document ID for the applet that you want to create notifications for. The document ID is the sys_id of the primary screen for that applet.
   a) Navigate to System Mobile > Mobile Applications.
   b) Open the application record that contains your applet.
   c) In the Folders related list, click the folder where you created your applet.
      If you do not have any defined folders, you see a single folder called (empty).
   d) In the Screens related list, right-click the entry for your applet, and select Copy sys_id to copy the sys_id to your clipboard.

2. Create a message for the push notification.
   a) Navigate to System Notification > Push > Push Message Content.
   b) Click New.
   c) In the Name field, create a name for the push notification message. Select a name that you can easily identify later.
   d) From the Push app field, select ServiceNow Mobile application.
   e) In the Push Message field, enter a JSON payload with the content that you want to appear in the push notification.
      For example, you can use this code block. Enter the sys_id for your applet from the previous steps for the value in the DocumentId line.

```javascript
(function buildJSON(/*GlideRecord*/ current, /*String*/ message, /*Object*/ attributes) {
    var json = {};
    json = {
      "aps" : {
        "sound": "default"
      },
      "Redirection" : {
        "Title": "Work Order Task Acceptance Push Notification",
        "To": "Embedded",
        "Destination": {
          "DocumentId": "<sys_id of your Master-detail screen here>",
          "ItemId": current.sys_id
        }
      },
      "Layout": {
        "Identifier": current.number,
        "Description": current.short_description,
        "AdditionalInfo": current.location.getDisplayValue(),
        "StatusImage": current.priority_image + ".iix?t=small"
      }
    };
    return json;
```

Where <document_id> is the sys_id of your Master-detail screen.

f) Click **Submit**.

3. Create the push notification message.
   a) Navigate to **System Notification > Push > Push Messages.**
   b) Click **New.**
   c) On the form, fill in the fields.

### Push Notification Message form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for the notification message.</td>
</tr>
<tr>
<td>Push App</td>
<td>Push application used with this notification. Select <strong>ServiceNow Mobile Application.</strong></td>
</tr>
<tr>
<td>Push Message Content</td>
<td>Push message content that was created in the previous step.</td>
</tr>
<tr>
<td>Table</td>
<td>Table that the notification appears for. Ensure that you select the same table that you did for the applet that you are creating the notification for.</td>
</tr>
<tr>
<td>Message</td>
<td>Message that appears to the users who receive the notification.</td>
</tr>
</tbody>
</table>

4. Create the push notification.
   a) Navigate to **System Notification > Push > Push Notifications.**
   b) Click **New.**
   c) On the form, fill in the fields.

For more detailed information on push notifications, see **Push notifications.**

### Push Notification form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for the notification</td>
</tr>
<tr>
<td>Active</td>
<td>Option to make this notification active.</td>
</tr>
<tr>
<td>Table</td>
<td>Table that the notification appears for. Ensure that you select the same table that you did for the applet that you are creating the notification for.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Category</td>
<td>Category to which this notification belongs. A category identifies and groups related notifications. This notification, if active and subscribable, is listed in the selected Category in the notification preferences for each user (Notifications tab in the System Settings window). <strong>Note:</strong> Do not leave the category as Uncategorized, as users may not be able to find the notification in their list of notifications. If you need a new category, see Create notification categories.</td>
</tr>
<tr>
<td>Description</td>
<td>Short description of the push notification.</td>
</tr>
<tr>
<td>When to send</td>
<td></td>
</tr>
<tr>
<td>Inserted</td>
<td>Notification that is sent when a record is inserted.</td>
</tr>
<tr>
<td>Updated</td>
<td>Notification that is sent when a record is updated.</td>
</tr>
<tr>
<td>Conditions</td>
<td>Conditions under which this notification is sent. For example, select Priority &gt; greater than &gt; 3 to send the notification only for High and Critical priority incidents.</td>
</tr>
<tr>
<td>Who will receive</td>
<td></td>
</tr>
<tr>
<td>Users</td>
<td>Users who receive the messaging notification.</td>
</tr>
<tr>
<td>Users/Groups in fields</td>
<td>Users or groups from reference fields. For example, if a notification uses the Incident (incident) table, you can select users or groups from incident fields like Opened by and Assignment group. <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. If you address the notification to a user with an inactive record in the User (sys_user) table, the system does not send the notification to that user.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groups</td>
<td>Groups to receive the messaging notification. You can search for groups with the reference lookup icon or manually enter the group name. This list of groups is static.</td>
</tr>
</tbody>
</table>

**What to send**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push Messages</td>
<td>Name of the push message that you created in step 3c.</td>
</tr>
</tbody>
</table>

**d)** Click **Submit**.

**5.** Add your notification to the push default registration  
**a)** Switch to the **Global** scope.  
**b)** Navigate to **System Notification > Push Application**.  
**c)** Open the ServiceNow Mobile Application record.  
**d)** Right-click the record header and select **Configure > Related Lists** from the menu.  
**e)** Add **Push Default Registration->Push App** to the **Selected items** list.  
**f)** In the Push Default Registrations related list, click **New**.  
**g)** In the **Notification** field, select the notification that you created in the previous steps.  
**h)** Click **Submit**.

A notifications folder appears in your mobile scoped app. Your push message is rendered on mobile devices that it is registered to. To see how push notifications appear to your users, see [Manage notifications from your mobile device](#).

---

**Note:** After you create or modify a notification, your users need to log out and log in again for the changes to take effect.

---

**Mobile offline mode**

Offline mode lets users who have no internet connection continue working from a mobile device. Configure specific application, applets, or functions for users to use offline.

Watch this three-minute video to learn how offline mode works, and how your users can download data, sync an outbox, and resolve sync errors.

**Request offline mode**

Request the SG Offline support plugin (com.glide.sg.offline) through the HI Customer Service system to activate offline mode for ServiceNow Agent.

**Role required:** admin

**Note:** Offline mode is supported on on-premise instances.

**1.** In the HI Service Portal, click **Service Requests > Activate Plugin**.

---

© 2019 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.
2. On the form, fill in the fields.

### Plugin activation form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Instance</td>
<td>Instance on which to activate the plugin.</td>
</tr>
<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 that 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 time zone, 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 | Information that would be helpful for the ServiceNow personnel activating the plugin. For example, if you need the plugin activated at a specific time instead of during one of the default activation windows. |

3. Click Submit.

### Configure offline mode behavior

Customize the offline mode behavior in ServiceNow Agent to fit your organization's needs.

Install or request installation of the SG Offline support plugin (com.glide.sg.offline).

Role required: admin

1. In the application navigator, enter `sys_properties.list`.
2. In the Name search box, enter the name of a property from the table below, and press Enter to display the property record.
3. Set values to the property record according to how you would like offline mode configured.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.sg.offline.attachment.max_size</td>
<td>Maximum size, in bytes, for attachments that are cached in offline mode. If no download occurs, a placeholder appears. The default size is 50 MB.</td>
</tr>
<tr>
<td>glide.sg.offline.expiration</td>
<td>Length of time before the cached data expires. The default length is 48 hours. After 48 hours, the system deletes the data due to a security protocol.</td>
</tr>
<tr>
<td>glide.sg.offline.attachment.allowed_content_types</td>
<td>Comma-separated list of allowed file types for attachments in offline mode. The default list includes these types: image/png, image/jpg, image/gif, video/quicktime. The default is empty. If empty, no files are downloaded.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.sg.offline.roles</td>
<td>Comma-separated list of role names that are allowed to work in offline mode. If empty, all users are allowed.</td>
</tr>
<tr>
<td>glide.sg.offline.enabled</td>
<td>Offline capabilities that are enabled on your instance. Offline mode is enabled by default. To disable offline mode, create a system property with this name and set the value to false.</td>
</tr>
</tbody>
</table>

4. Repeat steps 2 and 3 for the remaining properties in the table.

To see an example of what it’s like to work in offline mode, see *Working offline in mobile*.

Configure offline mode for applications or applets

Configure applications and applets to be available in ServiceNow Agent for users that are not connected to the internet.

Because applications contain applets, if you set an applet to work offline, you must also set the application to work offline.

Role required: admin

1. In *Studio*, navigate to the application that you want to configure offline mode for, and then click the pop-out icon (改编) that appears when you point to the application name.
2. Optional: Open an applet that you want to let users use offline.
3. In the header of the application or the applet, click *Properties*.
4. Use the Available offline toggle switch to determine if the application or applet is available offline.

Determine which functions are available to users in offline mode.

Configure offline properties for action functions

Determine which fields and functions are available to users who access the mobile app in offline mode.

Role required: admin

1. In *Studio*, navigate to *Mobile Development > Functions > Actions*.
2. Click the pop-out icon (改编) to open the Actions list.
3. Open an action that you want to configure for offline mode.
4. Select the *Offline* option.
5. In the *Hide field* and *Show field* fields, move field names to the Selected list to determine which fields are available after the user performs the action in offline mode. For example, after a user assigns a task to themselves, you could hide the *Assigned to* field or show the *Work notes* field.
6. Use the *Hide functions* and *Show functions* fields, add functions to the lists to determine which functions are available after the user performs the action in offline mode. For example, when a user taps the Start Work function in offline mode, that action function is hidden and the Close Complete and Close Incomplete functions are displayed instead.
7. From the *Mark as zombie on screens* field, search for screens to gray out when a user performs an action.

© 2019 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.
In offline mode, an indicator appears next to a record after a user makes a change, as seen in the following image. A screen goes into zombie mode when a record is grayed out in the list after a user makes a change.
INC0000057 has successfully been updated.

Unassigned incidents

INC0000039  5 - Planning
Trouble getting to Oregon mail server
Assigned to  
State  New

INC0000057  5 - Planning
Performance problems with wifi
Assigned to  Beth Anglin
State  In Progress

INC0010003  1 - Critical
Need access to sales DB for the West
Assigned to  
State  New

INC0000059  3 - Moderate
Unable to access team file share
Assigned to  
State  New

Online behavior for zombie mode

© 2019 ServiceNow, Inc. All rights reserved. ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.
8. Click **Update**.

You can also configure offline mode for **navigation** and **smart button** functions.

**Mobile authentication**

Users are required to log in to an instance on their mobile device. Depending on how you configure authentication for mobile devices, users may be required to enter additional information.

For more information on configuring authentication for mobile devices, see [Set up OAuth](#).

Follow the instructions for using a third-party OAuth provider.

**Create a QR code for mobile login**

Create and use a QR code containing JSON to provide a method for your users to log in with pre-defined parameters.

**Role required:** none

1. Use a text editor to create JSON using the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuthenticationUrl</td>
<td>The URL used for authentication</td>
</tr>
<tr>
<td>ClientId</td>
<td>The client ID string used for OAuth 2.0</td>
</tr>
<tr>
<td>ClientSecret</td>
<td>The client ID string used for OAuth 2.0</td>
</tr>
<tr>
<td>InstanceUrl</td>
<td>The URL for your instance. For example, <a href="https://hi.servicenow.com">https://hi.servicenow.com</a></td>
</tr>
<tr>
<td>Name</td>
<td>The name of your instance. For example, Hi.</td>
</tr>
<tr>
<td>PublisherUrl</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td></td>
</tr>
</tbody>
</table>

The JSON consists of parameters and values, separated by commas, and enclosed in curly braces. Each parameter is separated from its value by a colon, and each parameter and value is enclosed in double quotes, as shown in this example.

```json
{
  "Name":"Hi",
  "InstanceUrl":"https://hi.servicenow.com"
}
```

**Note:** Parameters are not supported for the Android OS. To use a QR code for Android, create a QR code containing only the URL for the instance. For example, [https://hi.servicenow.com](https://hi.servicenow.com).

2. Use a QR code generator of your choice to encode this JSON into a QR code.

3. Use your QR code to access the mobile app. For steps to use a QR code with the app, see [Log in to an instance with the mobile app](#).
Configure mobile app session timeout

The native mobile app times out after a certain amount of inactivity. Sessions are considered active if the app is in the foreground or if the app is processing a long running task in the background. Configure the length of time it takes for the app to time out.

Role required: admin

The app is considered inactive in any of the following scenarios.

- Explicitly sending the app to the background.
- Locking the screen or having the screen go to sleep.
- Switching to a different app.

Any of the following states are considered actively running:

- The app is visible on the screen and the screen is unlocked.
- The app is processing a long running task even if the app is in the background, such as uploading or downloading a large attachment.

The greater of all of the following settings determine mobile app session time-out.

These instructions apply to both the ServiceNow mobile app and the ServiceNow Classic mobile app. Make sure you select the correct application depending on which app you are doing configuration for.

1. Configure the OAuth entity for the mobile app.
   a) Navigate to System OAuth > Application Registry and open the ServiceNow Mobile App OAuth entity. If your configuration is for the Classic app, select that OAuth entity instead.
   b) In the Refresh Token Lifespan field, change the number in seconds to the amount of time the refresh token is valid.
   c) In the Access Token Lifespan field, change the number in seconds to the amount of time the access token is valid.

2. Configure the web session time-out property.
   a) In the navigation filter, type sys_properties.list.
   b) Search for the glide.ui.session_timeout property.
   c) Change the value in minutes to the amount of time before a session times out. The default value is 30 minutes.

3. Configure the integration session time-out property.
   a) In the navigation filter, type sys_properties.list.
   b) Search for the glide.integration.session_timeout property. If the property does not exist, create a system property.
   c) Configure the property with the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>glide.integration.session_timeout</td>
</tr>
<tr>
<td>Type</td>
<td>integer</td>
</tr>
</tbody>
</table>
For example, if you want the mobile app to sign out after 30 minutes of inactivity, use the following configuration.

<table>
<thead>
<tr>
<th>Configuration point</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh Token Lifespan for the ServiceNow mobile app</td>
<td>1800 seconds</td>
</tr>
<tr>
<td>Access Token Lifespan for the ServiceNow mobile app</td>
<td>1800 seconds</td>
</tr>
<tr>
<td>glide.ui.session_timeout system property</td>
<td>30 minutes</td>
</tr>
<tr>
<td>glide.integration.session_timeout system property</td>
<td>Not defined, or 30 minutes or less.</td>
</tr>
</tbody>
</table>

Sign out and sign back in to the mobile app. Otherwise the app uses the previously granted sessions and tokens.

Mobile geolocation tracking configuration

Activate geolocation tracking for your ServiceNow Agent mobile app users by installing the Geolocation plugin (com.snc.geolocation). Complete other geolocation tracking tasks to finalize the configuration.

When you activate the Geolocation plugin, the system automatically adds the Geolocation tracked option to the User form.

After this permission is granted, your users are prompted to enable location tracking on their devices the first time that they access it.

To complete the remaining geolocation tracking configuration tasks, see Geolocation

Additional mobile app configuration options

Several system properties are available for you to further configure the mobile app. For example, use system properties to require a PIN, hide the image on the app homepage, configure the blur in background option, or disable sharing attachments from the mobile app.

Set a default image for mobile users

To provide a default image to users whose records do not already have an image uploaded, configure the system property glide.sg.image.default.sys_user.

Role required: admin

If you do not upload images to user records, users in the mobile app receive an avatar by default. The file name for the avatar is no_picture.jpg.
To set a different default user image, upload an image and then enter its file name as the value for system property `glide.sg.image.default.sys_user`.

1. If you have not already uploaded the image that you intend to set as the default user image, navigate to `System UI > Images` and then upload your file to the Images (db_image) table.
2. Type `sys_properties.list` in the Application Navigator.
3. Open the record for `glide.sg.image.default.sys_user`.
4. In the form, match the following values:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td><code>glide.sg.image.default.sys_user</code></td>
</tr>
</tbody>
</table>
Hide the hero image on the application homepage

To hide the default hero image that appears on the application homepage, add the system property `glide.sg.hide_applications_screen_image`.

Role required: admin

By default, the mobile app displays a banner across the top of the application homepage. This banner is called the hero image.
1. Type `sys_properties.list` in the Application Navigator.
2. Open the record for `glide.sg.hide_applications_screen_image`.
3. In the form, match the following values:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td><code>glide.sg.hide_applications_screen_image</code></td>
</tr>
<tr>
<td>Type</td>
<td>true</td>
</tr>
<tr>
<td>Value</td>
<td>true</td>
</tr>
</tbody>
</table>

After you reload the mobile app, the hero image does not appear on the application homepage.
Require an app PIN for the mobile app

To require the mobile user to set and enter a local application PIN, add the system property `glide.sg.require_mobile_application_pin`.

Role required: admin

Users generate a six-digit code for the app PIN and enter it for each time they log in to an instance from their mobile device.
1. Type `sys_properties.list` in the Application Navigator.
2. Open the record for `glide.sg.require_mobile_application_pin`.
3. In the form, match the following values:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td><code>glide.sg.require_mobile_application_pin</code></td>
</tr>
<tr>
<td>Type</td>
<td>`true</td>
</tr>
<tr>
<td>Value</td>
<td><code>true</code></td>
</tr>
</tbody>
</table>

**Note:** The mobile app is automatically locked after 5 minutes of inactivity. Users accessing the mobile app after a period of inactivity need to enter their PIN code.
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. 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. Type `sys_properties.list` in the Application Navigator.
2. Open the record for `glide.sg.blur_ui_when_backgrounded`.
3. In the form, match the following values:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td><code>glide.sg.blur_ui_when_backgrounded</code></td>
</tr>
<tr>
<td>Type</td>
<td>`true</td>
</tr>
<tr>
<td>Value</td>
<td><code>true</code></td>
</tr>
</tbody>
</table>

**Note:** The system property `glide.sg.blur_ui_when_backgrounded` is supported also in the ServiceNow Classic mobile app.

With the property in place, the app is blurred when not in focus.
Configure the mobile app to clear the copy/paste clipboard when moved to background

To have the mobile app clear the pasteboard when the app enters the background, add the system property `glide.sg.clear_pasteboard_when_backgrounded`.

Role required: admin
By default, content that you copy in the mobile app remains on your device clipboard even when you close the mobile app. The system property `glide.sg.clear_pasteboard_when_backgrounded` deactivates this default setting.

1. Type `sys_properties.list` in the Application Navigator.
2. Open the record for `glide.sg.clear_pasteboard_when_backgrounded`.
3. In the form, match the following values:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td><code>glide.sg.clear_pasteboard_when_backgrounded</code></td>
</tr>
<tr>
<td>Type</td>
<td>`true</td>
</tr>
<tr>
<td>Value</td>
<td><code>true</code></td>
</tr>
</tbody>
</table>

Configure the maximum number of records returned for data items

To set the maximum number of rows retrieved for the table defined in the data item, add the system property `glide.sg.data_item.row_count`.

Role required: admin

The system property `glide.sg.data_item.row_count` sets the maximum number of rows retrieved for the table defined in the data item. By default, the value is 1000. The system accepts no value greater than 10000.

1. Type `sys_properties.list` in the Application Navigator.
2. Click New, and then enter the following values:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td><code>glide.sg.data_item.row_count</code></td>
</tr>
<tr>
<td>Type</td>
<td><code>integer</code></td>
</tr>
<tr>
<td>Value</td>
<td><code>&lt;maximum-number-of-records-retrieved&gt;</code></td>
</tr>
</tbody>
</table>

Configure pagination size for search lists

To configure the amount of search list results that load to the screen as the user scrolls down, add the system property `glide.sg.list.pagination_size`.

Role required: admin

By default, the mobile app returns 50 results at a time in a search list. Increasing the pagination size may cause the search request to take longer.

1. Type `sys_properties.list` in the Application Navigator.
2. Open the record for `glide.sg.list.pagination_size`.
3. In the form, match the following values:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td><code>glide.sg.list.pagination_size</code></td>
</tr>
</tbody>
</table>
Configure the maximum number of records returned for list UI parameters

To configure the maximum number of records returned for a list of parameters, add the system property `glide.sg.list.max_items_number`.

Role required: admin

Default is 1000.

**Note:** The maximum number of rows returned for a parameters list is limited by the maximum number of rows returned for data items. In other words, the value for system property `glide.sg.list.max_items_number` cannot be greater than the value for system property `glide.sg.data_item.row_count`. For example, if you set the value of `glide.sg.list.max_items_number` to 50 but the value of `glide.sg.list.max_items_number` is 20, then you may only receive 20 records in your parameters list.

1. Type `sys_properties.list` in the Application Navigator.
2. Open the record for `glide.sg.list.max_items_number`.
3. In the form, match the following values:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td><code>glide.sg.list.max_items_number</code></td>
</tr>
<tr>
<td>Type</td>
<td>string</td>
</tr>
<tr>
<td>Value</td>
<td><code>&lt;maximum-number-of-parameters-items&gt;</code></td>
</tr>
</tbody>
</table>

Mobile app configuration in the platform

Most mobile app configuration should take place in the ServiceNow Studio, however a few more advance configuration options are only available in the lists and forms view.

Studio has more precautions in place that prevent you from configuring anything that will break the mobile app. The majority of app configuration should take place in Studio. Use the platform for specific advanced configurations. With a few exceptions, the layout in Studio mimics the form layout in the platform, therefore the instructions for creating each of the components for the mobile app should be similar for lists and forms as they are for Studio.

However, if you require a more advanced customization, the platform form provides a few scripting options. Use the platform form view to configure the following components:

- Scripted data item
- Carried parameters

To learn more about building map and list apps, watch these two videos.
Configure a data item in the platform

Data items represent information from a specific table. Most screens, such as a list, map, grouped list, or calendar screen, require a data item to display information. Configure data items to associate with a screen.

Studio has more precautions in place that prevent you from configuring anything that will break the mobile app. The majority of app configuration should take place in Studio. Use the platform for specific advanced configurations. For example, scripted data items.

Role required: admin

1. Navigate to **System Mobile > Mobile Data Items**.
2. Click **New**.
3. Complete the following fields as needed.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A title for the data item. You can have multiple data items with the same name. Make sure this name is unique so that you can find the data item you are looking for when you associate it with a screen.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional additional details about the data item</td>
</tr>
<tr>
<td>Table</td>
<td>The name of the table you want the data item to pull information from.</td>
</tr>
<tr>
<td>Condition type</td>
<td>Determine whether the condition for the data item is declarative or scripted. Use a declarative condition to create conditions for the data item using the condition builder. Use a scripted conditions to create a scripted data item.</td>
</tr>
<tr>
<td>Query Condition</td>
<td>Set of filter conditions for the data item to conform to, for example, active incidents that are only in a state of New, In progress, or On hold. This option only appears if you select Declarative as the condition type. Use the condition builder to add a filter to a list, or add a parameter to the condition. Parameters act as variables that allow the system to ask for additional information before querying the database. If you want to add a parameter to the condition builder, use the Item parameters belonging to data item related list to create one. Then use the contextual reference value icon ( ) to select the parameter. For example, if you create a parameter for the priority field, your condition would be something like Priority is &lt;priority&gt;.</td>
</tr>
</tbody>
</table>

© 2019 ServiceNow, Inc. All rights reserved.

ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query condition script</td>
<td>Scripted condition for the data item. This field only appears if you select <strong>Scripted</strong> as the condition type.</td>
</tr>
</tbody>
</table>

**Note:** The Query condition script field is not visible in Studio. To create a scripted condition, follow the optional steps listed below.

<table>
<thead>
<tr>
<th>Group by</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The field your items are grouped by. This field is only if you are creating a grouped list. For example, you could group open incidents by priority.</td>
</tr>
</tbody>
</table>

4. Optional: In the Item parameters belonging to data item related list, create a parameter to pass in to the query condition.

5. Optional: Configure a scripted condition for your data item.
   a) Create your data item as described above, and save the item.
   b) Navigate to **System Mobile > Mobile Data Items**.
   c) Open the data item record create in previous steps.
   d) Change the **Condition type** field to **Script**.
   e) Enter your script in the **Query Condition Script** field.

A scripted condition must return a query string, which the instance uses to filter the data item. In this example, the query filters for records that match the company and location of the input record. Note that in each case, the script uses an if statement to make sure the input has a value in that field. This check prevents the query from filtering using empty fields. Before the returning the query, the script includes a **ORDERBYDESC** command to sort using the **sys_updated_on** field.

```javascript
(function getQueryString(input) {
  var queryString = ''; 
  if(input.company) {
    queryString += 'company='+input.company;
  }
  if(input.location) {
    queryString += '^location='+input.location;
  }
  queryString += '^EQ^ORDERBYDESCsys_updated_on';
  return queryString;
})(input);
```

Configure a parameter for a data item in the platform

Add parameters to a data item to allow the system to look for more information before querying the database.

Role required: admin

1. From the data item you want to add a parameter to, in the Item Parameters belonging to data item related list, click **New**.
2. Complete the following fields as needed.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the parameter. For example, for the sysID asset data item, you could create a parameter called Asset_SysID.</td>
</tr>
<tr>
<td>Parent table</td>
<td>The table the parameter is being created for. This field is automatically populated with the data item table (sys_sg_data_item).</td>
</tr>
<tr>
<td>Parent</td>
<td>The data item the parameter is for. This field is automatically populated with the name of the data item.</td>
</tr>
<tr>
<td>Type</td>
<td>Determines how the user interacts with the field. Most data item parameters have a type of String, but you can use any of the following. • String: Opens a full keyboard • Integer: Opens a numbers-only keypad • Decimal: Opens a numbers-only keypad • Boolean: Opens a true or false selection option • DateTime: Opens a calendar with an exact time selector • Date: Opens a calendar</td>
</tr>
<tr>
<td>Default value</td>
<td>The default value to be used in the parameter in case no value comes from a UI parameter.</td>
</tr>
</tbody>
</table>

3. Click Submit.

Create a mobile application for the mobile app in the platform

Applications are the highest level in a mobile app. They contain folders, screens, and anything else users need to find information in the app. To get started configuring the mobile app, first create an application.

Studio has more precautions in place that prevent you from configuring anything that will break the mobile app. The majority of app configuration should take place in Studio. Use the platform for specific advanced configurations.

Role required: admin

1. Navigate to System Mobile > Mobile Applications.
2. Click New.
3. Complete the following fields as needed.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the application, for example, ITSM. This name appears in the mobile app.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Description</td>
<td>Additional information about the application. The text you add to this field also appears in the mobile app. Users can tap the i icon to see more information about the application.</td>
</tr>
<tr>
<td>Icon</td>
<td>An image to represent the application on the app homepage.</td>
</tr>
<tr>
<td>Roles</td>
<td>List of roles that have access to the application. If you do not select any roles, all roles can access the application.</td>
</tr>
<tr>
<td>Order</td>
<td>The order in which the application displays on the app homepage. The lower the number, the earlier in the app the application appears.</td>
</tr>
<tr>
<td>Active</td>
<td>Whether or not the application is active in the mobile app. Applications not marked as Active are not visible in the mobile app.</td>
</tr>
<tr>
<td>Available offline</td>
<td>Whether or not the application is available offline. For more information on configuring applications for offline mode, see <a href="#">Configure offline mode behavior</a>.</td>
</tr>
</tbody>
</table>

4. Click **Submit**.

The application appears in the mobile app on the homepage.
Applications in the mobile app
Create a folder to contain screens for your mobile application.

Create a folder for the mobile app

Folders help you organize applets in an application. Create a folder to contain the applets for your application.

Studio has more precautions in place that prevent you from configuring anything that will break the mobile app. The majority of app configuration should take place in Studio. Use the platform for specific advanced configurations. Folders are not required to contain applets. In Studio you can create a folder on the application main page. From the platform view, you can create folders from the related list within an application.

Role required: admin

1. Navigate to System Mobile > Mobile Applications.
2. From the application list, select the application you want to create a folder for.
3. In the Folders related list, Click New.
4. Complete the following fields as needed.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for the folder, for example, Incident. Content in this field appears within the application with the list of screens. Folders are not required to have a name. Screens belonging to an unnamed folder are grouped together and do not have a title.</td>
</tr>
<tr>
<td>Application</td>
<td>The application you are creating the folder within. Make sure the application you created the folder from is selected.</td>
</tr>
<tr>
<td>Order</td>
<td>The order you want the folder to appear. You can have multiple folders on a page. The lower the order number, the higher on the page the folder appears.</td>
</tr>
</tbody>
</table>

5. Click Save.
   The Screens belonging to folder related list appears.

Create screens for the folder to contain.

Configure a mobile screen in the platform

Screens contain information specific to the folder or master item containing them. For example, a folder called Incidents could have screens for Open Incidents, P1 Incidents, and Unassigned incidents. Each of those screens contains additional screens with more information such as screen details or an activity stream.

Studio has more precautions in place that prevent you from configuring anything that will break the mobile app. The majority of app configuration should take place in Studio. Use the platform for specific advanced configurations.

Applets are made up of a series of screens. You cannot see the applet view from the platform, only from Studio. Use Studio to avoid any issues with configuration.
Role required: admin

1. Navigate to **System Mobile > Mobile Applications**.
2. Open the application you want to create screens for. For example, ITSM.
3. From the Folders related list for the application, open a folder you want to create screens for, for example, Incidents.
   Folder names are not required. If you leave the folder name empty, all screens without a folder are grouped together. The folder name then appears as (empty).
4. In the Screens belonging to Folder related list, click **New**.
5. From the Select Screen Type list, select a type of screen to create.

6. In the screen record, complete the fields as needed. For more information on configuring a specific screen, see [Mobile screen configuration by screen type](#)

Configure the fields that appear on a screen.

**Mobile screen configuration by screen type**

Configuration for each screen varies by the type of screen. Some screens require embedded screens to appear properly in the mobile app. To ensure screens are configured properly, use [Studio](#) for configuration.

**Master-detail screen fields**

The master-detail screen contains a master screen (with the list view) and at least one embedded screen – usually a Details screen. The data item associated with the master-detail screen determines the information that appears in the list.

Having at least one embedded screen for the master-detail screen is not enforced in the platform configuration view. Use [Studio](#) for a more secure configuration experience. If you want to create a list view without any embedded screens, create a list screen instead.
Open Incidents

1 - Critical  INC0000015
I can't launch my VPN client since the last software update
Open Date  2018-04-01 16:38:46
Assigned To  Don Goodliffe

1 - Critical  INC0000016
Rain is leaking on main DNS Server
Open Date  2018-03-27 16:40:23
Assigned To  ITIL User

1 - Critical  INC0000017
How do I create a sub-folder
Open Date  2015-08-12 16:41:00
Assigned To  Fred Luddy

1 - Critical  INC0000018
Sales forecast spreadsheet is READ ONLY
### Master-detail screen fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the master-details screen. This name is a user facing field.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional additional details about the master-screen. Users can see the description in the mobile app if they tap the information icon for the applet.</td>
</tr>
<tr>
<td>Parent table</td>
<td>Automatically populated with the master item (<code>sys_sg_master_item</code>) table.</td>
</tr>
<tr>
<td>Parent</td>
<td>Automatically populated with the name of the folder or master item the screen was created from. This field is how the screen gets associated with a folder or master item.</td>
</tr>
<tr>
<td>Application</td>
<td>Automatically populated with the current application the record was created in.</td>
</tr>
<tr>
<td>Order</td>
<td>The order in which the screen appears. The lower the number the earlier it appears on a screen.</td>
</tr>
<tr>
<td>Icon</td>
<td>The image that appears to represent the screen. This field is required.</td>
</tr>
<tr>
<td>Hidden</td>
<td>Whether or not users can see the screen from the mobile homepage. You can hide screens from the homepage that you only want users to access from fields on a form.</td>
</tr>
<tr>
<td>Data Item</td>
<td>Defines what information is being pulled into the screen. For more information on configuring a data item, see <a href="#">Configure a data item in the platform</a>.</td>
</tr>
<tr>
<td>Available offline</td>
<td>Whether or not the screen is available in offline mode.</td>
</tr>
</tbody>
</table>

After you click **Save**, additional related lists appear. The master-detail screen requires at least one embedded screen. To embed a Details screen, do the following.

1. In the Master Item related list for the screen, click **New**.
   
   Master items are containers for additional screens such as record details, activity stream, and related lists. Only the items contained by the master item appear on a mobile device.

2. Complete the fields as needed.

### Master item record fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the master item. You can have multiple master items with the same name so make sure this name is unique. This name does not appear on a mobile device.</td>
</tr>
<tr>
<td>Screen</td>
<td>The screen the master item is related to. This field is automatically populated with the name of the screen you created the master item from.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>The table you are creating items in the master item for. This field is automatically populated with the table of the screen you created the master item from. If the data item you are using for the screen that the master item belongs to is a parent table, you can select child tables from the list.</td>
</tr>
<tr>
<td>Condition</td>
<td>Conditions or filter for the master item. For more information on using the condition builder, see <a href="#">condition builder</a></td>
</tr>
<tr>
<td>Condition order</td>
<td>Determine the order that the conditions for a master item execute. This field is only used if the server identifies multiple master items with conditions that contradict each other.</td>
</tr>
<tr>
<td>Item view</td>
<td>The appearance of the items contained by the master item. The item view contains JSON that determines how fields appear on a screen.</td>
</tr>
</tbody>
</table>

3. From the context menu, click **Save**.

4. From the Screen belonging to Master Item related list, click **New**.

5. From the Select Screen Type list, select a screen.
   - Choose from
     - Activity Stream Screen
     - Details screen
     - Related list screen

6. In the screen record, complete the fields as needed. Available fields vary by screen type.
   - For example, if you have a master item for open incidents, create three screens for users to navigate between: details, activity stream, and related lists. When users open an incident from the Open Incidents list, they can swipe between each of the screen options.

### List (group by) screen

The list screen shows a list of items grouped by a particular field, for example, Priority. The list screen does not require any embedded screens. When you tap an item, you are taken to another screen where you can select the field to group by.
# My Group Incidents

Grouped by category

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>1</td>
</tr>
<tr>
<td>Inquiry / Help</td>
<td>3</td>
</tr>
<tr>
<td>Network</td>
<td>1</td>
</tr>
<tr>
<td>Software</td>
<td>5</td>
</tr>
</tbody>
</table>
List screen fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the details or list screens. Name is a user facing field.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional additional details about the detail or list screens. Information in the description field appears in the mobile app if the user taps the information icon for the applet.</td>
</tr>
<tr>
<td>Parent table</td>
<td>Automatically populated with the name of the folder or master item the screen was created from.</td>
</tr>
<tr>
<td>Parent</td>
<td>Automatically populated with the name of the folder or master item the screen was created from. This field is how the screen gets associated with a folder or master item.</td>
</tr>
<tr>
<td>Application</td>
<td>Automatically populated with the current application the record was created in.</td>
</tr>
<tr>
<td>Order</td>
<td>The order in which the screen appears. The lower the number the earlier it appears on a screen.</td>
</tr>
<tr>
<td>Item view</td>
<td>Preconfigured JSON that defines where and how fields appear on the top of the screen.</td>
</tr>
<tr>
<td>Icon</td>
<td>The image that appears to represent the screen. The image for the detail or list screens is fixed.</td>
</tr>
<tr>
<td>Hidden</td>
<td>Whether or not users can see the screen from the application home screen on a mobile device.</td>
</tr>
<tr>
<td>Data Item</td>
<td>Defines what information is being pulled into the screen. For more information on configuring a data item, see Configure a data item in the platform.</td>
</tr>
<tr>
<td>Available offline</td>
<td>Whether or not the screen is available in offline mode.</td>
</tr>
</tbody>
</table>

Details screen

The details screen provides additional information when you tap an item in a list. The activity stream screen is only available as an embedded screen. You can only configure it from a screen that has a Master Item related list, such as a master-detail screen or a calendar screen.
INC0000015

I can't launch my VPN client since the last software update

Opened 2018-03-25 16:38:46
Caller Fred Luddy

Priority image 4f718bc1db925300e897769e0f96196b.iix

Caller image 8f590e66ff021300f1f2883b73e11c31.iix

Category
Software

Contact phone number

Description
There was a software update two days ago and now my VPN doesn't work. Nothing loads when I click on the VPN client icon on my desktop.

© 2019 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent table</td>
<td>Automatically populated with the name of the master item or the map screen the screen was created from.</td>
</tr>
<tr>
<td>Parent</td>
<td>Automatically populated with the name of the master item or map screen the screen was created from. This field is how the screen gets associated with a master item.</td>
</tr>
<tr>
<td>Application</td>
<td>Automatically populated with the current application the record was created in.</td>
</tr>
<tr>
<td>Order</td>
<td>The order in which the screen appears. The lower the number the earlier it appears on a screen.</td>
</tr>
<tr>
<td>Item view</td>
<td>Preconfigured JSON that defines where and how fields appear on the top of the screen in a details header.</td>
</tr>
<tr>
<td>Hidden</td>
<td>Whether or not users can see the screen from the application home screen on a mobile device.</td>
</tr>
<tr>
<td>Available offline</td>
<td>Whether or not the screen is available in offline mode.</td>
</tr>
</tbody>
</table>

On the details screen below the details header, you can add more body fields to appear on the screen. For more information on configuring the fields that appear in the details body, see [Configure the fields that appear on a mobile screen](#).

**Activity Stream Screen**

Use the activity stream screen to determine if the activity stream is available for an applet on a mobile device. The activity stream screen is only available as an embedded screen. You can only configure it from a screen that has a Master Item related list, such as a master-detail screen or a calendar screen.
### Activity stream screen fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the activity stream screen. Name is a user facing field.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional additional details about the activity stream screen. This information does not appear in the mobile app.</td>
</tr>
<tr>
<td>Parent table</td>
<td>Automatically populated with the master item or map screen table.</td>
</tr>
<tr>
<td>Add additional comment button</td>
<td>Show or hide the Add additional comment button from the activity stream for a particular applet.</td>
</tr>
<tr>
<td>Add work note button</td>
<td>Show or hide the Add work note button from the activity stream for a particular applet.</td>
</tr>
<tr>
<td>Add attachment button</td>
<td>Show or hide the Add attachment button from the activity stream for a particular applet.</td>
</tr>
<tr>
<td>Parent</td>
<td>Automatically populated with the name of the master item or map screen the screen was created from. This field is how the screen gets associated with a folder or master item.</td>
</tr>
<tr>
<td>Application</td>
<td>Automatically populated with the current application the record was created in.</td>
</tr>
<tr>
<td>Order</td>
<td>The order in which the screen appears. The lower the number the earlier it appears on a screen.</td>
</tr>
<tr>
<td>Hidden</td>
<td>Whether or not users can see the screen on a mobile device.</td>
</tr>
<tr>
<td>Available offline</td>
<td>Whether or not the screen is available offline.</td>
</tr>
</tbody>
</table>

### Related list screen

Related lists display additional lists that are available for a record, for example SLAs, or configuration items. The related list screen can only be created from a master item, not a folder.
### Open Incidents

<table>
<thead>
<tr>
<th>Task SLAs (0)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Affected CIs (0)</td>
<td></td>
</tr>
</tbody>
</table>
### Related list screen fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the related lists screen. Name is a user facing screen.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional additional information about the map screen. This information does not appear in the mobile app.</td>
</tr>
<tr>
<td>Parent table</td>
<td>Automatically populated with the master item or map screen table.</td>
</tr>
<tr>
<td>Parent</td>
<td>Automatically populated with the name of the master item or map screen the screen was created from. This field is how the screen gets associated with a master item for a master-detail or calendar screen.</td>
</tr>
<tr>
<td>Order</td>
<td>The order in which the screen appears. The lower the number the earlier it appears on a screen.</td>
</tr>
<tr>
<td>Icon</td>
<td>The image that represents the related list screen.</td>
</tr>
<tr>
<td>Show count</td>
<td>Whether or not the number of records available for each related list item displays</td>
</tr>
<tr>
<td>Application</td>
<td>Automatically populated with the current application the record was created in.</td>
</tr>
<tr>
<td>Hidden</td>
<td>Whether or not users can see the screen on a mobile device.</td>
</tr>
<tr>
<td>Table</td>
<td>The table the related lists are on. For example, the incident table includes the Task SLAs and Affected CIs related lists.</td>
</tr>
<tr>
<td>Available offline</td>
<td>Whether or not the screen is available offline.</td>
</tr>
</tbody>
</table>

After you create a related lists screen, you need to add related lists. You can do this in the related list screen, from the Related lists mappings related list.

### Map screens

Use map screens to add a map to the mobile app. Create a master detail applet before you create a map screen.
## Map screen fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the map screen. This name appears in the mobile app.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional additional information about the map screen. This information appears when you tap the information icon for the applet on the applet screen.</td>
</tr>
<tr>
<td>Parent table</td>
<td>Automatically populated with the name of the folder the screen was created from.</td>
</tr>
<tr>
<td>Parent</td>
<td>Automatically populated with the name of the folder the screen was created from. This field is how the screen gets associated with a folder or master item.</td>
</tr>
<tr>
<td>Order</td>
<td>The order in which the screen appears. The lower the number the earlier it appears on a screen.</td>
</tr>
<tr>
<td>Icon</td>
<td>The image that represents the map screen from the folder view.</td>
</tr>
<tr>
<td>Hidden</td>
<td>Whether or not users can see the screen on a mobile device.</td>
</tr>
<tr>
<td>Application</td>
<td>Automatically populated with the current application the record was created in.</td>
</tr>
<tr>
<td>Data Item</td>
<td>Defines what information is being pulled into the screen. Make sure the data item for a map screen includes a reference to the cmn_location table. For more information on configuring a data item, see <a href="#">Configure a data item in the platform</a>.</td>
</tr>
<tr>
<td>Title</td>
<td>The field that appears as the title for records in the map.</td>
</tr>
<tr>
<td>Subtitle</td>
<td>The field that appears underneath the title for records in the map.</td>
</tr>
<tr>
<td>Info</td>
<td>A field with any additional information you want to appear in the record information on the map. For example, the location name.</td>
</tr>
<tr>
<td>Location</td>
<td>The form field that determines the location of pins on the map. The field should be a reference to the cmn_location table.</td>
</tr>
<tr>
<td>Tag</td>
<td>A field that appears in the top right for record information on the map. Select fields such as priority or status.</td>
</tr>
<tr>
<td>Tag style</td>
<td>Applies a UI style to map items. For more information on creating UI styles, see <a href="#">Mobile UI styles</a>.</td>
</tr>
<tr>
<td>Phone</td>
<td>Field for the contact phone number for the map item.</td>
</tr>
<tr>
<td>Pin color type</td>
<td>The color of the pin that appears on the map. Choose from Default or Custom. If you select Custom, another field to select the pin color appears.</td>
</tr>
</tbody>
</table>
Configure the fields that appear on a mobile screen

Determine the fields that appear on a screen on a mobile device using the Screen fields related list.

Role required: admin

1. Navigate to **System Mobile > Mobile Applications**.
2. Open the application that contains the screen you want to add fields to.
3. From folders related list for the application, open the folder that contains the screen you want to add fields to.
4. Open the screen that you want to configure fields for.
   You can only configure fields for details screens. The fields appear in the body section of the details screen.
5. In the Screen fields related list, click **New**.
6. Complete the following fields as needed.

**Screen fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data item field name</td>
<td>The field you want to appear in the mobile app. The list of available fields is determined by the data item associated with the screen.</td>
</tr>
<tr>
<td>Screen</td>
<td>Automatically populated with the name of the details screen.</td>
</tr>
<tr>
<td>Type</td>
<td>Data type for the field. Choose from:</td>
</tr>
<tr>
<td></td>
<td>- Auto</td>
</tr>
<tr>
<td></td>
<td>- Text</td>
</tr>
<tr>
<td></td>
<td>- Percentage</td>
</tr>
<tr>
<td></td>
<td>- Image</td>
</tr>
<tr>
<td>Hidden</td>
<td>Whether the field appears in the mobile app or not.</td>
</tr>
<tr>
<td>Order</td>
<td>The order in which the field appears.</td>
</tr>
</tbody>
</table>

7. From the Data item field name list, select a field from the list that you want to appear in the mobile app.
   Available fields are determined by the data item associated with the screen.
8. Click **Save**.

Configure screen parameters

You can pass in a parameter for a screen to call by associating a parametrized data item with a screen.
Role required: admin

1. Navigate to **System Mobile > Mobile Applications**.
2. Open an application.
3. From the folders related list for the application, open a folder with screens you want to parametrize.
4. From the folder, select a screen to configure parameters for.
5. In the data item field for the screen, select a parametrized data item.
   Any parameters in the data item appear in the Screen parameters mappings and the UI parameters related lists. Change the UI parameter input type as needed. The default input type is **Text** but other options are available, such as QR/barcode scan.
6. From the Screen parameters mappings related list, open an item parameter.
7. Configure the following fields as needed.

<table>
<thead>
<tr>
<th>Screen parameters mappings fields</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Field</strong></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Automatically completed with the name of the parameter associated with the data item</td>
</tr>
<tr>
<td>Parent table</td>
<td>Automatically completed with the data item table.</td>
</tr>
<tr>
<td>Parent</td>
<td>Automatically completed with the data item associated with the screen.</td>
</tr>
<tr>
<td>Type</td>
<td>Determines the type of keyboard that appears on the user's screen for input.</td>
</tr>
<tr>
<td></td>
<td>· String: Uses a full keyboard for input</td>
</tr>
<tr>
<td></td>
<td>· Integer: Uses a numbers-only keypad for input</td>
</tr>
<tr>
<td></td>
<td>· Decimal: Uses a numbers-only keypad for input</td>
</tr>
<tr>
<td></td>
<td>· Boolean: Uses a true or false selection option</td>
</tr>
<tr>
<td></td>
<td>· DateTime: Uses a calendar with an exact time selector</td>
</tr>
<tr>
<td></td>
<td>· Uses a calendar</td>
</tr>
<tr>
<td>Default value</td>
<td>The value that appears by default for the field</td>
</tr>
</tbody>
</table>

8. Click **Save**.
9. From the UI parameter related list for the screen, open a UI parameter.
10. Complete the following fields as needed.

<table>
<thead>
<tr>
<th>User Input Parameter Definition fields</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Field</strong></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Name for the UI parameter. You can have multiple UI parameters with the same name so make sure you choose something you can easily discover later. The name you select appears in the mobile app. For this example, use Priority as the name.</td>
</tr>
</tbody>
</table>
### Field | Definition
--- | ---
Input type | The UI used to complete the variable. For example, if you have a parameter for the Assigned to field, select **List** so that users have a list of users in that field to search for. Choose from one of the following options.

- **Text**: Provides a simple text field. This option works best for fields that require free text, such as work notes or resolution details. Text is the default type.
- **List**: Opens a list for the user to select from. This option works well for reference fields that require specific information.
- **SearchList**: Provides a search bar so that users can search in a list.
- **QR/Barcode**: Provides the option to search by QRC or barcode.

Table name | The table for the field you want to create a UI parameter for. For example, Incident.

Field name | The field name you want to create for the UI parameter. For example, Priority.

Input style | How the user interacts from the UI. Choose from inline or popup. For this example, choose Inline.

Default value type | The value that appears by default in the UI field. Select one of the following options.

- **None**: There is no default text. This works well for a list input type.
- **Manual**: An additional field appears for you to enter a default term. For example, *Search for a field*. The manual default works well for search or text input types.
- **Source field**: Pulls in information from the field selected in the Field name section.

For this example, leave the default value type as **None**.

Mandatory | Whether or not the user is required to enter information for that field. For this example, leave this checkbox cleared.

Placeholder | Text that appears below the field type. This option does not appear if you have a default value selected.

11. Click **Save**.

12. From the screen record, click **Update**.

### Creating functions for the mobile app in the platform

Functions determine actions a user can perform in the mobile app. Use the function type and button instance to configure button behavior. Function configuration should take place in Studio, however for more advanced configuration, such as scripted display conditions, use the platform view.

Studio has more precautions in place that prevent you from configuring anything that will break the mobile app. The majority of app configuration should take place in Studio. Use the platform for specific advanced configurations.

There are seven different button types you can use to configure actions in the mobile app.

- **Action item**: Actions that change data, such as, assigning a task to yourself, or adding a comment to a record.
• Navigation: Actions that move you to a new screen, for example, opening another record from a field on a record.
• URL
• Phone
• Address
• Email
• Upload attachment

Create a mobile action item in the platform

Action items determine how action functions apply on a mobile device. Action functions are actions in the app that change the data. For example, assigning a task or adding a comment.

Role required: admin

1. Navigate to System Mobile > Mobile Action Items.
2. Click New.
3. In the Name field, enter a name for the action item.
   You can have multiple action items with the same name. Make sure you choose a name that is easily identifiable.
4. In the Description field, add a description for the action item.
5. From the Type list, select a type of action item.
   • New: Add something new to the database.
   • Update: Update something in the database.
   • Delete: Delete something in the database.
   • Script: Create a scripted action item.
6. From the Table list, select the table that you want the action item to apply to.
7. Select or clear the Use record as current condition check box to determine if the action item uses the record as the condition. Selecting the check box hides the Query Condition fields. This option is not available if you create an action item with the type New.
8. From the context menu, click Save.
   The item parameters related list appears.
9. Optional: Create parameters for the action item.
   Parameters allow you to create a condition that is waiting for more information. For example, you can create an action item to assign a record, and create a parameter for assignee sys ID. When the user goes to assign the record, the system uses the parameter to find a list of assignees by sys ID.
   a) In the item parameters related list, click New.
      a) In the Name field, enter a name of the parameter. For example, Accept task sysID.
      You can have multiple parameters with the same name, so choose a name that you can distinguish easily. Typically, you should create at least one parameter to call the sys ID for the record.
      b) In the Type list, select the type of field for the parameter. The type determines the kind of keyboard that appears for the user. The majority of parameters use the parameter type of String. If you want only a numbers keyboard to appear, select Integer.
      The Parent table and Parent fields are automatically populated with the action item table and the record you are creating the parameter for.
      c) In the Default value field, enter a value for the field to use as a default. For example, for an integer parameter, include a number as a default.
d) Click **Submit**.

10. From the Use current record as condition checkbox, select or clear the checkbox to determine whether the action uses the record the user is currently on to determine what the condition should be. If selected, the Query conditions field is disabled.

11. In the Query Condition section, add a query condition to the action item. Create a query condition based on the options for creating a filter, or use the following steps to pass a parameter into the query condition.
   a) In the Query Condition field, click **Add Filter Condition**.
   b) From the field list, select a field to use as a query condition. For example, select **Sys ID**.
   c) Click the contextual reference value icon ( ) to select an item parameter.
   d) From the list of item parameters, select the one you want to pass in the to the query condition. In this example, select the sys ID item parameter.

12. In the Set field values section, add field values that you want to populate when the action is updated.
Set a field value based on the options for creating a filter, or use the following steps to pass in a parameter to set the field value for a record.
   a) In the Set field values field, from the field list, select a field. For example, select **Resolution notes**.
   b) Click the contextual reference value icon ( ) to select an item parameter.
   c) From the list of item parameters, select the parameter you want to pass into the field value. For example, to require resolution notes, pass in a parameter item for Resolve Resolution Note.

13. Click **Update** to save the information in the action item.

After you create an action item, you need to associate it with a specific button in the mobile app.

Create a function for the mobile app

Functions represent actions that users can perform in the mobile app. Different functions are available depending on what you want users to be able to do.

Role required: admin

1. In the application navigator, navigate to **System Mobile > Mobile Functions**.
2. Click **New**.
3. Complete the following fields as needed.

<table>
<thead>
<tr>
<th>Button fields</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Field</strong></td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Field</td>
</tr>
<tr>
<td>------------</td>
</tr>
<tr>
<td><strong>Label</strong></td>
</tr>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>Type</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Application</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Context</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Condition</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Table</strong></td>
</tr>
<tr>
<td>Field</td>
</tr>
<tr>
<td>-----------------------</td>
</tr>
<tr>
<td><strong>Condition</strong></td>
</tr>
<tr>
<td><strong>Roles</strong></td>
</tr>
<tr>
<td><strong>Messages</strong></td>
</tr>
<tr>
<td>Show confirmation message</td>
</tr>
<tr>
<td>Success message</td>
</tr>
<tr>
<td>Failure message</td>
</tr>
<tr>
<td><strong>Offline Properties</strong></td>
</tr>
<tr>
<td>Offline</td>
</tr>
</tbody>
</table>

4. From the context menu, save the function record.
The UI parameters and Redirection destination fields related lists appear.

5. Add a UI parameter.
   a) In the UI parameter related list, click **New**.
   b) Complete the following fields as needed.

### UI parameter fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Name for the parameter</td>
</tr>
<tr>
<td><strong>Parameter type</strong></td>
<td>Whether the UI parameter is a button or a screen. For this example, choose <strong>Button</strong>.</td>
</tr>
<tr>
<td><strong>Button</strong></td>
<td>Auto-filled from the button record. Name of the button you are creating the UI parameter for.</td>
</tr>
<tr>
<td><strong>Mandatory</strong></td>
<td>Whether or not the parameter is required.</td>
</tr>
<tr>
<td><strong>Order</strong></td>
<td>Order in which the parameter appears.</td>
</tr>
<tr>
<td><strong>Input source</strong></td>
<td>Whether the field is automatically filled or the user has to input information.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Input type</td>
<td>The type of field. The options in this list vary depending on if you select Auto fill or User input as the input source.</td>
</tr>
<tr>
<td>Input style</td>
<td>The type of input.</td>
</tr>
<tr>
<td></td>
<td>- Inline: User enters information inline</td>
</tr>
<tr>
<td></td>
<td>- Popup: User enters information in a popup screen.</td>
</tr>
<tr>
<td>Placeholder</td>
<td>Not used</td>
</tr>
<tr>
<td>Default value type</td>
<td>The type of value the parameter uses by default. Choose from Manual or Source field.</td>
</tr>
<tr>
<td>Default value</td>
<td>The value that appears in the field by default.</td>
</tr>
<tr>
<td>Button parent table</td>
<td>Auto-filled from the table you created the button for. This field only appears if Source Field is selected as the Input type.</td>
</tr>
<tr>
<td>Source field</td>
<td>The field on the parent table that the parameter pulls information from. This field only appears if Source Field is selected as the Input type.</td>
</tr>
</tbody>
</table>

6. Click **Update** to save the Button record.

Determine the behavior for the button by configuring the button instance and where it appears in the mobile app.

Configure a function instance for the mobile app

Function instances determine where a function is on a page and how it behaves depending on that location.

Function instances reference records in the master item table. Configure a master item to assign the function instance to before creating a function instance.

Role required: admin

1. From the master item or screen you want to create a button instance for, in the Function Instance related list, click **New**.
2. Complete the following fields as needed.

**Function instance fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for the button instance.</td>
</tr>
<tr>
<td>Description</td>
<td>Additional information about the button instance</td>
</tr>
<tr>
<td>Parent table</td>
<td>Table that the button instance applies to. If you want to create a top button action, this field should be set to the Screen (sys_screen) table. If you want to create a swipe action, this field should be set to the Master Item screen.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Parent</td>
<td>The master item that the button instance is for. If you create a button instance from a master item, this field is already populated.</td>
</tr>
<tr>
<td>Function</td>
<td>The name of the button you want to create a button instance for.</td>
</tr>
<tr>
<td>Label</td>
<td>The name that appears for the button in the mobile app.</td>
</tr>
<tr>
<td>Location</td>
<td>Where the button instance appears on the page. The location of the button defines the behavior. Select from:</td>
</tr>
<tr>
<td></td>
<td>- Top: Function appears in the menu at the top of the screen.</td>
</tr>
<tr>
<td></td>
<td>- Trailing swipe: Function appears when a user swipes an item to the left.</td>
</tr>
<tr>
<td></td>
<td>- Leading swipe: Function appears when a user swipes an item to the right.</td>
</tr>
<tr>
<td></td>
<td>- Field: The field is the function.</td>
</tr>
<tr>
<td></td>
<td>- List item: Navigation is from the list screen.</td>
</tr>
<tr>
<td></td>
<td>For the swipe or top button actions, make sure you select the correct parent table.</td>
</tr>
<tr>
<td>Order</td>
<td>The order the function appears in. You can have multiple functions in the same location.</td>
</tr>
<tr>
<td>Active</td>
<td>Whether or not the function instance is active.</td>
</tr>
</tbody>
</table>

3. Click **Submit**.

Roles and permissions for the mobile app

Limit user access to parts of the mobile app.

Limit user access to the following components of the mobile app.

- **Application**: Limit a user’s ability to access certain applications in the mobile app. For example, prevent ITIL users from accessing field service applications. For more information on limiting user access by role to an application in the mobile app, see [Create an application for a mobile device in Studio](#).

- **Functions**: Only allow users with certain roles to perform certain actions in the app. For example, limit an ITIL user’s ability to reassign an incident from a swipe action. For more information on limiting user access by role to a specific function, see the steps for creating each function type listed in this section.

If you do not add a role to any of these components, any user who has access to the mobile app can see that component.

Localization on mobile devices

Native mobile apps are localized in four languages: Spanish, French, German, and Japanese.

Right to left languages are not supported in the native mobile app.
The native mobile applications are localized in two different ways, which means that there is a blending of localization visible inside of the native applications.

- **Native (on device) localization**: Controlled by the device’s language preference, which means that many components are localized with the language preference for the user’s device. These components can include local screen titles (such as Settings) and local button titles (such as the Clear All button on the filters screen).

  The on-device localization supports Spanish, German, French, and Japanese. These translations can’t be customized since they are shipped with the application binary and translated natively on the device according to the language settings for the device.

- **Server-side localization**: Controlled the same was as desktop web localization (server system language / user preference on server). Components that are localized on the server include things like field labels, web content, and other data that is stored on the server in a translated field.

  You can customize translations that use server-side localization the same way you do on the desktop. For more information on configuring server-side localization, see [System localization](#).

**Domain separation in the Mobile application**

This is an overview of domain separation in the Mobile application. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can then control several aspects of this separation, including which users can see and access data.

**Overview**

**Support: Data only**

Domain separation in this application is supported at the **Data only** level, meaning it supports the data security model of separating visibility of data from one domain to another. To learn more, see [Application support for domain separation](#).

**Migrate from classic mobile to ServiceNow mobile**

Migrate from ServiceNow Classic to ServiceNow mobile to take advantage of features such as rapid development, offline capability, and integration with native mobile device features.

**Understand the benefits of migration to ServiceNow mobile**

ServiceNow mobile is a native, mobile-first design that introduces new capabilities that were not available in ServiceNow Classic:

- Rapid, low code application development using Studio.
- Base system applications for Approval Management, ITSM, and Field Service Management.
- Integration with native mobile device features, such as camera, phone, SMS, and map apps.
- Offline capabilities, including offline read and write.
- Enhanced security features including data loss prevention.

For more detail on ServiceNow mobile, see [Mobile navigation and configuration](#).
Evaluate your organization’s current mobile use

Start off with a list of business goals and determine which features your users need most. Consider which screens, fields, and actions your users use frequently, as well as what functionality your users do not use.

There is no direct method of transferring ServiceNow Classic components into the ServiceNow mobile product. You can create the same functionality available to your users with base system applications and manually building mobile applications in the ServiceNow mobile framework.

Use base system applications when possible

ServiceNow provides several base system applications to get you started quickly with commonly used functionality. Base system applications are tested by ServiceNow, and receive updates automatically as you upgrade your instance. The Madrid release includes applications for Approval Management, Field Service Management, and Incident Management. Use these applications right away, or customize them to suit your business needs using Studio.

Mobile experience for Approvals

Approve catalog requests, requested items, or change requests from anywhere using the ServiceNow Agent application. For more detail, see Mobile experience for Approvals.

Mobile experience for Field Service Management

Manage your field service tasks anywhere using the Field Service mobile application. With this application, you stay connected with your mobile device, can access important information, and complete your tasks quickly. For more detail, see Mobile experience for Field Service Management.

Mobile experience for Incident Management

Manage incident tasks anywhere using the Incidents mobile application, which enables you to stay connected and access important information to complete tasks quickly. For more detail, see Mobile experience for Incident Management.

Create your own applications

Use Studio to make new applications, and allow your users to work in areas not covered by the current base system applications. For details on the components of ServiceNow mobile and examples of application creation, see ServiceNow mobile app configuration.

Understand the current limitations for ServiceNow mobile

General

- Client scripts are not supported.
- The home screen image cannot be customized.
- The Agent app theme cannot be customized.

Screen UI Policies

- Screen UI Policies can’t control whether field is read-only.
- Screen UI Policy conditions cannot be scripted.
Supported devices for the ServiceNow Agent mobile app

The ServiceNow Agent app has been officially tested on the following devices, although additional devices might work as well. The app requires Madrid or later.

Mobile supported devices

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Supported devices</th>
<th>Supported versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple® iPhone®</td>
<td>• iPhone 7 and 7s&lt;br&gt;• iPhone 8 and 8s&lt;br&gt;• iPhone X</td>
<td>The application is supported on iOS11 and higher. For optimal performance use the latest OS and devices.</td>
</tr>
<tr>
<td>Apple® iPad®</td>
<td>• iPad&lt;br&gt;• iPad Pro</td>
<td>The application is supported on iOS11 and higher. For optimal performance use the latest OS and devices.</td>
</tr>
<tr>
<td>Android™</td>
<td>• Samsung S8&lt;br&gt;• Samsung S9&lt;br&gt;• Xiaomi Mi8&lt;br&gt;• Huawei P20&lt;br&gt;• Google Pixel 2&lt;br&gt;• Google Pixel 2 XL&lt;br&gt;Android tablet is not officially supported.</td>
<td>The application is supported on Lollipop 5.0 and higher. For optimal performance use the latest OS and devices.</td>
</tr>
</tbody>
</table>

Support for ServiceNow Classic

Specific devices and operating systems are supported by the ServiceNow Classic mobile app and by the mobile browser interface.

Devices supported by the ServiceNow Classic mobile app interface

<table>
<thead>
<tr>
<th>Device</th>
<th>Supported versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple® iPhone®</td>
<td>Version 6.0 and higher of the app for iOS requires iOS 11 and above for all models of iPhone. Previous versions of the app support all models of iPhone running iOS 9 and above.</td>
</tr>
<tr>
<td>Apple® iPod® touch</td>
<td>Version 6.0 and higher of the app requires iOS 11 and above for all models of iPod. Previous versions of the app support all models of iPod running iOS 9 and above.</td>
</tr>
<tr>
<td>Apple® iPad®</td>
<td>Version 6.0 and higher of the app requires iOS 11 and above for all models of iPad. Previous versions of the app support all models of iPad running iOS 9 and above.</td>
</tr>
<tr>
<td>Apple® watch</td>
<td>All models of watch running iOS 2 and above</td>
</tr>
<tr>
<td>Device</td>
<td>Supported versions</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Android™ mobile</td>
<td>All models of Android phone running KitKat (4.4) and above.</td>
</tr>
<tr>
<td>devices</td>
<td></td>
</tr>
</tbody>
</table>

**Limitations**

The platform does not have a native Blackberry or Windows phone app. Use the mobile web experience on these devices.

The mobile apps do not give access to dashboards as dashboards are not optimized for mobile screen sizes. You can access dashboards on a tablet using the standard web interface.

**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>Apple® iPad®</td>
<td>All models of iPad® running iOS 9 and above are supported. 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>Android™ tablet</td>
<td>Any Android tablet running 4.4 (KitKat) 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. 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.</td>
</tr>
<tr>
<td>BlackBerry®</td>
<td>All BlackBerry devices running BlackBerry 10 and above. Some configuration is required.</td>
</tr>
</tbody>
</table>

In UI16, browsers on a tablet use the same interface as the desktop browser. Mobile browsers do not support the UI16 interface. Instead they use the same interface as the ServiceNow Classic mobile app.

**Note:** Do not use the mobile interface on a desktop browser except for testing purposes.

**Mobile browser limitations**

The iOS version of Firefox does not support ServiceNow Community or other Service Portal-based pages.

© 2019 ServiceNow, Inc. All rights reserved. ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.
Enterprise mobility management (EMM)

Use an EMM suite to distribute the ServiceNow mobile app or the ServiceNow Classic mobile app. You can only distribute the iTunes or Google Play store versions of the mobile app. The .ipa and .apk files are not publicly available so the apps cannot be re-signed for internal distribution. You can find more information on using an EMM suite in the documentation for the EMM suite of your choice. Internal distribution is not supported.

App wrapping or SDK embedding is currently not supported, therefore not all app policies are supported.

App Config

When you distribute the app through an EMM suite, such as AirWatch, you can use App Config to pre-configure the default instance URL. Use the following key/value pairing:

- Key: SNDefaultInstanceURL
- Value: URL for your instance

AirWatch Mobile Device Management (MDM)

Use VMware AirWatch to distribute and manage the ServiceNow mobile app on user devices. Use the following steps to configure the ServiceNow mobile app for iOS and Android.

You should have access to AirWatch before attempting any configuration.

Add the ServiceNow app for iOS to AirWatch

Configure the ServiceNow app for iOS for AirWatch distribution.

Role required: admin

1. Sign in to the AirWatch portal.
3. Click Add Application.
4. In the Platform field, select Apple iOS.
5. In the Source field, select Search App Store then search for ServiceNow.
6. Next to the ServiceNow App search result, click Select.
7. Click Save & Assign.
8. Click Add Assignment.
9. Click Assignment Group.
10. From the Assignment Group list, select Application Configuration Policy.
11. From the Application Configuration field, select Enabled.
12. In the Application Configuration section, enter the following information.

<table>
<thead>
<tr>
<th>Application Configuration fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
</tr>
<tr>
<td>Configuration key</td>
</tr>
<tr>
<td>Value type</td>
</tr>
<tr>
<td>Field</td>
</tr>
<tr>
<td>-----------------------</td>
</tr>
</tbody>
</table>
| Configuration value   | https://<instance name>.service-
                          | now.com/                      |
Would you like to enable Data Loss Prevention (DLP)?

DLP policies provide controlled exchange of data between managed and unmanaged applications on the device.

To prevent data loss on this application, make it "Managed Access" and create "Restriction" profile policies for desired device types.

<table>
<thead>
<tr>
<th>Managed Access</th>
<th>Enabled</th>
<th>Disabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove On Unenroll</td>
<td>Enabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>Prevent Application Backup</td>
<td>Enabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>Make App MDM Managed if User Installed</td>
<td>Enabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>App Tunneling</td>
<td>Enabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>Application Configuration</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Enter Key-Value pairs to configure applications for users:

<table>
<thead>
<tr>
<th>Configuration Key</th>
<th>Value Type</th>
<th>Configuration Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNDefaultInstanceURL</td>
<td>String</td>
<td>https://&lt;instance name&gt;.servil</td>
</tr>
</tbody>
</table>

Add
13. Click Add.
14. Click Save & Publish.
15. Click Publish.

After adding the app to AirWatch, it appears in the Apps & Books > Applications > Native > Public section of the AirWatch portal. Mobile users can download the app from the AirWatch App Catalog app.

Add the ServiceNow Android app to AirWatch

Configure the ServiceNow Android app for AirWatch distribution.

Role required: admin

1. Sign in to the AirWatch portal.
3. Click Add Application.
4. In the Platform field, select Android.
5. In the Source field, select Search App Store then search for ServiceNow.
   If the search cannot find the ServiceNow application, the app might not be approved for your organization. You can approve the app in the Google Play store.
6. Next to the ServiceNow App search result, click Select.
7. Click Save & Assign.
8. Click Add Assignment.
9. Click Assignment Group.
10. From the Assignment Group list, select Application Configuration Policy.
11. From the Application Configuration field, select Enabled.
12. In the Application Configuration section, enter the following information.

<table>
<thead>
<tr>
<th>Application Configuration fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
</tr>
<tr>
<td>Title</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Value</td>
</tr>
</tbody>
</table>
ServiceNow - Add Assignment

Select Assignment Groups
- All Devices (ServiceNow, Inc.)
- Start typing to add a group

App Delivery Method *
- Auto
- On Demand

Policies

Adaptive Management Level: Managed Access
Apply policies that give users access to apps based on administrative management of devices.

Would you like to enable Data Loss Prevention (DLP)?
DLP policies provide controlled exchange of data between managed and unmanaged applications on the device.
To prevent data loss on this application, make it "Managed Access" and create "Restriction" profile policies for desired device types

Managed Access
- Enabled
- Disabled

App Tunneling
- Enabled
- Disabled

Application Configuration
- Enabled
- Disabled

Enter Key-Value pairs to configure applications for users:

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server URL</td>
<td>The full ServiceNOW Instance URL</td>
<td>https://&lt;instance names&gt;.service-now</td>
</tr>
</tbody>
</table>
13. Click Add.
14. Click Save & Publish.
15. Click Publish.

After adding the app to AirWatch, it appears in the Apps & Books > Applications > Native > Public section of the AirWatch portal. Mobile users can download the app from the AirWatch App Catalog app.

If your instance is on an internal network, you might need to configure the VPN. For more information on configuring the VPN, see the AirWatch documentation on configuring per-app tunnel profiles.

For more information on app distribution with AirWatch, see the official AirWatch documentation.

BlackBerry Unified Endpoint Management (UEM)

Use BlackBerry UEM as a mobile device management system to distribute and manage the ServiceNow mobile app on user devices.

Before adding the ServiceNow app to BlackBerry UEM, complete the BlackBerry UEM setup procedures.

Add the ServiceNow app for iOS to BlackBerry UEM

Configure the ServiceNow app for iOS for BlackBerry UEM management and distribution.

Role required: admin
1. Sign in to BlackBerry UEM.
2. In the menu bar, click Apps.
3. Click the Add an app icon (➕).
4. Click App Store.
5. From the country list, select United States.
6. In the search field, type ServiceNow.
7. In the search results list, click the add button next to the ServiceNow app.
8. In the app options page, in the app configuration section, click the add icon to add a default instance.
9. Click Configure manually.
10. In the name field for the app configuration, type a name for the configuration. For example, `Default Instance`.

11. Click + and add the following key/value pairing.
App configuration fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration key</td>
<td>SNDefaultInstanceURL</td>
</tr>
<tr>
<td>Value type</td>
<td>String</td>
</tr>
<tr>
<td>Configuration value</td>
<td>The URL for the instance. For example, https://&lt;instance name&gt;.service-now.com/</td>
</tr>
</tbody>
</table>

12. Click **Save**.
13. Click **Add**.
14. Assign the app to users or groups. When you assign the app, make sure you select the name of that you assigned in the App Configuration section.

![Assign apps](image)

**BlackBerry app assignments**

After successfully adding the ServiceNow app from the App Store, it appears in the Work Apps section on the BlackBerry UEM Portal. Mobile users can download it from the BlackBerry UEM portal.

**Add the ServiceNow app for Android to BlackBerry UEM**

Configure the ServiceNow app for Android for BlackBerry UEM management and distribution.

Role required: admin

1. Sign in to BlackBerry UEM.
2. In the menu bar, click **Apps**.
3. Click the Add an app icon (➕).
4. Click **Google Play**.
5. Paste the following URL into the search bar to search for the ServiceNow app for Android.

6. In the search results list, click the add button next to the ServiceNow app.

7. In the app configuration section, click the add icon to add a default instance.

8. In the name field for the app configuration, type a name for the configuration. For example, Default Instance.

9. In the server URL field, add the full URL for your instance. For example, https://<instance name>.service-now.com/.

10. Click Save.

11. Click Add.

12. Assign the app to users or groups. When you assign the app, make sure you select the name of that you assigned in the App Configuration section.

---

**BlackBerry app assignments**

*Note:* For App Configuration to be applied to the ServiceNow app for Android, use the Work and personal - user privacy (Android for Work) or the Work and personal - user privacy (Android for Work - Premium) activation policy. Currently, if you use the MDM control activation policy, you cannot apply the App configuration. For more information on Android activation types, see the BlackBerry UEM documentation on Android activation types.

---

After successfully adding the ServiceNow app for Android from the Google Play store, it appears in the Google Play store managed by your Android Work Profile.

For more information on app distribution using the public app store with BlackBerry UEM, see the BlackBerry UEM documentation.
If your instance is on an internal network, you might need to configure the VPN. For more information on configuring the VPN, see the BlackBerry UEM documentation on VPN configuration.

Citrix XenMobile

Configure Citrix XenMobile to distribute and manage the ServiceNow mobile app on user devices.

Use the following steps to distribute the ServiceNow mobile app using the Citrix XenMobile service. You can also configure the mobile app for your default ServiceNow instance for the app.

You should have access to Citrix before attempting any configuration.

Add the ServiceNow app to Citrix XenMobile

Configure the ServiceNow for Citrix XenMobile distribution.

Role required: admin

1. Sign in to the XenMobile console.
2. Click **Configure > Apps**.
3. Click **Add**, then choose **Public App Store**.
4. In the App Information pane, complete the following information.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type a descriptive name for the app. The name appears under App Name on the apps table.</td>
</tr>
<tr>
<td>Description</td>
<td>Type an optional description of the app.</td>
</tr>
<tr>
<td>App category</td>
<td>In the list, click the category you want to add the app to.</td>
</tr>
</tbody>
</table>

5. Click **Next**.
6. Under Platform, select the following platforms, or the platforms that are applicable to your distribution plan.
   - iPhone
   - iPad
   - Android for Work

7. For each of the platforms, complete the following steps.
   a) In the search bar, search for ServiceNow.
   b) Click the ServiceNow tile.
   c) Configure the deployment rule depending on your organizations deployment plan.
   d) Click **Next**.

   If Android for Work cannot find the ServiceNow application, the app might not be approved for your organization. You can [approve the app in the Google Play store](https://play.google.com/work/apps/details?id=com.servicenow.servicenow).

   If the search results still do not display in Citrix, click **Didn’t find the app you were looking for?** then enter the URL for the Android app: **https://play.google.com/work/apps/details?id=com.servicenow.servicenow**.

8. Click **Next**.
9. Optional: From the Approvals page, configure the workflow for your company.
You only need to use workflows when you need approval when creating user accounts. If you do not need to set up approval workflows, skip to the next step. For more information on configure workflows for Citrix, see the Citrix product documentation.

10. From the Delivery Group Assignments page, select the delivery groups for users that you want to have the ServiceNow application. You can also configure a deployment schedule.

After successfully adding the ServiceNow app from the iOS store platforms, install the ServiceNow application from the Citrix Hub app from your devices. For Android, the ServiceNow app appears in the Google Play Store managed by your Android Work Profile.

### Configure the ServiceNow app for the default instance

Configure apps that support managed configuration by deploying an XML configuration file to users' iOS devices.

Configuring the app for a default instance is only available for iOS devices.

1. In the XenMobile console, click **Configure > Device Policies**.
2. Click **Add**.
3. In the Add a New Policy page, under Apps, click **App Configuration**.
4. On the Policy Information page, enter the following information.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy Name</td>
<td>Type a descriptive name for the policy.</td>
</tr>
<tr>
<td>Description</td>
<td>Optionally, type a description of the policy</td>
</tr>
</tbody>
</table>

5. Click **Next**.
6. In the Policy Platforms panel, select **iOS**.
7. In the Identifier list, select **Add new**.
8. Enter the following in the add new identifier field: `com.servicenow.servicenow`.
9. In the Dictionary content field, enter the following text.

```
<dict>
  <key>SNDefaultInstanceURL</key>
  <string>https://<instance_name>.service-now.com</string>
</dict>
```

Use the name of your own instance where it says `<instance_name>`.

10. Click **Check Dictionary** to verify the XML.
12. Click **Next**.
13. From the Delivery Group Assignments page, select the delivery groups for users that you want to have the ServiceNow application. You can also configure a deployment schedule.
14. Click **Save**.

### IBM Maas360 mobile device management (MDM)

Use IBM Maas360 to monitor and manage smartphones, tablets, and other mobile devices from a web-based portal. Add the ServiceNow mobile app to the IBM Maas360 portal to monitor and manage distribution.

Before adding the ServiceNow app to the IBM Maas360 portal, complete the IBM Maas360 setup procedures.
Add the ServiceNow app for iOS to the IBM MaaS360 portal

Configure the ServiceNow app for iOS for IBM MaaS360 distribution.

Role required: admin

1. Sign in to the IBM MaaS360 portal.
2. From the portal homepage, navigate to Apps > Catalog.
3. Click Add to display the list of apps.
5. In the App Details tab, type ServiceNow into the app field.
6. Select the ServiceNow app from the results.

7. In the Policies and Distribution tab, select the applicable policies.
8. In the Configuration tab, in the App Config Source list, select Key/Value.
9. In the Key/Value section add the following information.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute name</td>
<td>SNDefaultInstanceURL</td>
</tr>
<tr>
<td>Attribute value</td>
<td>The full URL for the instance. For example, https://&lt;instance name&gt;.service- now.com/</td>
</tr>
</tbody>
</table>

10. Click Add.

After adding the ServiceNow app from the Apple store, it appears in the App Catalog application. Users can download the app to their iOS devices.
Add the ServiceNow Android app to the IBM MaaS360 portal

Configure the ServiceNow Android app for IBM MaaS360 distribution.

Role required: admin

1. Sign in to the IBM MaaS360 portal.
2. From the portal homepage, navigate to Apps > Catalog.
3. Click Add to display the list of apps.
5. In the App Details tab, type ServiceNow into the app field.
6. Select the ServiceNow app from the results.
7. In the Policies and Distribution tab, select the applicable policies.
8. In the Configuration tab, select the Configure App Settings checkbox.
9. In the Server URL field, enter the full URL for the instance. For example, https://<instance name>.service-now.com/.
10. Click Add.

After adding the ServiceNow app from the Google Play store, it appears in the App Catalog application. Users can download the app to their Android devices.

If your instance is on an internal network, you might need to configure the VPN. For more information on configuring the VPN, see the IBM MaaS360 documentation on the Maas360 VPN module.

For more information on app distribution, see the following IBM documentation for each operating system.

- Add an iTunes App Store app to the App Catalog
- Add a Google Play app to the App Catalog

Intune mobile device management (MDM)

Microsoft Intune uses Azure to manage mobile devices and apps. With some additional configuration, you can manage the ServiceNow mobile app in Intune.

You need to have access to an Azure account in order to add the ServiceNow mobile app to the store. Complete the Intune configuration steps before adding any apps to the Intune portal.

Add the ServiceNow app for iOS to Microsoft Intune

Configure the ServiceNow app for iOS for Microsoft Intune distribution.

Role required: admin

1. Sign in to the Microsoft Azure portal.
2. Select More Services > Monitoring + Management > Intune.
3. On the Intune blade, choose Manage apps.
4. In the Mobile apps workload, choose Manage > Apps.
5. Above the list of steps, click Add.
6. In the type field, select the iOS store app.
7. In the Add App blade, click Search the App Store.
8. In the Apple App Store blade, for the country locale, select United States.
9. In the search field, type servicenow. Then select the ServiceNow app from the results.
10. In the App information, make sure **Display this as a featured app in the Company Portal** is set to **Yes**.

11. In the Add App blade, click **Add**.

12. From the list, choose the app you want, then click **OK**.

**Add the default instance for iOS using App Config**

Use the application configuration values to add a default instance of the app for iOS.

**Role required:** admin

1. In the Mobile apps workload, select **Manage > App configuration policies**.

2. Click the **Add** button above the app config policies list.

3. In the Name field, enter a relevant name for the default instance policy for the ServiceNow app.

4. From the Device enrollment type list, select **Managed devices**.

5. From the Platform list, select **iOS**.

6. In the Associated app field, select the ServiceNow App.

7. In the Configuration settings format list, select **Use configuration designer**.

8. Set the following values.

**App Config values**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Key</td>
<td>SNDefaultInstanceURL</td>
</tr>
<tr>
<td>Value type</td>
<td>String</td>
</tr>
<tr>
<td>Configuration value</td>
<td>The full URL for your instance. For example, https://&lt;instance name&gt;.service-now.com/</td>
</tr>
</tbody>
</table>
9. Save the App configuration policy, then make sure Assigned is set to Yes.

Add the ServiceNow app for Android to Microsoft Intune

Configure the ServiceNow app for Android for Microsoft Intune distribution.

Role required: admin

To set up the app, set up the Android for Work in Intune, then follow the instructions on how to assign apps to Android for Work devices with Intune.

1. In the Mobile apps workload, select Manage > App configuration policies.
2. Click the Add button above the app config policies list.
3. In the Name field, enter a relevant name for the default instance policy for the ServiceNow app.
4. From the Device enrollment type list, select Managed devices.
5. From the Platform list, select Android for Work.
6. In the Associated app field, select the ServiceNow App.
7. In the Configuration settings, select the Use configuration designer, then click Add.
8. In the new editor window, make sure the server URL is the full instance URL for the app, then click **Ok**.

9. In the Configuration Value field, enter the name of your instance.

10. Save the App configuration policy, then make sure Assigned is set to **Yes**.
Jamf Pro mobile device management (MDM)

Use Jamf Pro to distribute and manage the ServiceNow mobile app on user devices. Use the following steps to configure the ServiceNow mobile app for iOS and Android. You should have access to Jamf Pro before attempting any configuration.

Configure the ServiceNow mobile app for Jamf Pro

Use app configuration to pre-configure the instance URL in Jamf Pro.

Role required: admin

1. Log in to Jamf Pro.
2. Click Devices.
3. In the Devices panel, click Mobile Device Apps.
4. On the mobile device apps page, click New.
5. Under Choose an App type, select App Store app or VPP store app.
6. Click Next.
7. In the search field, type ServiceNow.
8. Next to the ServiceNow app, click Add.
10. In the Preferences field, paste the following string:

   
   ```
   <dict>  
   <key>SNDefaultInstanceURL</key>  
   <string>https://<instance_name>.service-now.com</string>  
   </dict>
   
   
   Use the name of your own instance where it says <instance_name>.

11. Click Save.
After the instance is configured for the mobile app, configure the app settings to distribute the app to mobile devices in the scope.

MobileIron Mobile Device Management (MDM)

MobileIron is a cloud-based service you can use to manage applications for your users. Users can access apps from MobileIron while you manage and secure any content on the network.

Before adding the ServiceNow app to MobileIron, complete the MobileIron setup procedures.

Add the ServiceNow app for iOS to MobileIron

Configure the ServiceNow app for iOS for MobileIron distribution.

Role required: admin

1. Sign in to the MobileIron portal.
2. Navigate to Apps > App Catalog, then click Add.
3. From the search menu list, select the Apple App Store.
4. In the search bar, type ServiceNow.
5. Select the ServiceNow app from the results.
6. Click Next.
7. Optional: Describe and add screenshots of the app.
8. Select one of the following distribution levels for this configuration of the app:
   - Everyone: The app is added to all user-compatible devices
   - No one: The app is staged for distribution at a later date
   - Custom: The app is distributed only to users or groups you choose

9. Click Next.
10. In the App Configuration section, click the add icon (➕) next to iOS Managed App Configuration.
11. In the Name field, enter a name for the configuration. For example, Default Instance.
12. Enter the following key and value pairing.

   **iOS Managed App settings**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>SNDefaultInstanceURL</td>
</tr>
<tr>
<td>Value</td>
<td>The URL for the instance. For example, https://&lt;instance name&gt;.service-now.com/</td>
</tr>
</tbody>
</table>

13. Select one of the following distribution levels for this configuration of the app:
   - Everyone: The app is added to all user-compatible devices
   - No one: The app is staged for distribution at a later date
   - Custom: The app is distributed only to users or groups you choose

14. Click Next.
15. Click Done.

After successfully adding the ServiceNow app for iOS from the App Store, it appears in the App Catalogs section on the MobileIron Portal. Mobile users can download it from the MobileIron App Catalog.

**Add the ServiceNow app for Android to MobileIron**

Configure the ServiceNow app for Android for MobileIron distribution.

Role required: admin

1. Sign in to the MobileIron portal.
2. Navigate to Apps > App Catalog, then click Add.
3. From the search menu list, select the Google Play store.
4. In the search bar, type ServiceNow.
5. Select the ServiceNow app from the results.
6. Click Next.
7. Optional: Describe and add screenshots of the app.
8. Click Next.
9. Select one of the following distribution levels for this configuration of the app:
   - Everyone: The app is added to all user-compatible devices
   - No one: The app is staged for distribution at a later date
- Custom: The app is distributed only to users or groups you choose

10. In the App Configuration Summary section, click **Managed Configurations for Android**.

11. To add the default instance configuration, complete the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for the configuration. For example, Default instance.</td>
</tr>
<tr>
<td>Configuration value</td>
<td>The full instance URL. For example, https://&lt;instance name&gt;.service-now.com/</td>
</tr>
</tbody>
</table>

12. Select one of the following distribution levels for this configuration of the app.
   - Everyone: The app is added to all user-compatible devices
   - No one: The app is staged for distribution at a later date
   - Custom: The app is distributed only to users or groups you choose
ServiceNow Agent

ServiceNow | Version 7.1 | Delegation Status: App is delegated

App Configurations Summary

Managed Configurations for Android

Set up

Name
Default Instance

Add Description

Managed Configurations

Apply these configurations to the app when silently distributed to the users.

- Blocks app from sharing widget across profiles.
- Blocks the user from uninstalling the app.

Managed Configurations

<table>
<thead>
<tr>
<th>Configuration Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server URL</td>
<td>[http://&lt;instance name&gt;](http://&lt;instance name&gt;)</td>
</tr>
<tr>
<td>Server Name</td>
<td>[ServiceNow Instance Name](http://&lt;instance name&gt;)</td>
</tr>
</tbody>
</table>

Runtime Permissions

For Apps created with API 23+ and Android 6.0+

Manage Permissions

Runtime Permissions

Distribute this App Config

Choose one of these options

- **Everyone with App**: All Users who have the app
- **No One**: Stage this App Config for later distribution
- **Custom**: This config goes to a custom defined set of users and/or user groups

© 2019 ServiceNow, Inc. All rights reserved.

ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.
13. Click **Next**.
14. Click **Done**.

After successfully adding the ServiceNow app for Android from the Google Play store, it appears in the App Catalogs section on the MobileIron Portal. Mobile users can download it from the MobileIron App Catalog.

For more information on the MobileIron App Catalog, see the [MobileIron documentation](#).

If your instance is on an internal network, you might need to configure the VPN. For more information on configuring the VPN, see the MobileIron documentation on [VPN configuration](#).

**Approve the ServiceNow app for Google Play**

Android for Work may not be able to find the ServiceNow application. If that is the case, the ServiceNow App might not be approved for your organization. You can approve the app by finding it in the Google Play store.

Role required: none, however, you must either go to Google Play through the MDM, or use the same email address with the MDM that you use to sign in to the Google Play store.

1. Navigate to the Google Play store.
2. Search for the ServiceNow mobile app.
3. Click **Approve**.
ServiceNow Classic mobile app

Use the ServiceNow Classic mobile app to access records, update information, and collaborate with other users.

ServiceNow Classic device security

This document applies to the current ServiceNow app for iOS and Android for Madrid. This document may be subject to change for future mobile releases and re-platforming efforts.

Components and architecture for ServiceNow Classic

The ServiceNow solution consists of the ServiceNow server instance and the iOS and Android hybrid apps. A hybrid app includes both native and web components. The mobile client applications communicate over a wireless connection with the server and pull live data for the end user.

Component explanations

**App for iOS**

The ServiceNow app for iOS is a hybrid application that can be used on iPhone, iPad, and Apple Watch. Most components are native, however, there are web components, such as forms. It can be downloaded from the app store directly by a user, or can be pulled, dynamically configured, and distributed using MDM (more information available in EMM section). ServiceNow does not currently distribute the IPA file to customers.

**App for Android**

The ServiceNow App for Android is a hybrid application that can be used on Android devices. Most components are native, however, there are web components, such as forms. It can be downloaded from the app store directly by a user, or can be pulled, dynamically configured, and distributed using MDM (more information available in EMM section). ServiceNow does not currently distribute the APK file to customers.

Identity and access management

Control user access with user authentication, session timeout, and termination.

User authentication for ServiceNow Classic

The mobile app supports platform authentication using OAuth 2.0. Authentication mechanisms include multi provider SSO, MFA, LDAP, Local DB, and Digest.

**Multi Provider SSO**

The mobile app uses federated login when using the multi provider SSO plugin (com.snc.integration.sso.multi.installer). For more information on configuring multi provider SSO, see [Single sign on for the ServiceNow Classic mobile app](#).
Multifactor authentication

Users can access the instance via Multifactor Authentication using the MFA plugin (com.snc.integration.multifactor.authentication). For more information on MFA configuration, see mobile multifactor authentication.

LDAP

Use LDAP authentication to access using LDAP credentials. For more information on LDAP configuration, see LDAP integration and authentication.

Local DB

The user name and password in the user record in the instance database.

Digest

The digest token authentication passes user credentials and a digest token within an unencrypted HTTP header. For more information on digest configuration, see Digest token authentication.

Not supported

- SAML 2.0 plugin, however SAML with the Multi-provider SSO plugin is supported.
- Kerberos
- Certificate-based authentication

Storage/Keychain

When you sign in to the app on your mobile device, the app uses your credentials to negotiate an OAuth Token with the instance. The iOS Keychain stores the token and Android uses KeyStore. The keychain encryption is AES 128 in Galois/Counter Mode (GCM).

The mobile app never stores the user password.

Session length and timeout

The session length and timeout is configurable by the organization. For more information on configuring session time out, see mobile session app timeout.

User termination

When an administrator deletes or removes a user from the system, they are logged out of the mobile client.

Mobile data flow for ServiceNow Classic

Data can be retrieved, downloaded from, and written back to a mobile device.

Retrieval

The following describes how data is retrieved from the ServiceNow mobile app.

Read data

When a user requests to view information on the mobile app, the following steps occur.
1. The mobile app sends a request to access data from the instance. The request includes the token and any relevant data field needed for the request.
2. The instance receives the request and checks if the Token is valid.
3. If the token is valid, the request is directed to the relevant API to fetch the information.
4. The information is returned to the mobile app.

**Downloading documents**

When a user requests to download documents from the app, the following steps occur.

1. The mobile app sends a request to access the document. The request includes the Token.
2. The instance receives the request and checks if the Token is valid.
3. If valid, the document becomes available to view or take further actions on the device.

**Write-backs**

The following describes how data is written back from the ServiceNow mobile app.

**Updating fields**

When a user updates a field in the mobile app, the following steps occur.

1. The mobile app sends the Token and the action metadata, for example the ID, or the field to be updated, to the instance.
2. The instance directs the action based on the relevant API.
3. The instance completes the action and sends a response to the mobile app.
4. Based on the response, the mobile app reflects the field changes and action availability in the UI.

**Attaching documents**

When attaching documents, the following steps occur.

1. The mobile app asks the user to attach a document, for example, an image.
2. The mobile app sends the document and Token to the instance.
3. The instance places the document based on the relevant API.
4. The instance sends a response back to the mobile app.

**Internal mobile app distribution**

Internal distribution of the ServiceNow app is supported through all major EMM vendors. Customers are able to pull the app for iOS or Android from the Apple App store and Google Play respectively, dynamically configure the apps to point to the correct ServiceNow instance, and
Mobile navigation and configuration

This way, the MDM can fully manage the app as part of a customer portfolio.

**Note:** ServiceNow does not currently distribute the IPA/APK files, or any other unpublished app to customers as it breaches the Apple Enterprise Developer License Agreement.

Mobile app distribution providers:

- AirWatch Mobile Device Management (MDM)
- BlackBerry Unified Endpoint Management (UEM)
- Citrix XenMobile
- Intune mobile device management (MDM)
- Jamf Pro mobile device management (MDM)
- IBM MaaS360 mobile device management (MDM)
- MobileIron Mobile Device Management (MDM)

Data security for ServiceNow Classic

The ServiceNow app uses SSL/TLS for Over-the-Air communication encryption. The OAuth authorization endpoints are HTTPS.

**Data at rest**

Application preference data such as favorites, home screen, and the mobile navigator items are stored and cached locally on the device. The mobile app does not store record data such as incidents, problems, etc. on the device unless the organization has specifically enabled offline syncing for Field Services. The record data is encrypted with AES 128.

**Data in motion**

Data in motion is over a secure SSL/TLS channel and encrypted with HTTPS.

**Offical access and data cache configuration**

Some field service tables are available to cache locally on the device at the customer’s discretion.

**Push notifications**

Administrators create push notifications and users are able to receive them.

**Cloud**

For more information on the push notification system including process, configuration, and architecture, see Push notification system. Administrators can configure push notification delays using scheduled jobs. To view an example included with the base system, navigate to System Scheduler > Scheduled jobs, then search for a job with the name Push. 5 seconds is the minimum time allowed for the push delay.
Mobile security practices

Mobile security practices include mobile-specific system properties, attachment control, password reinforcement, security patching, and controlling shared data.

Security controls for ServiceNow Classic

Configure security controls to restrict copy/paste, enable biometric controls, enforce passwords, or control attachment functionality.

Restrict copy/paste

Copy/paste restrictions are defined in the system properties (sys_properties) table. There are two applicable security properties:

- glide.ui.m.clear_pasteboard_when_backgrounded: Clears the copy/paste clipboard when the ServiceNow app enters the background
- glide.ui.m.blur_ui_when_backgrounded: Forces the app to blur the screen when the app enters the background on iOS. This property prevents users from being able to take screenshots and also blurs the screen when in app switcher on Android.

Enable biometric controls for ReAuth

Determine whether biometric controls are allowed for approval re-authentication using the following system property in the system properties (sys_properties) table.

- glide.ui.m.auth.allow_biometrics

Enabling this property does not allow users to sign in to the mobile app using biometrics, it is restricted to only work with the e-signature plugin for approvals.

PIN/Password reinforcement

Standard platform password requirements are enforced. Any additional device hardening is the responsibility of the customer.

Attachment control

Use an ACL to block specific access on mobile. Use the isMobile method to check if a request comes from a mobile device. For example, you could add an ACL for the attachment (sys_attachment) table where the read and write scripted ACLs includes the following check. You can also add this code to any existing ACLs you have for the attachment table. If have multiple attachment ACLs, all of the need to have Admin override option unchecked.

```java
if( gs.isMobile() ){
    answer = false;
}
```

Note: You need elevated privileges to create ACLs.
Security patching
In the event a security patch is needed, the mobile development team aligns with standard SDLC properties in order to patch.

User data collection
The mobile app does not specifically collect any user data.

Any user transactions or usage within the app is tracked on the ServiceNow instance just as it is on the web. For user credentials, after a user logs in, the mobile app negotiates an OAuth Token that is stored in the Apple Keychain or the Android Keystore. User credentials are never saved. If the user opts in, the following information is collected:

- Location
- Access to camera
- Notifications

Shared data for ServiceNow Classic
The mobile app communicates with a third party Google program called Fabric for app crash reporting. No customer information is shared.

Incident reporting
Mobile app issues should be reported through the standard support channels. You can report incidents by contacting ServiceNow support.

Mobile configuration for ServiceNow Classic
As an administrator, configure the ServiceNow Classic mobile experience for your users to access an instance on a tablet or smartphone.

Requirements
Configuration for the mobile experience takes place in a desktop browser. Most of the configuration options are available for both the ServiceNow Classic mobile app and the mobile web experience.

Use ServiceNow Classic or the mobile web to test your configuration. Depending on your device, go to the Apple App Store or the Google Play store and search for ServiceNow to download the ServiceNow Classic mobile app.

Role required
admin

Before you begin
Consider the following questions with your stakeholders 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

**Configure the application menu**

Determine which applications and modules you want to appear in the mobile application navigator.

**Customize the mobile home screen**

Customize the appearance of the mobile app or on a mobile web browser by navigating to System Mobile UI > Home Screen. Use the home screen configuration options to customize the mobile experience for each user by role.

**Create a mobile theme**

Add a consistent look and feel to the mobile experience by creating a mobile theme.

**Set up mobile lists**

Mobile devices have less screen real estate so you may need to change the length of titles or the default fields that appear in a mobile list. Customize the mobile list view 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.

**Configure available search fields.**

**Configure Connect Chat for mobile**

Configure the users or roles that can access Connect Chat on a mobile device.

**Next steps**

Most configuration items for the mobile UI are optional. Click any of the topics below for additional configuration.

**Mobile authentication for ServiceNow Classic**

Administrators can configure mobile devices to use different levels of authentication.

Customizing the mobile login page is not supported.

To use multi-provider SSO, the multi-provider SSO plugin must be enabled.

The following authentication options are not supported or have limited support for mobile:

- **SAML 2.0 plugin:** Not supported for mobile.
- **Kerberos:** Native mobile apps do not support Kerberos authentication.
- **Okta:** Using the Okta app to launch the ServiceNow app is not supported. You can, however, launch the native mobile apps using the web version of Okta as long as the multi-provider SSO plugin is enabled.

**ServiceNow Classic multifactor authentication**

Users can access an instance on a mobile device - with either the ServiceNow Classic mobile app or a mobile browser - using multifactor authentication (MFA). The Integration - Multifactor Integration plugin (com.snc.integration.multifactor.authentication) must be active for MFA to apply to mobile authentication.

For more information on configuring multifactor authentication for users, see Configure multifactor authentication.
The mobile web experience has a secondary authentication screen that requests the MFA code. The native mobile apps do not currently prompt for the MFA code on a second screen. The MFA code needs to be appended to the password on the login screen when using the ServiceNow Classic mobile app. For example, P@ssw0rd135642, where 135642 is the MFA code.

Single sign on for the ServiceNow Classic mobile app

The classic mobile app leverages federated login when using the multi-provider SSO plugin. Make sure the Integration - Multiple Provider Single Sign-On Installer (com.snc.integration.sso.multi.installer) plugin is activated. The SAML 2.0 plugin is not supported on mobile. For more information on how to configure multi-provider SSO, see Set up Multi-Provider SSO.

Enable multi-provider SSO

For multi-SSO to work on mobile devices, you need to enable the glide.authenticate.multisso.enabled property.

1. Navigate to Multi-Provider SSO > Administration > Properties.
2. Select the Enable multiple provider SSO check box.
3. Click Save.

SSO and the OAuth token

Once a successful session is established with the federated identity provider and the instance, the ServiceNow Classic mobile application negotiates an OAuth token with the instance. This allows the applications to re-establish connections without storing any user credentials on the device.

The glide.authenticate.sso.redirect.idp system property is not required to use SSO with mobile applications. Mobile applications bypass the local login page and go directly to the SSO/IDP if this property is set.

To enable both local and external login for mobile applications, disable the glide.authenticate.sso.redirect.idp property.

Enable e-signature for approvals for the ServiceNow Classic mobile app

Administrators can configure e-signature to enforce approvals with passwords or biometrics. Role required: admin

E-signature uses the following re-authentication behavior:

- If a user logs in with SSO, e-signature attempts to re-authenticate using the normal SSO authentication flow.
- If a user logs in with a local username and password, e-signature attempts to re-authenticate the user with the local password only, not the username.
- If a user has biometrics, or TouchID, enabled on their device, e-signature attempts to re-authenticate with the user's biometrics.

1. Activate the Approvals with e-signature (com.glide.e_signature_approvals) plugin.
   For more information on activating a plugin, see Activate a plugin.
2. To enable a specific table for e-signature approval, navigate to System Definition > e-Signature Registry, then click New and complete the form.
Authentication using biometrics is enabled by default. To disable biometric authentication, enter `sys_properties.list` in the navigation filter, open the `glide.ui.m.auth.allow_biometrics` property, and set it to false.

Enable an application menu for the ServiceNow Classic mobile app

Define which application menus and modules are available on the mobile application navigator. Enable applications to show new applications in the ServiceNow Classic app or restore applications that were previously removed.

You cannot hide the application navigator from users, however you can control the contents based on role. Use the Roles field for both the application menu and the modules as listed in the following tables to configure the application navigator specifically by role.

1. On a desktop browser, 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</td>
</tr>
<tr>
<td></td>
<td>100 would place this application menu before one with an Order entry of 200.</td>
</tr>
<tr>
<td>Roles</td>
<td>Click the lock icon to select the roles for this application menu. Only users</td>
</tr>
<tr>
<td></td>
<td>with the designated roles can access the modules under this application</td>
</tr>
<tr>
<td></td>
<td>menu.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box to activate this application menu.</td>
</tr>
</tbody>
</table>

4. Save the application menu.

**Note:** Until you add at least one module, the new application menu does not appear on the mobile application navigator.

5. Click **New**.
6. 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</td>
</tr>
<tr>
<td></td>
<td>menu. For example, an entry of 100 would place this module before one with</td>
</tr>
<tr>
<td></td>
<td>an Order entry of 200.</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>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------</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>Enter a custom URL for the module in this format: <code>&lt;type_of_link&gt;/table/parameters</code>.</td>
</tr>
<tr>
<td></td>
<td>· <code>type_of_link</code>: determines what kind of page opens, for example, form, list, or view.</td>
</tr>
<tr>
<td></td>
<td>· <code>table</code>: Refers to the table being referenced. For example, <code>incident</code>.</td>
</tr>
<tr>
<td></td>
<td>· <code>parameters</code>: Any additional information you want to use to direct the user a specific URL. For example, you can include a query as part of the URL to direct users to a specific filtered list.</td>
</tr>
<tr>
<td></td>
<td>You can also use the URL format, <code>$___.do</code>.</td>
</tr>
</tbody>
</table>

| Path Relative To Root | Allows mobile navigator modules that link to desktop pages. Select this check box to use mobile URL in the Path field, such as `$sp.do`, `$chat.do`, or `$vtb.do`. |
|                       | When the Path Relative To Root is cleared, the path is considered a mobile specific path and is relative to `$m.do`. |
|                       | For example, a path like `/form/incident/-1` would be a mobile specific path and would be considered relative to `$m.do`. This would ultimately result in navigating to a path of `$m.do#/form/incident/-1`. |
|                       | Use mobile specific paths when possible and only use Path Relative To Root when there is not a mobile specific path for the desired resource. |

**Note:** If you are creating a module for a map page, see [Create a smartphone map page module](#).

7. Click Submit.

Customize lists and forms for the ServiceNow Classic mobile app

In order to optimize views on smaller screens, lists and forms display differently on a mobile device. Use any of the following options to customize different aspects of mobile lists.

**Customize a mobile list**

Customize which fields display in a list on a mobile device using table titles.
Configure field styles
Field styles allow you to set individual styles for each item in a list. You can configure field styles for a mobile list that are separate from the desktop view.

Configure the navigation behavior when a user taps an item in a list
Determine whether tapping an item in a list opens the record or the activity stream for an item.

Configure how a form displays on a mobile device
Limit the number of fields on a form for a better mobile experience.

Hide mobile filters
Hide the filter option or the activity stream from users to prevent them from filtering out search results.

ServiceNow Classic mobile list view
The information that appears in a mobile list is different than a list on a desktop. Smaller screen real estate limits the information that is easily viewable. You can control mobile list text using the table titles module.

Use Table Titles to control the information that appears in a mobile list. The information that appears in a list item is made up of the following components.

- **Title**: Appears prominently at the top of the list item. You can configure the list item title by selecting the fields that appear as the title. For a more advanced configuration, you can use a script to determine how and what displays as the list item title.
- **Display title**: Appears as subtext beneath the title for a list item. You can customize the display title by using the configure dictionary option for a field on a form.
- **Extras**: Any extra information you want to appear in the list. Extras are only configurable using a script.
Customize the list title for the ServiceNow Classic mobile app

The information that appears in a mobile list is different than a list on a desktop. Smaller screen real estate limits the information that is easily viewable. You can control mobile list text using the table titles module.

1. In the desktop interface, navigate to System Mobile UI > Table Titles.
2. Click New, or select an existing table to edit.
3. If you are creating a new list, select the table from the list. The list shows only tables and database views that are in the same scope as the title.

4. Use the Fields slushbucket to add items that appear in the list item. The order of the items in the selected column determines the order they appear in on a mobile device.

Important:
Table titles define the title appearance for a list on a mobile device as well as in the split pane view on the desktop interface. You can change the title for any table or add a title record for a different table. However, the change affects both mobile lists and the list card in the split pane view.

For example, the default title for the Incidents (incidents) table is the short description. If you change the table title to display the priority field instead, the priority field appears as the table title for incidents on a mobile device as well as in the list card in the split pane view.

5. Click Submit.
This example shows a list of incident (incident) records with the Short Description, Assigned to, and Updated fields included.

Script items in a list for the ServiceNow Classic mobile app
For a more advanced option for configuring the information that appears in a mobile list, you can create customized scripts.

Role required: admin

Use scripts to control the following components in a mobile list.
- Title: Use the `setTitle()` method to script a title. The list title is connected to a specific table in the system, for example Problem. Each table contains a list of field options, defined by `field_list` that you can use to determine the title.
- Extras: Use the `setExtras()` method with an array of strings to configure the information that displays after the list title. Scripting extras adds new rows to the list items.

1. Navigate to **System Mobile UI > Table Titles**.
2. In the **Script** field, create a script to define which fields appear in the mobile list.

   For example, the following code block uses `setTitle` and `setExtras` to configure a mobile list for the problem table. The title shows the short description for a problem. The extras include additional information for the problem priority.

   ```javascript
   var title = current.short_description;
   var extras = [];
   if (current.priority === 1){
       extras.push("Critical!");
   }
   else if(current.priority === 2) {
       extras.push("High Priority!");
   }
   else {
       extras.push("Carry on");
   }
   titleValues.setTitle(title);
   titleValues.setExtras(extras);
   ```

3. Click **Submit**.

Add a display title for the ServiceNow Classic mobile app

Use the display title option to add a subtitle to list items on a mobile device.

Role required: admin

1. Navigate to the table you want to configure the display title for. For example, Problem.
2. Right-click the field you want to display and click **Configure Dictionary**. For example, Urgency.
3. Select the **Display** check box.

   You can only have one display title for a table. The system automatically clears the selection for any previously selected display titles when you select a new one.

Configure field status indicators for the ServiceNow Classic mobile app

Differentiate items in a list by configuring the field status indicators.
Define field styles in the **System UI > Field Styles** module. For more information on how to customize field styles, see *Define field styles*.

Role required: admin

1. Navigate to **System Mobile UI > Table Titles**.
2. Select a table to add field status indicators to.
3. From the **Style Field** list, select the styled field you want to appear with the list on a mobile device.

**Note:** If the **Style Field** does not appear on the form, you may need to configure the form.

---

Configure list item navigation for the ServiceNow Classic mobile app

Determine whether a tapping an item in a list opens the record or the activity stream using a system property.

Role required: admin

1. Navigate to **System Properties > Mobile UI properties**.
2. From the **Destination when navigating to a record from a list** property (glide.ui.m.default_record_navigation), select **Activity Stream** or **Form**.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity Stream</td>
<td>Opens the list item in the activity stream</td>
</tr>
</tbody>
</table>

How do I create a sub-folder

Number: INC0000017  
Category: Inquiry / Help  
Priority: 1 - Critical  

Related Lists

Participants

Aug 25, 2015

Don Goodliffe  
I got this  

Don Goodliffe  
Task reassigned due to employee termination
ServiceNow Madrid Mobile navigation and configuration

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form</td>
<td>Opens the list item in the form</td>
</tr>
</tbody>
</table>

Configure list search fields for the ServiceNow Classic mobile app

Configure search fields to determine what fields are searched for in the table. If search fields are not configured, only the columns in the mobile list layout are searched.

1. Navigate to System Mobile UI > Table Titles.
2. Open the table you want to configure search fields for, or click New.
3. Move fields from the Available list to the Selected list to add fields to the mobile list.
Note: The Search fields field might not appear on the Table Title form. Add Search Fields to the form by editing the form layout.
<table>
<thead>
<tr>
<th>Table: Incident (incident)</th>
<th>Application: Global</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fields: Short description, Number, Category</td>
<td></td>
</tr>
<tr>
<td>Updated: 31-May-2016 15:03:34</td>
<td></td>
</tr>
</tbody>
</table>

### Search Fields

<table>
<thead>
<tr>
<th>Available</th>
<th>Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>Number</td>
</tr>
<tr>
<td>Activity due</td>
<td>State</td>
</tr>
<tr>
<td>Actual end</td>
<td>Category</td>
</tr>
<tr>
<td>Actual start</td>
<td></td>
</tr>
<tr>
<td>Additional assignee list</td>
<td></td>
</tr>
<tr>
<td>Additional comments</td>
<td></td>
</tr>
<tr>
<td>Approval</td>
<td></td>
</tr>
<tr>
<td>Approval history</td>
<td></td>
</tr>
<tr>
<td>Approval set</td>
<td></td>
</tr>
</tbody>
</table>
Configure the list layout for the ServiceNow Classic mobile app

Configure and order the fields displayed in a list view in the ServiceNow Classic mobile UI. Ensure that your users have the most relevant information when browsing on their mobile devices.

1. From a desktop browser, navigate to any list 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. Move fields from the Available list to the Selected list to add fields to the list.
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.

**List layout results**

<table>
<thead>
<tr>
<th>Record field</th>
<th>Sortable column</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Caller First name</strong></td>
<td><strong>Margaret</strong></td>
</tr>
<tr>
<td><strong>Title</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Number</strong></td>
<td>INC0000053</td>
</tr>
<tr>
<td><strong>Category</strong></td>
<td>Request</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>1 - Critical</td>
</tr>
</tbody>
</table>

Related Lists

Participants

Oct 20, 2015

System Administrator 8 mos

© 2019 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.
Hide filters for the ServiceNow Classic mobile app

Prevent users from filtering content in a list by hiding the filter.

Role required: admin

1. Navigate to the table you want to hide filters for.
2. Open the list configuration page by performing the appropriate action for the list version. For more information on configuring list controls, see Configure list controls.

<table>
<thead>
<tr>
<th>Version</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>List v2</td>
<td>Right-click any column heading and select Configure &gt; List Control.</td>
</tr>
<tr>
<td>List v3</td>
<td>Open the list title menu and select List Control.</td>
</tr>
</tbody>
</table>

3. Select the **Omit filters** check box.
   This field is only available for standard lists.

Customize default mobile home pages for the ServiceNow Classic mobile app

Administrators can create customized home pages for their users on the ServiceNow Classic mobile app. Administrators can customize mobile home pages by section and module. Depending on the level of customization required, administrators can limit user access by role to each component of the mobile home page. Mobile home pages apply to both the ServiceNow Classic mobile app and the mobile browser.

Role required: admin

Use the following components to create a customized home page:

- Home Page Collections
- Home Pages
- Home Page Section
- Home Page Modules

Each component can be configured separately or using the following steps.

1. Navigate to **System Mobile UI > Home Screen > Home Page Collections**.
2. Click **New** or open an existing Home Page Collection. For more information on completing the collection fields, see Home Page Collections.
   If you want to configure an entire homepage to be role specific with no default modules, you can create a new Home Page Collection. Otherwise, simply open an existing collection to configure the page.
3. From the related list on the Home Page Collection form, click **Edit...** to add an existing Home Page or click **New**.
   For more information on completing the Home Page form, see Mobile Home Pages form.
4. From the related list on the Home Page form, click **Edit...** to add an existing Home Page Section to the Home Page, or click **New**.
   For more information on completing the Home Page Section form, see Mobile Home Page Sections.
5. To add modules to a homepage, from the related list on the Page Section form, click **Edit...** to add an existing module, or click **New**.
   For more information on completing the Home Page Modules form, see Mobile Home Page Modules.
Home Page Collections

Use the Home Page Collections as a starting point for your home page customization.

Administrators can configure Home Page Collections by role so only specific users can see a particular home page. For example, you can configure an ITIL user homepage to only have modules that are relevant to the ITIL user. However, if you have default apps that different roles use, you can also create a basic homepage, with the option of customize specific pages by roles.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the collection. This field won’t appear in the app or on the mobile web.</td>
</tr>
<tr>
<td>Priority</td>
<td>The numbered priority of the collection. Set a higher priority than the default collection if you want the homepage collection to show up for specific roles. For example, if the default is set to 500, set the role specific collection to 200.</td>
</tr>
<tr>
<td>Hide Favorites Page</td>
<td>Users can configure their own homepage by favoriting items in the system. The favorites page is accessible with a right swipe. Administrators can disable the favorites page by selecting this option.</td>
</tr>
<tr>
<td>Roles</td>
<td>Use this option to configure home page collections for specific roles.</td>
</tr>
</tbody>
</table>

Mobile Home Pages form

Configure the look and feel of mobile home pages.

Administrators can configure more than one home page for their users. Each additional home page is accessible to users by swiping to the left from the main home page. The sort order, accessible from the Home Page Collections related lists, defines the Home page order. Administrators can also configure home pages by role.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Title of the home page. The title appears on the home page.</td>
</tr>
<tr>
<td>Title position</td>
<td>Determines the location of the title in the app or mobile web. Selecting Navigation Bar overrides the theme configuration and adds the title in place of the main header bar. Use Hidden to hide the title.</td>
</tr>
<tr>
<td>Subtitle</td>
<td>Create an additional title to describe the page. Subtitle appears below the title or header image.</td>
</tr>
</tbody>
</table>
Mobile Home Page Sections
Administrators can configure Home Page Sections to determine how apps appear in a mobile home page.

Mobile home page section fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Title of the home page section. This appears in the app or mobile home page above the list of modules.</td>
</tr>
<tr>
<td>Hide Title</td>
<td>Hide the home page section title so it doesn’t appear on the page.</td>
</tr>
<tr>
<td>Module Style</td>
<td>How the modules appear in the section. <strong>Regular</strong> shows the modules as rectangles that stretch the length of the app or mobile web. <strong>Compact</strong> displays the modules as smaller “app-like” squares.</td>
</tr>
<tr>
<td>Roles</td>
<td>Configure Home Page Sections by role.</td>
</tr>
</tbody>
</table>
Mobile Home Page Modules

Home Page Modules define what information appears on a mobile home page. Modules must be enabled for mobile to appear on a mobile home page. Administrators can also configure Home Page Modules by role.

Mobile Home Page Module fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Name of the module that appears in the app or mobile web.</td>
</tr>
<tr>
<td>Subtitle</td>
<td>Optional module description that appears beneath the title.</td>
</tr>
<tr>
<td>Module</td>
<td>Select a module from the list of available mobile modules</td>
</tr>
<tr>
<td>Active</td>
<td>Whether the module is available or not. Clear this check box to disable mobile access to a mobile module.</td>
</tr>
<tr>
<td>Roles</td>
<td>Configure module access by role. Users without the specific role won’t see the module.</td>
</tr>
<tr>
<td>Icon</td>
<td>Select an icon from the list of available icons.</td>
</tr>
<tr>
<td>Image...</td>
<td>Upload your own module image</td>
</tr>
<tr>
<td>Background color</td>
<td>Select a background color for the module. Use color name or Hex color. For example, #81878e or light gray.</td>
</tr>
<tr>
<td>Content Style</td>
<td>Select a style from available themes.</td>
</tr>
</tbody>
</table>

Configure a theme for the ServiceNow Classic mobile app

Configure a theme for the ServiceNow Classic mobile UI to make the experience more consistent for your users. Theme changes apply to both the ServiceNow Classic mobile app and the mobile web experience.

Role required: admin

1. Navigate to System Mobile UI > Themes.
2. Click an existing theme to edit, or click New.
3. Complete the mobile theme form fields.
Field | Description
--- | ---
Name | Title for the theme. This does not appear anywhere for end users.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base theme</td>
<td>The underlying theme to build your styles on. Use the base theme to determine if the main background is darker or lighter.</td>
</tr>
<tr>
<td>Navigation bar title</td>
<td>A title that appears in the navigation bar of the app or mobile web page. For example, your company name or the group you are creating the app for. The navigation bar title also appears at the title when you save a mobile web shortcut on your phone.</td>
</tr>
<tr>
<td>Title image</td>
<td>An image that appears below the navigation bar. The title image overrides the navigation bar title.</td>
</tr>
<tr>
<td>Theme Colors</td>
<td>Navigation bar and toolbar colors for the native app or mobile web. Use color names or Hex codes. For example, red or #ff0000.</td>
</tr>
</tbody>
</table>

4. Click **Activate Theme**. The sysID for the new theme appears in the Set the active theme for the mobile experience field on the mobile UI properties page. You can change the theme the ServiceNow Classic mobile app uses by changing the sysID in this field.
Configure Connect Chat for the ServiceNow Classic mobile app

Configure Connect Chat to show or hide on the ServiceNow Classic mobile app homepage. Connect Chat appears on the ServiceNow Classic mobile app homepage by default if it is active on the instance and the user has the correct roles. Hide Connect Chat in the mobile app by disabling the Connect module in the mobile homepages.

Connect Chat has only been optimized for the ServiceNow Classic mobile app, not the mobile web.

1. Navigate to System Mobile UI > Home Screen > Home Page Modules.
2. Open the Connect home page module.
3. Clear the Active check box.
   You can also use the Roles field to configure access to Connect Chat by role.

Configure the appearance of the Connect Chat homepage module. For more information on configuring home page modules, see Mobile home page module fields.
Mobile UI actions for the ServiceNow Classic mobile app

UI actions function the same on the mobile interface as on desktop interface, but are configured on a different table: UI Action - Mobile (sys_ui_ng_action).

You can use the following types of UI actions in the mobile interface:

- List buttons
- Form buttons
- Form more items (items that display when the user taps the More button)

<table>
<thead>
<tr>
<th>UI Actions - Mobile (UI Actions - Mobile)</th>
<th>Name</th>
<th>Active</th>
<th>Order</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>*sysapproval_approver</td>
<td>Search</td>
<td>Search</td>
<td>Search</td>
<td>Search</td>
</tr>
<tr>
<td>sysapproval_approver Reject</td>
<td>true</td>
<td>100</td>
<td>current.state == 'requested' &amp; (gs.hasR...</td>
<td></td>
</tr>
<tr>
<td>sysapproval_approver Approve</td>
<td>true</td>
<td>200</td>
<td>current.state == 'requested' &amp; (gs.hasR...</td>
<td></td>
</tr>
</tbody>
</table>

UI action

The lowest number in the Order column identifies the primary button at the top of the form.

**Note:** Mobile UI actions automatically reload the form when an action is submitted. You do not need to configure a UI action to make this happen.

Back navigation in the ServiceNow Classic mobile app and mobile web 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.
Enable client scripts for the mobile browser for the ServiceNow Classic mobile app

Control whether client scripts for forms run on the mobile interface, the desktop UI, or both. Use this option to create scripts specific to one interface, or prevent long running from executing on the more limited hardware of a mobile device.

1. On a desktop, navigate to System Definition > Client Scripts.
2. Create a client script that is compatible with the mobile interface. To ensure compatibility, see Mobile client GlideForm (g form) scripting and migration.
3. In the UI Type field, select Mobile/Service Portal or All to have the script run on the mobile interface.
   If the UI type field is not available on the form, configure the form to add the UI Type field.
4. Fill in the fields, as appropriate.
5. Click Submit.
Mobile client GlideForm (g form) scripting and migration

Client scripting for mobile is identical to scripting for the web, with some exceptions. All new scripts must conform to certain guidelines. The following items 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.

**Requirements**

<table>
<thead>
<tr>
<th>Task</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use the new mobile methods</strong></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.</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.getElement()`  
  
  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. |
| **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. |
<table>
<thead>
<tr>
<th>Do not make synchronous JavaScript calls</th>
<th>The mobile platform does not allow synchronous JavaScript calls. The <code>g_form.getReference()</code> method must now have the callback parameter defined. For example:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>g_form.getReference(fieldName, callback)</code></td>
</tr>
<tr>
<td></td>
<td>Be sure that all <code>g_form.getReference()</code> calls include the callback parameter. For example, the following script does not include a callback and is incompatible with the mobile platform:</td>
</tr>
<tr>
<td></td>
<td><code>var userName = g_form.getReference('assigned_to').user_name;</code></td>
</tr>
<tr>
<td></td>
<td><code>g_form.setValue('u_assigned_user_name', userName);</code></td>
</tr>
<tr>
<td></td>
<td>The following script has been updated to include the callback and is compatible with the mobile platform:</td>
</tr>
<tr>
<td></td>
<td><code>g_form.getReference('assigned_to', function(gr) {</code></td>
</tr>
<tr>
<td></td>
<td><code>g_form.setValue('u_assigned_user_name', gr.user_name);</code></td>
</tr>
<tr>
<td></td>
<td>});</td>
</tr>
<tr>
<td>Do not make synchronous Ajax calls</td>
<td>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="https://servicenow.com">AJAX</a>. For information on the available GlideAjax methods, refer to the .</td>
</tr>
</tbody>
</table>
Do not make synchronous GlideRecord calls

The mobile platform does not allow synchronous 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:

```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);
});
```

Do not use methods unavailable on the mobile platform

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:

- `showRelatedList ()`
- `hideRelatedList ()`
- `showRelatedLists ()`
- `hideRelatedLists()`
- `flash()`
- `getSections()`
- `enableAttachments()`
- `disableAttachments()`
- `setReadOnly()` (Note that `setReadOnly()` is available)
- `getParameter()`

Enable scripts for mobile

Scripts must be enabled for the mobile platform. See Enable client scripts for the mobile browser for the ServiceNow Classic mobile app.

**Note:** Focusing an element on a mobile form is not supported.
UI policies

Use the Run scripts in UI type field to determine whether scripts run on the mobile platform, the desktop, or both. 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 mobile browser.

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 Enable an application menu for the ServiceNow Classic mobile app.

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 Mobile UI actions for the ServiceNow Classic mobile app.

Enable UI policies for the mobile browser

Define UI policies to run on forms on the mobile 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 mobile interface. To ensure the UI policy is compatible on a mobile device, see mobile UI policies.
3. Configure the UI Policies form to add the Run scripts and the Run scripts in UI type fields.
4. Select the Run scripts check box.

Note: The Run scripts in UI type field does not appear unless the Run scripts check box is selected.

5. In the Run scripts in UI type field, select Mobile/Service Portal or All to have the policy run on the mobile interface.
6. Complete the fields, as appropriate.
7. Click **Submit**.

Configure 3D Touch options for iOS for ServiceNow Classic

Favorites determine the options that appear on the 3D Touch option for the ServiceNow app on iOS. You can configure your own options for your users using the Shortcut Items module.

Role required: admin

3D Touch allows users to access shortcuts to items in an app with a hard press on the app. 3D Touch is only available on the iPhone 6s and newer.

1. Navigate to **Mobile App > Shortcut Items**.
2. Click **New**.
3. Complete the fields on the form.

### Shortcut item fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Name for the shortcut that appears in the 3D Touch menu. For example, Create new incident.</td>
</tr>
<tr>
<td>Subtitle</td>
<td>Optional secondary title for the shortcut that appears in the 3D Touch menu. For example, Create a new incident if there's a problem.</td>
</tr>
<tr>
<td>Icon</td>
<td>A visual indicator for the shortcut.</td>
</tr>
<tr>
<td>Module</td>
<td>Module that opens when a user taps the shortcut. The module must be configured to appear in a mobile device. For example, the create new incident module.</td>
</tr>
<tr>
<td>Order</td>
<td>The order in which shortcuts appear. Lower numbers appear in the list first. Configured shortcuts also appear before user favorites in the 3D Touch menu.</td>
</tr>
<tr>
<td>Roles</td>
<td>Limit the users who can access the 3D Touch menu item by role.</td>
</tr>
</tbody>
</table>
Enable mobile location and barcode scanning for the ServiceNow Classic mobile app

Take advantage of mobile devices by allowing location and barcode scanning in the ServiceNow Classic mobile app.
Role required: admin

1. On a desktop, 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. From the View name list, select Mobile.
5. In the Create new field section, 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 need to allow the app to access their location and camera on their devices. These fields only appear when you create a record, they do not appear in one that is already existing.

The native apps for iOS and Android support scanning the following barcode types:
- 2D barcodes: QR Code, Data Matrix, PDF-417, AZTEC

Disable mobile access for the ServiceNow Classic mobile app

Administrators can disable or enable different parts of the mobile UI.

Navigate to System Properties > Mobile UI Properties to enable or disable any of the properties below.

<table>
<thead>
<tr>
<th>UI Agent Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Enable mobile web UI</td>
</tr>
<tr>
<td>Enable native mobile applications (Android, iOS)</td>
</tr>
<tr>
<td>Use the mobile web UI if one of these strings (comma-separated) appears in the browser's User Agent</td>
</tr>
</tbody>
</table>

Get started with the ServiceNow Classic app

Use your smartphone or tablet to access an instance in a web browser or the ServiceNow Classic mobile app.

Before you begin

You can access your instance from a web browser on your mobile device or from the ServiceNow Classic mobile app. Depending on your device, go to the iTunes Store or the Google Play store.
and search for ServiceNow to download the native mobile 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.

In the ServiceNow Classic mobile app, add a visualization to the favorites on your homepage.

**Use lists 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.

Search for specific records in a mobile list.

Use Mobile lists for ServiceNow Classic on a tablet.

**Use Service Catalog, Connect Chat, location, and barcode scanning**

Use mobile Service Catalog to order materials.

Collaborate and stay connected with other users using Connect Chat.

Allow the app to access your location and camera to check in or scan barcodes.

**Manage your notification settings**

Use Notification Settings to enable or disable the notifications that you receive and the channels for receiving them.

**Next steps**

The mobile UI varies depending on your device, the way your administrator configured the mobile UI for your company, and whether you access your instance from the ServiceNow Classic mobile app or mobile browser.

**Log in on a mobile device**

You can use the URL for your instance to log in using a mobile device. Access your instance from a mobile web browser or using the ServiceNow Classic mobile app.

Depending on your device, go to the iTunes Store or the Google Play store and search for ServiceNow to download the native mobile app.

**Access an instance from the ServiceNow Classic mobile app**

Access an instance using the ServiceNow Classic mobile app for supported devices.

1. After you download the app from the iTunes store or the Google Play store, tap the icon on the homepage.
You do not need to include service-now.com at the end of the instance name.

Even with the newest version of the app, you can access any instance version as far back as Geneva patch 8. Some functionality is version specific so if you do access an older version you might have more limited access. For example, only instances starting with Istanbul have access to customizable mobile home pages.

2. Use your regular login to sign in to the app.
After you log in, the app remembers your information. As long as you do not 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. Select the instance from the History list to log back in.

**Note:** The ServiceNow Classic mobile app does not have a screen for multi factor authentication (MFA). If your organization uses multi factor authentication, append the MFA code to your password on the login screen. For example, P@ssw0rd642135, where 642135 is the MFA code.

Access an instance from a mobile browser

Access an instance using a supported mobile browser.

On a **supported device**, navigate to the URL of an instance in a supported mobile browser. Devices are automatically detected and the interface appends $m.do to the end of the URL. Accessing the mobile interface does not prevent the user from also accessing the tablet or desktop interfaces.
1. Enter the base URL in any supported mobile web browser.
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 [Change settings for the Remember me check box and cookie](#).
4. Optional: If you forget your password, tap **Forgot password?**. The system walks you through a verification process to confirm your information.

5. Tap **Continue** to complete your login.

   If your organization uses multi factor authentication (MFA), the web login includes an additional screen to input the MFA code.

---

**Add a shortcut to the home screen**

Add an Apple home screen shortcut that opens a mobile browser page.

1. In Safari on a mobile device, 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 **Add to Home Screen** icon.

4. Type 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 browser home screen shortcuts in the mobile 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.
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. Select the following options.
   - 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, you can change the default font size to a smaller size. 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 is difficult for some users to read.

Mobile application navigator in the ServiceNow Classic app

The application navigator 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.

The application navigator also provides access to favorites and recently viewed items.

Administrators need to customize the application navigator so that the required modules appear on a mobile device. Most modules have not been enabled for mobile. For more information, see **Enable an application menu for the ServiceNow Classic mobile app**.

Use the application navigator in the ServiceNow Classic app

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.
Use the tablet application navigator in the ServiceNow Classic app

Access different modules in the user interface using the application navigator just as you do in the mobile app.

1. To open the application navigator, tap the menu icon ( ).
2. Scroll the navigator up and down using your fingers. Tap a menu application to open.
To close the navigator, tap outside of the window.

**Mobile lists in the ServiceNow Classic app**

Lists appear as a single column of records on the mobile app UI display. Each row represents a separate record.

Live list updates are not available on mobile devices. Mobile lists only automatically reload when a record change is made by the currently logged in user. The list does not automatically update when other users make changes.
<table>
<thead>
<tr>
<th>Mobile UI lists for iPhone</th>
<th>Mobile UI lists for Android</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unable to get to network file shares</strong></td>
<td><strong>Unable to get to network file shares</strong></td>
</tr>
<tr>
<td>INC0000002</td>
<td>INC0000002</td>
</tr>
<tr>
<td><strong>Wireless access is down in my area</strong></td>
<td><strong>Wireless access is down in my area</strong></td>
</tr>
<tr>
<td>INC0000003</td>
<td>INC0000003</td>
</tr>
<tr>
<td><strong>Need access to sales DB for the West</strong></td>
<td><strong>Need access to sales DB for the West</strong></td>
</tr>
<tr>
<td>INC0000007</td>
<td>INC0000007</td>
</tr>
<tr>
<td><strong>I can't launch my VPN client since the last software update</strong></td>
<td><strong>I can't launch my VPN client since the last software update</strong></td>
</tr>
<tr>
<td>INC0000015</td>
<td>INC0000015</td>
</tr>
<tr>
<td><strong>Rain is leaking on main DNS Server</strong></td>
<td><strong>Rain is leaking on main DNS Server</strong></td>
</tr>
<tr>
<td></td>
<td>INC0000016</td>
</tr>
<tr>
<td><strong>How do I create a sub-folder</strong></td>
<td><strong>How do I create a sub-folder</strong></td>
</tr>
<tr>
<td>INC0000017</td>
<td></td>
</tr>
</tbody>
</table>
Mobile lists consist of the following elements.

**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 in the ServiceNow Classic

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.

Rendering HTML text in activity streams is not supported for the mobile apps.
iPhone list activity stream
List activity stream for Android

Elements of the mobile UI: activity stream

© 2019 ServiceNow, Inc. All rights reserved. ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.
Activity stream updates consist of the following elements.

**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 in the ServiceNow Classic app

Use the condition builder to create complex filters in the native mobile app.

Not all filters are available on the mobile app. Saving a mobile filter is not supported. You can, however, add a filtered list to the favorites menu by selecting the star icon, which saves the list with the filter.

<table>
<thead>
<tr>
<th>iPhone UI filter</th>
<th>Android UI filter</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="iPhone UI filter" /></td>
<td><img src="image" alt="Android UI filter" /></td>
</tr>
</tbody>
</table>

Mobile filters consist of the following elements.
### Mobile UI filters

<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>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>

**Mobile condition sets**

Condition sets generate a list by combining the results of multiple sets of conditions. Use condition sets 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.

**Sample list filter with two condition sets**
In this example, set 1 requires matching all the conditions and returns 8 results. Set 2 allows matching to any condition and returns 29 results. Total results return 37 records for the combination of record results from the individual condition sets.

View or modify a mobile list in the ServiceNow Classic app

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.
Search a mobile list in the ServiceNow Classic app

Search in a list on a mobile device.

1. From the mobile homepage or the navigation menu, tap the list you want to search within.
2. In the search field at the top of the screen, type your search terms. On an Android device,
<table>
<thead>
<tr>
<th>iPhone search field</th>
<th>Android search field</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Incidents</strong></td>
<td><strong>Active</strong></td>
</tr>
<tr>
<td><img src="image" alt="Search field" /></td>
<td><img src="image" alt="Activity Stream" /></td>
</tr>
<tr>
<td><img src="image" alt="Search icon" /></td>
<td><img src="image" alt="Sort" /></td>
</tr>
<tr>
<td><img src="image" alt="Star icon" /></td>
<td><img src="image" alt="Share" /></td>
</tr>
</tbody>
</table>

- **Manager can't access SAP Controlling application**
  - INC0000051
  - Category: Software

- **Can't access SFA software**
  - INC0000046
  - Category: Software

- **Please remove the latest hotfix from my PC**
  - INC0000027
  - Category: Software

- **Can't launch 64-bit Windows 7 virtual machine**
  - INC0000019
  - Category: Software

- **SAP Materials Management is slow or there is an outage**
  - INC0000054
  - Priority: 1 - Critical

- **The SAP HR application is not accessible**
  - INC0000053
  - Priority: 1 - Critical

- **SAP Sales app is not accessible**
  - INC0000055
  - Priority: 1 - Critical

- **Manager can't access SAP Controlling application**
  - INC0000051
  - Priority: 1 - Critical
3. To change the sort order, tap **Sort**.

**Mobile lists for ServiceNow Classic on a tablet**

Lists appear as a single column of records on the mobile tablet UI display. Each row represents a separate record.

Tap a record in the column on the left to see item details.
Mobile lists on a tablet consist of the following elements.
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 and add a record.</td>
</tr>
<tr>
<td>4</td>
<td>Application search</td>
<td>Displays records that match the search string.</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>
<tr>
<td>8</td>
<td>Record details</td>
<td>View record details in a split pane view.</td>
</tr>
<tr>
<td>9</td>
<td>Record UI actions</td>
<td>Make a particular record a favorite, share the record, add an attachment, work notes, or comments.</td>
</tr>
</tbody>
</table>

Mobile activity streams in the ServiceNow Classic app for a tablet
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 in the same panel as the list. Select an item to see more details in the right panel.

Elements of the tablet UI: activity stream
Issue with email

Spoke to customer, appears to need a new VPN token.

Joe Employee 2 INC0000047 • 4 mos

Issue with email

Will deliver new token tomorrow when I'm in the San Diego office.

Joe Employee INC0000047 • 4 mos

Issue with email

Can't connect to Email from home. Webmail appears to be down.

Joe Employee INC0000047 • 4 mos

Issue with email

Can't connect to Email from home. Webmail appears to be down.

Joe Employee INC0000047 • 4 mos

Spoke to customer, appears to need a new VPN token.
Activity stream updates consist of the following elements.

**Tablet UI elements: activity stream**

<table>
<thead>
<tr>
<th>Number</th>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Back button</td>
<td>Return to the list.</td>
</tr>
<tr>
<td>2</td>
<td>User</td>
<td>Displays the user photo and name of the user who made the activity update.</td>
</tr>
<tr>
<td>3</td>
<td>Record details</td>
<td>Displays the record number and relative time since the update was made.</td>
</tr>
<tr>
<td>4</td>
<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.

*Filters for ServiceNow Classic on a tablet*

Construct complex filters with the ServiceNow Classic mobile app UI.
Mobile filters consist of the following elements.

Mobile UI filters

<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>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 ServiceNow Classic mobile app.

Mobile favorites in the ServiceNow Classic app

Mobile favorites provide links to records in the system. Favorites display as icons on the homepage in the ServiceNow Classic mobile app.
**Note:** There are a few limitations for mobile favorites:

- You cannot disable the favorite icon on mobile lists or forms.
- **Fixed Query** modules are not supported and do not appear on a mobile device even when selected as a favorite.
- Favorites containing `home.do` are not supported and do not appear on mobile devices.

<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>

Long press a favorite to edit the following properties.

- Label
- Text and icon color
- Icon image
- Enable visualizations for some favorites

Favorites automatically synchronize between the mobile UI and the desktop UI for your instance.

**Add a mobile favorite**

Make any list, record, or module a favorite to add it as an icon to the mobile favorites page.

1. Navigate to the screen that you want to add as a favorite.
2. Tap the star icon.
3. From the Customize Favorite page, give the favorite a name, tag color, and icon. Then tap **Save**.
Edit a mobile favorite

Make changes to the appearance of a favorite on a mobile favorites 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.

Add a Visualization to a favorite

Add visualizations to mobile favorites.
Visualizations are specific to your device. If you add a visualization to a list on one mobile device, it won't display the same way on another device.

1. In the mobile app homepage, long press the favorite you want to add a visualization to.
2. Tap the **Visualizations** tab.
3. Select a visualization from the available options. Visualization options vary depending on the type of favorite you select.
Visualizations tab
Visualization options

Visualizations allow users on the mobile app UI to see relevant table information directly from the home page favorite icon.

Use the following visualizations to customize the appearance of favorites on the mobile app homepage.

**Favorite visualization options**

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Favorite icons displays the number of records returned by the list filter.</td>
<td><img src="image" alt="Count" /></td>
</tr>
<tr>
<td>Favorite icons displays the number of records with a particular field value. Select the field values you want to display when you set up the visualization.</td>
<td><img src="image" alt="Bar chart" /></td>
</tr>
</tbody>
</table>

Using the ServiceNow Classic mobile app

Access an instance from the mobile interface using the ServiceNow Classic mobile app.

Download the ServiceNow Classic mobile app from the iTunes store for devices running iOS 9 and above or from the Google Play store and for Android phones running version 4.4 (KitKat) and above.
Use the ServiceNow Classic mobile app to do the following:

- **Access lists and forms.**
- Save [favorite lists and records](#) to the app homescreen.
- Access the Service Catalog from your mobile device.
- Communicate with other users within the platform using [Connect Chat](#) for mobile.
- Share your [location](#).

The mobile UI is configurable by your administrator.

### Manage notification settings in the ServiceNow Classic mobile app

In the ServiceNow Classic mobile app, use **Notification Settings** to enable or disable your notifications and the channels for receiving them.

Role required: user

You can set additional notification preferences, such as conditions or filters that affect notification delivery, through the System Settings on a desktop instance or mobile web browser. For details, see [Setting notification preferences in UI](#).

1. On your profile screen, tap **Notification Settings**.
2. On the Notifications Settings screen, enable or disable your notifications.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Notifications</td>
<td>Global switch for enabling or disabling all your notifications.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The system does not disable any notifications configured as mandatory by your administrator.</td>
</tr>
<tr>
<td>Notification channels</td>
<td>Switches for enabling or disabling the notifications that you receive by device.</td>
</tr>
</tbody>
</table>
| Notification categories | List of notification categories. Each category identifies and groups related notifications. To select a notification:  
  1. Tap the notification category.  
  2. Tap the notification to be updated.  
  3. Enable or disable the channels for that notification. |

3. Tap the back button (arrow) to navigate back to your profile screen.

### Mobile Connect Chat in the ServiceNow Classic app

Use Connect Chat to communicate with coworkers in the ServiceNow Classic mobile app.

Activate Connect before using chat in the ServiceNow Classic mobile app. Many, but not all the features supported in the desktop interface are available in the mobile app UI. Administrators can show or hide Connect Chat by configuring it by role.

Connect Chat is not supported on the mobile web although some limited functionality exists.
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. Chat 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

Have conversations with groups of people in Connect in the ServiceNow Classic mobile app.
Have you seen this?

ITIL User

System Administrator has been added to the group

System Administrator

that's not your car is it??

ITIL User

No! So glad I parked on the street today!

incident.do?
sys_id=23d487264f8c12002fa02f1e0210c746&sysparm_record_target=incident&sysparm_record_row=2&sysparm_record_fields=32&sysparm_record_list=active %3Dtrue%5EORDERBYDESC

Parking garage flooded!
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>
<tr>
<td>Conversation details</td>
<td>Opens the conversation details page, which displays the following information.</td>
</tr>
<tr>
<td></td>
<td>• Record details (record conversations only)</td>
</tr>
<tr>
<td></td>
<td>• Push notification preferences</td>
</tr>
<tr>
<td></td>
<td>• Conversation members</td>
</tr>
<tr>
<td></td>
<td>For group and record conversations, the conversation details page also provides capabilities to add or remove conversation members and to leave the conversation.</td>
</tr>
<tr>
<td>Avatar</td>
<td>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.</td>
</tr>
<tr>
<td>Attachment button</td>
<td>Enables you to include attachments in a message, including photos and documents.</td>
</tr>
<tr>
<td>Message field</td>
<td>Enables you to enter and send messages.</td>
</tr>
</tbody>
</table>

**Note:** For record conversations, all messages are sent as comments. Sending a message as a worknote is not supported.

Mobile device location on the ServiceNow Classic app

Administrators can set up the ServiceNow Classic mobile app UI to request a current location for a mobile device and store GPS coordinates in a string field.

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 ServiceNow Classic mobile app UI to display a special icon to request the current location of the mobile device.
Elements of the mobile UI: current location field

GPS coordinates are stored in the string field. Administrators can assign any label they want to this field.

**Get current location** allows the app to access the current location of your mobile device.

Current location field asks for permissions to use your current location the first time you use that option.

Change the app access to location from your device Settings.

Mobile barcode scanning on the ServiceNow Classic app

Administrators can set up the mobile app UI to request access to a mobile device camera to scan and store barcodes in a string field.
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 camera.

**Elements of the mobile UI: barcode scanning field**

- **String field** displays and stores the numeric value for the barcode. Administrators can assign any label they want to this field.
- **Scan barcode** opens the mobile device camera to scan a barcode.
- When you open the barcode scanner for the first time, the app asks your mobile device for permissions to access the camera.
- The native apps for iOS and Android support scanning the following barcode types:
  - 2D barcodes: QR Code, Data Matrix, PDF-417, AZTEC
Index

A
action function 59
action item 59
add app for Android BlackBerry UEM 195
add app for Android to Microsoft Intune 202
add app for iOS Intune 200
add app for iOS to BlackBerry UEM 193
AirWatch EMM 188
AirWatch MDM 188
AirWatch mobile device management 188
app config
  iOS 201
applet 59

B
BlackBerry MDM 193
BlackBerry mobile device management 193
BlackBerry UEM MDM 193
BlackBerry Unified Endpoint Management mobile device management 193

C
components and architecture 20, 210
configure a mobile theme 237
configure mobile classic location and barcode scanning 250
configure mobile push notifications 135
customize mobile homepages 235

e
Enterprise Mobility Management (EMM) 188

e
functions 59

F
functions 59

I
IBM Maas360 EMM 198
IBM Maas360 enterprise mobility management 198
IBM Maas360 iOS distribution 199
IBM Maas360 MDM 198
IBM Maas360 mobile device management 198
Intune mobile device management 200

M
mobile
  action 118
  action item parameters 104
  action tutorial 118
  button actions 177
button instance 182
calendar screen 81
calendar screen item view 84
configure employee list applet 86
data item 155
data item parameters 70
home page collections 235
list screen 72
list search 274
list title 221
map screen 75
navigation functions 96
Offline Mode 140
parameterized action functions 107
parameterized data item 156
parameterized navigation function 125
related list 89
roles and permissions 183
screen fields 175
screen parameters 175
Studio 62
URL screen 77
write-back action 178

© 2019 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.
<table>
<thead>
<tr>
<th>Term</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Citrix XenMobile</td>
<td>197</td>
</tr>
<tr>
<td>Jamf Pro</td>
<td>204</td>
</tr>
<tr>
<td>mobile device management (MDM)</td>
<td>188</td>
</tr>
<tr>
<td>mobile device management Intune</td>
<td>200</td>
</tr>
<tr>
<td>mobile device options</td>
<td>5</td>
</tr>
<tr>
<td>mobile experience</td>
<td></td>
</tr>
<tr>
<td>setup for administrators</td>
<td>56, 215</td>
</tr>
<tr>
<td>mobile force touch</td>
<td>248</td>
</tr>
<tr>
<td>mobile functions</td>
<td>109</td>
</tr>
<tr>
<td>mobile hide filters</td>
<td>234</td>
</tr>
<tr>
<td>mobile home page fields</td>
<td>235</td>
</tr>
<tr>
<td>mobile home page modules</td>
<td>237</td>
</tr>
<tr>
<td>mobile home page sections</td>
<td>236</td>
</tr>
<tr>
<td>mobile interface</td>
<td></td>
</tr>
<tr>
<td>access</td>
<td>254</td>
</tr>
<tr>
<td>client scripts</td>
<td>242</td>
</tr>
<tr>
<td>disable</td>
<td>251</td>
</tr>
<tr>
<td>home page</td>
<td></td>
</tr>
<tr>
<td>favorite</td>
<td>285</td>
</tr>
<tr>
<td>navigating applications</td>
<td>259</td>
</tr>
<tr>
<td>mobile list</td>
<td></td>
</tr>
<tr>
<td>display title</td>
<td>225</td>
</tr>
<tr>
<td>mobile list view</td>
<td>220</td>
</tr>
<tr>
<td>mobile lists</td>
<td>226</td>
</tr>
<tr>
<td>mobile localization</td>
<td>183</td>
</tr>
<tr>
<td>mobile log in</td>
<td>252</td>
</tr>
<tr>
<td>mobile multifactor authentication</td>
<td>216</td>
</tr>
<tr>
<td>mobile navigation</td>
<td></td>
</tr>
<tr>
<td>parameters</td>
<td>121</td>
</tr>
<tr>
<td>mobile parameterizations</td>
<td>116</td>
</tr>
<tr>
<td>mobile parameterized action</td>
<td>118</td>
</tr>
<tr>
<td>mobile parameters</td>
<td>59</td>
</tr>
<tr>
<td>mobile platform</td>
<td>154</td>
</tr>
<tr>
<td>mobile push notifications</td>
<td>133</td>
</tr>
<tr>
<td>mobile related lists</td>
<td>87</td>
</tr>
<tr>
<td>mobile screen type</td>
<td>161</td>
</tr>
<tr>
<td>mobile screen type fields</td>
<td>161</td>
</tr>
<tr>
<td>mobile security</td>
<td></td>
</tr>
<tr>
<td>architecture</td>
<td>20, 210</td>
</tr>
<tr>
<td>components</td>
<td>20, 210</td>
</tr>
<tr>
<td>components and architecture</td>
<td>20, 210</td>
</tr>
<tr>
<td>controls and practices</td>
<td>214</td>
</tr>
<tr>
<td>data flow retrieval</td>
<td>211</td>
</tr>
<tr>
<td>data security</td>
<td>213</td>
</tr>
<tr>
<td>identify and access management</td>
<td>21, 210</td>
</tr>
<tr>
<td>information storage</td>
<td>23, 211</td>
</tr>
<tr>
<td>internal mobile app distribution</td>
<td>212</td>
</tr>
<tr>
<td>security controls</td>
<td>26, 214</td>
</tr>
<tr>
<td>security patching</td>
<td>215</td>
</tr>
<tr>
<td>session length</td>
<td>211</td>
</tr>
<tr>
<td>session length and timeout</td>
<td>211</td>
</tr>
<tr>
<td>shared data</td>
<td>215</td>
</tr>
<tr>
<td>timeout</td>
<td>211</td>
</tr>
<tr>
<td>user authentication</td>
<td>21, 210</td>
</tr>
<tr>
<td>user data collection</td>
<td>215</td>
</tr>
<tr>
<td>user termination</td>
<td>23, 211</td>
</tr>
<tr>
<td>mobile SSO</td>
<td>217</td>
</tr>
<tr>
<td>mobile Studio</td>
<td></td>
</tr>
<tr>
<td>action functions</td>
<td>105</td>
</tr>
<tr>
<td>mobile supported devices</td>
<td>186</td>
</tr>
<tr>
<td>mobile theme configuration</td>
<td>237</td>
</tr>
<tr>
<td>mobile tutorial</td>
<td></td>
</tr>
<tr>
<td>applet parameter configuration</td>
<td>116</td>
</tr>
<tr>
<td>data item with parameters</td>
<td>116</td>
</tr>
<tr>
<td>parameterized navigation</td>
<td>121</td>
</tr>
<tr>
<td>mobile UI</td>
<td></td>
</tr>
<tr>
<td>activity streams</td>
<td>265, 278</td>
</tr>
<tr>
<td>application navigator</td>
<td>259</td>
</tr>
<tr>
<td>barcode scanning</td>
<td>294</td>
</tr>
<tr>
<td>condition sets</td>
<td>270</td>
</tr>
<tr>
<td>Connect Chat</td>
<td></td>
</tr>
<tr>
<td>conversations</td>
<td>291</td>
</tr>
<tr>
<td>device location</td>
<td>293</td>
</tr>
<tr>
<td>favorites</td>
<td>282</td>
</tr>
<tr>
<td>filters</td>
<td>269, 280</td>
</tr>
<tr>
<td>get started for end users</td>
<td>251</td>
</tr>
<tr>
<td>home page</td>
<td></td>
</tr>
<tr>
<td>favorite</td>
<td>284</td>
</tr>
<tr>
<td>list</td>
<td>271</td>
</tr>
<tr>
<td>list configuration</td>
<td>231</td>
</tr>
<tr>
<td>lists</td>
<td>263, 276</td>
</tr>
<tr>
<td>mobile app</td>
<td>288, 289</td>
</tr>
<tr>
<td>visualizations</td>
<td>285, 288</td>
</tr>
<tr>
<td>mobile UI styles</td>
<td>131, 131</td>
</tr>
<tr>
<td>MobileIron MDM</td>
<td>205</td>
</tr>
<tr>
<td>MobileIron mobile device management</td>
<td>205</td>
</tr>
<tr>
<td>N</td>
<td></td>
</tr>
<tr>
<td>navigation function</td>
<td>59</td>
</tr>
<tr>
<td>notification settings</td>
<td>289</td>
</tr>
<tr>
<td>O</td>
<td></td>
</tr>
<tr>
<td>offline mode</td>
<td></td>
</tr>
<tr>
<td>mobile</td>
<td>140</td>
</tr>
<tr>
<td>Offline Mode</td>
<td></td>
</tr>
<tr>
<td>mobile</td>
<td>140</td>
</tr>
<tr>
<td>S</td>
<td></td>
</tr>
<tr>
<td>set up</td>
<td></td>
</tr>
<tr>
<td>mobile experience admin</td>
<td>56, 215</td>
</tr>
<tr>
<td>smart button function</td>
<td>59</td>
</tr>
<tr>
<td>smartphone interface</td>
<td></td>
</tr>
<tr>
<td>blackberry</td>
<td>259</td>
</tr>
<tr>
<td>T</td>
<td></td>
</tr>
<tr>
<td>tablet app</td>
<td></td>
</tr>
<tr>
<td>application navigator</td>
<td>261</td>
</tr>
<tr>
<td>tablet support</td>
<td>12</td>
</tr>
</tbody>
</table>

© 2019 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.