# Table of Contents

**ServiceNow Mobile** .......................................................................................................................... 1

- Now Mobile app .......................................................................................................................... 4
- Mobile Agent app ..................................................................................................................... 5
  - Mobile Agent demo .................................................................................................................. 6
- Onboarding Mobile app ............................................................................................................. 11

**Considerations before implementation** .................................................................................... 12

- Base system functionality for ServiceNow mobile ........................................................................ 12
- Mobile Publishing .................................................................................................................... 15
- Mobile themes .......................................................................................................................... 35
- Mobile security ......................................................................................................................... 61
- Edge Encryption for ServiceNow mobile .................................................................................. 77
- Mobile GovCommunityCloud (GCC) Compliance .................................................................. 78
- Enterprise mobility management (EMM) ................................................................................ 80

**Tested devices and supported versions for ServiceNow mobile apps** ..................................... 113

**Using the mobile apps** .............................................................................................................. 115

- Log in to an instance with a mobile app ..................................................................................... 115
- Mobile app structure ................................................................................................................ 118
- Mobile search ............................................................................................................................ 123
- Accessing and posting to the mobile activity stream ................................................................ 128
- Filtering and sorting results in mobile ...................................................................................... 130
- Mobile saved views .................................................................................................................. 134
- Mobile platform dashboards ..................................................................................................... 137
- Handling mobile alerts .............................................................................................................. 148
- Mobile app feedback ................................................................................................................ 150
- Accessibility features in mobile ............................................................................................... 152
- Mobile app settings .................................................................................................................. 156
- Accessing an instance on a mobile device web browser ......................................................... 168

**Mobile plugins and upgrades** .................................................................................................. 170

- Mobile plugins ......................................................................................................................... 171
- Migrate from the ServiceNow Classic mobile app to the ServiceNow Mobile Platform ........ 176
- Mobile migration from Madrid to New York and later releases ............................................ 178
- Migration from New York and later releases ......................................................................... 187

**Building and configuring mobile apps** .................................................................................... 187

- ServiceNow mobile terminology changes .............................................................................. 188
- Mobile tutorials ....................................................................................................................... 191
- Design considerations for mobile apps .................................................................................... 219
- Mobile hierarchy ....................................................................................................................... 219
- Mobile impersonation .............................................................................................................. 224
- Mobile App Builder ................................................................................................................ 229

© 2022 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>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile Card Builder</td>
<td>244</td>
</tr>
<tr>
<td>Mobile Studio</td>
<td>265</td>
</tr>
<tr>
<td>Create a mobile application using Guided Application Creator</td>
<td>269</td>
</tr>
<tr>
<td>Empty state display</td>
<td>273</td>
</tr>
<tr>
<td>Navigation bar</td>
<td>278</td>
</tr>
<tr>
<td>Launcher screens</td>
<td>284</td>
</tr>
<tr>
<td>Data items</td>
<td>319</td>
</tr>
<tr>
<td>Mobile dashboard preview</td>
<td>335</td>
</tr>
<tr>
<td>Mobile screens</td>
<td>340</td>
</tr>
<tr>
<td>Enabling bookmarking screens and web pages</td>
<td>484</td>
</tr>
<tr>
<td>Mobile styles</td>
<td>486</td>
</tr>
<tr>
<td>Mobile functions</td>
<td>511</td>
</tr>
<tr>
<td>Mobile properties</td>
<td>556</td>
</tr>
<tr>
<td>Mobile push notifications</td>
<td>559</td>
</tr>
<tr>
<td>Mobile URLs</td>
<td>581</td>
</tr>
<tr>
<td>Roles and permissions for ServiceNow Mobile Apps</td>
<td>590</td>
</tr>
<tr>
<td>Siri shortcuts</td>
<td>590</td>
</tr>
<tr>
<td>Mobile surveys</td>
<td>591</td>
</tr>
<tr>
<td>Displaying campaigns on ServiceNow mobile</td>
<td>591</td>
</tr>
<tr>
<td>Mobile external users</td>
<td>600</td>
</tr>
<tr>
<td>Mobile system property configurations</td>
<td>600</td>
</tr>
<tr>
<td>Mobile authentication</td>
<td>610</td>
</tr>
<tr>
<td>User Experience Analytics for mobile applications</td>
<td>612</td>
</tr>
<tr>
<td>Mobile location tracking configuration</td>
<td>613</td>
</tr>
<tr>
<td>Offline mode</td>
<td>614</td>
</tr>
<tr>
<td>Request Offline Mode</td>
<td>616</td>
</tr>
<tr>
<td>Configure Offline Mode behavior</td>
<td>617</td>
</tr>
<tr>
<td>Configure Offline Mode for applications or applets</td>
<td>619</td>
</tr>
<tr>
<td>Define the number of displayed records in Offline Mode</td>
<td>620</td>
</tr>
<tr>
<td>Configure data items in Offline Mode</td>
<td>620</td>
</tr>
<tr>
<td>Configure Offline Mode properties for action functions</td>
<td>621</td>
</tr>
<tr>
<td>Configure Offline Mode properties for function instances</td>
<td>623</td>
</tr>
<tr>
<td>Scheduled offline caching</td>
<td>623</td>
</tr>
<tr>
<td>Mobile bar-code scanning</td>
<td>624</td>
</tr>
<tr>
<td>Multi-scan for mobile applications</td>
<td>626</td>
</tr>
<tr>
<td>Configure a grouped input for multiple scans</td>
<td>629</td>
</tr>
<tr>
<td>Localization on mobile devices</td>
<td>631</td>
</tr>
<tr>
<td>Create translation records for mobile</td>
<td>634</td>
</tr>
<tr>
<td>Translation values for mobile element</td>
<td>635</td>
</tr>
<tr>
<td>Domain separation and mobile</td>
<td>636</td>
</tr>
<tr>
<td>ServiceNow Classic mobile app</td>
<td>637</td>
</tr>
<tr>
<td>Device security for ServiceNow Classic</td>
<td>639</td>
</tr>
</tbody>
</table>
Mobile configuration for ServiceNow Classic .......................................................... 643
Get started with the ServiceNow Classic app ......................................................... 671
Support for ServiceNow Classic .............................................................................. 701
Disable the ServiceNow Classic mobile app ............................................................ 703

Index ........................................................................................................................................... a
**ServiceNow Mobile**

Access your instance from anywhere using the ServiceNow® Now Mobile, Mobile Agent, and Mobile Onboarding apps on your mobile device.

**Take your work wherever you go**

Your instance is available whether you’re away from your desk or at the office. Manage incidents, collaborate with your teams, respond to approval requests, access the knowledge base, or receive push notifications with your mobile device.

ℹ️ **Note:** The Now Mobile platform is only available for Madrid and later releases. For London and earlier releases, see Mobile configuration for ServiceNow Classic.

<table>
<thead>
<tr>
<th></th>
<th>Download the mobile apps</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Mobile App Icon]</td>
<td>The Now Platform provides three mobile apps. Each app is designed for a specific set of users. Download these apps for your Android or iOS devices using the Google Play store or Apple app stores.</td>
</tr>
</tbody>
</table>

ℹ️ **Note:** The Apple App Store and the Google Play Store are the only two official distribution channels that are supported for these mobile applications.

<table>
<thead>
<tr>
<th></th>
<th>Get started quickly using available base system applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Launch Icon]</td>
<td>Take advantage of pre-configured mobile applications to get your users started with common functionality like approvals for Field Service Management.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Use the features on your mobile device with your ServiceNow instance.</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Location Icon]</td>
<td>ServiceNow apps can use your mobile device’s features to make calls, send SMS messages, and display navigation information. For example, you can use the camera on your mobile device to scan bar codes, or create image attachments.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Use Studio to create custom applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Studio Icon]</td>
<td>Build applications that you can customize for your organization’s specific needs. With Studio, you can quickly create applications in a simple, low-code interface.</td>
</tr>
</tbody>
</table>

**Download the mobile apps**

ℹ️ **Note:**

The Mobile Onboarding app is being deprecated!

With the Now Platform Rome release in September 2021, we started phasing out support for the Mobile Onboarding. Customers may no longer activate it, and we are not offering enhancements or non-critical bug fixes. Mobile onboarding features are available in the Now Mobile app for HR Service Delivery."
Your users access ServiceNow mobile from a client such as Mobile Agent, Now Mobile, or Mobile Onboarding. These applications are available in the Google Play or Apple App Store. Each app works with a specific area of the ServiceNow platform.

**Tailored experiences**

Take advantage of pre-configured applications. These applications can get you started right away with the ServiceNow mobile platform.

ℹ️ **Note:**

The Mobile Onboarding app is being deprecated!

With the Now Platform Rome release in September 2021, we started phasing out support for the Mobile Onboarding. Customers may no longer activate it, and we are not offering enhancements or non-critical bug fixes. Mobile onboarding features are available in the Now Mobile app for HR Service Delivery."

**Now Mobile**

Now Mobile is tailored for your employees. Employees can find answers, create requests, and get help across all departments.

**Mobile Agent**

Mobile Agent is tailored for your service agents. Agents can get work done anywhere through an easy to navigate native mobile interface.

**Mobile Onboarding**

Mobile Onboarding is tailored for your new hires. New employees can get informed and become productive in a matter of minutes.

**Now Support**

Use Now Support to conduct key customer support tasks from a modern mobile app that is powered by the ServiceNow mobile platform.

ℹ️ **Note:** The Now Support is designed to interact with customer support, and is not a customizable mobile app.
Use the features on your mobile device with your ServiceNow instance.

Use built-in features on your mobile device to perform certain actions in your mobile apps. Use your device's camera to upload photos, perform image-based searches, or scan barcodes to catalog your assets. The Now Mobile app uses your mobile device's photo, phone, keyboard, messaging, and email applications.
Use Studio to create custom applications

Mobile Studio is part of the existing Studio environment. With mobile Studio, you can create, modify, and manage applets for use in ServiceNow mobile apps using an intuitive, low code interface.

Get started

- To learn more about the elements of ServiceNow mobile, see Mobile app layout.
- For more information on how to begin setup with ServiceNow mobile, see ServiceNow mobile app configuration.
- To learn about the migration process from classic mobile to ServiceNow mobile, see Migrate from classic mobile to ServiceNow mobile.
- Learn more about how to create your own applications on the ServiceNow® Developer Site. The Developer Site contains courses and examples to get you started on creating custom mobile applications suited to your organization’s specific needs. To access the mobile content on the Developer Site, see Mobile application training.

Applications and features

- Now Mobile app
- Agent mobile app
- Mobile Onboarding app
- Using the mobile apps
- Considerations before implementation
- Building and configuring mobile apps

Now Mobile app

Enable your users to submit incidents and requests, manage tasks, and access company resources from anywhere using the Now Mobile app.
For the full range of Now Mobile documentation, see Now Mobile app in the Employee Service Management business unit.

**Features of the Now Mobile app**

The Now Mobile app is a downloadable app designed to help your employees with the following tasks:

- Get help by submitting requests and issues.
- Find answers using global search and knowledge base.
- View and report issues around your assets
- View and complete tasks
- Upload images and attachments to ServiceNow records
- Request help from Agent Chat and Virtual Agent
- Use pre-configured Siri shortcuts unique to the Now Mobile app to complete common actions.

**Note:** For more information, see Base system functionality for ServiceNow mobile.

**Download the app**

Download the Now Mobile app for Apple iOS or Google Android from the Apple App Store or the Google Play store.

**Access the demo**

Take a look at the Now Mobile app demo. The demo is a curated experience designed to show how your employees can accomplish their work from anywhere. For more information on the app, see Now Mobile app demo.

**Mobile Agent app**

Use the ServiceNow Mobile Agent app to update records, coordinate with coworkers, track your location, and work without an internet connection — all from your mobile device.
Download Mobile Agent app for Apple iOS or Google Android from the Apple App Store or the Google Play store.

Demonstrates how employees can use ServiceNow Mobile Agent appsto perform work tasks remotely.

Features of the Mobile Agent app

- Give your agents a mobile interface to triage, address, and resolve requests.

- Use the Mobile Agent app’s offline mode to work in areas without an internet connection. For details see Offline mode.

- Use the Mobile Agent app’s location tracking feature to keep a record of where an agent last worked on an instance. For more detail, see Mobile location tracking configuration.

Note: For more information, see Base system functionality for ServiceNow mobile.

Download the app

Download the ServiceNow Mobile Agent for Apple iOS or Google Android from the Apple App Store or the Google Play store.

Access the demo

Take a look at the Mobile Agent demo. The demo is a curated experience designed to show how your agents can complete tasks wherever they are. For more information on the app, see Mobile Agent demo.

Mobile Agent demo

Access the demo in your Mobile Agent to learn how the app can help your agents work from anywhere using their mobile device.
The Mobile Agent demo includes curated demonstrations to show how your agents can perform tasks on their mobile device. Agents can track incoming work, access important knowledge, and collaborate with teams on the go.

Access the demo

In the Mobile Agent, tap the **TRY WITH A DEMO ACCOUNT** button at the bottom of the login screen. Then, select the role that matches the demo you’d like to see:

- **Customer Service Agent** - resolve customer issues and swiftly respond to customer’s requests
- **Field Service Agent** – manage their schedules, access knowledge and parts inventories, and track their work for the day
- **Project Manager** - initial, design, execute, monitor, and control projects
- **IT Asset Manager** - manage IT assets and knowledge of assets whereabouts for financial and compliance purposes
- **IT Operator** – triage and assign alerts, research root cause, and collaborate to act and resolve issues quickly
- **Security Analyst** - handle security incidents and protect security operation environments
- **Service Desk Agent** – prioritize their tasks for the day, manage their shifts, and swiftly respond to major incidents

Finally, tap the **Launch Demo** button to start using the demo. The demo automatically logs you in to a ServiceNow instance with an example account.
Demo highlights

The ServiceNow Agent mobile demo displays an applet launcher page divided into sections. Tap an item in any section to open the item or tap See All to view all items in that section. For details on how applet launchers, applets, actions work together in the mobile environment, see Mobile hierarchy.

Customer Service Agent

Manage priority cases

Agents use **Cases that need attention** to quickly identify priority cases needing their response. These are cases where customers have responded back, or where internal blocking tasks have been completed. Agents can open cases to view the latest updates, add comments or work notes, edit case details, and propose solutions. Related lists show SLAs, tasks, child cases, and other case-related lists.

Initiate case workflows

From the **New cases** list, agents can swipe to accept or reject a case. They have quick access to active critical cases. From this case list, agents view case
details and can initiate standard workflows, such as proposing a major case or escalating a case.

**Approve escalations and change requests**

If users such as managers and supervisors have approval roles, they see their escalation approvals and change request approvals as well. They can open the approvals, view details, and either approve or reject the requests. Managers can also easily reassign cases to other teams or individual users.

**Field Service Agent**

**Optimize my daily schedule**

My Task Map displays locations for each of an agent’s assigned tasks. Use the **Optimize Today’s Route** option to reorder tasks and show the quickest way to get to work locations.

**Access knowledge articles and parts inventories**

Agents view the parts required for the tasks they’re currently assigned, and also view a full inventory of on-hand and defective stock. Agents can locate needed parts and see a map of warehouses and technicians that carry the parts they need. The Mobile Agent includes knowledge management capabilities to provide immediate access to the knowledge that’s relevant to the task at hand. These capabilities include FAQs, error codes, schematics, and videos.

**Manage my work order tasks**

Once tasks are prioritized, your agents select their first work order and accept it. Thus acceptance provides visibility to the dispatcher and updating the agent’s location. With a swipe, they can track the parts needed to fix a problem or scan a barcode to consume an asset from inventory. As part of completing certain tasks, your agents can document conformance to a safety checklist, document equipment readings, or fill out an inspection questionnaire. These actions are done using forms specific to the type of work or equipment or even customer. Then agents can close their tasks, capturing a customer signature and adding any important work notes.

For more information on Field Service Management using mobile applications, see [Mobile experience for Field Service Management](#).

**Project Manager**

**Monitor all your projects on the go**

Quickly view key project status and details, with a tap or a swipe, all from this single, native, mobile app. Filter to view color-coded, at-a-glance status of projects by project manager, portfolio, or business unit. See Critical Projects and Projects in Red on the opening page of the app - and tap in for more actionable insights.

**Raise awareness and take action**

Gain mobile visibility into the number of open risks, issues decisions, actions, and changes (RIDAC) that may affect the overall project health. Enter new RIDAC items right from your mobile device throughout the project life cycle.

**Collaborate with project stakeholders**

Provide timely project updates anytime, anywhere. The native mobile app enables you to send emails, add work notes, or upload project documents. Team members and project stakeholders always know exactly where things stand with their projects.
IT Asset Manager

**Conduct an inventory audit directly from your phone**

Conduct scheduled or ad-hoc asset inventory audits of stockrooms, data centers or office locations by scanning barcodes with the Mobile Agent. Multiple technicians can scan against a single audit at the same time.

**Receive hardware assets with mobile barcode scanning**

Accurately scan and receive multiple assets against open purchase orders. Create new asset records directly from the Mobile Agent.

**Easily manage hardware assets throughout the disposal process**

Verify hardware assets that are ready for disposal or confirm which hardware assets will be disposed. Quickly scan hardware assets with your phone from any warehouse, datacenter, or stockroom.

**Improve IT asset management accuracy**

Accurately scan and receive assets for a streamlined operation with automatic updates to your asset data. Scan multiple assets in a single batch upload. Quickly resolve discrepancies between ServiceNow information and audits. Instantly lookup info about any hardware asset. For more information on Hardware Asset Management using mobile applications, see [Mobile app for Hardware Asset Management](#).

IT Operator

**Triage and assign alerts in a services context**

Use the Services section to prioritize which services need attention to address operational alerts. View service details, then view the primary alerts grouped by machine learning. View individual alerts to see if any other services are impacted by the alert, providing insight into the scope of the issue.

**Investigate root cause of alerts**

Drill into the alert details, review the secondary alerts that are automatically grouped with the alert. These alerts provide key insights into the root cause of the problem. View related incidents, repeated alerts, and log files attached to a related incident. Access knowledge articles that provide more information about the problem, and suggestions actions to address it.

**Collaborate and take action**

With the root cause and recommended solutions, agents can add work notes to the alert and even initiate actions. Actions can include like kicking off a workflow or proposing a major incident.

For more information on Event Management using mobile applications, see [Mobile experience for Event Management](#).

Security Analyst

**Manage security incidents anywhere**

Quickly view security incidents and review details, including the incident priority, affected user, configuration item, attacker information, and more. Analysts can search by keyword, and filter by security incidents, users, and vulnerability groups, to find exactly what they are looking for.

**Take action with the tap of a button**

Open IT requests, initiate orchestration tasks like firewall blocks and host isolation, and communicate with affected users, all from within the security incident on your mobile device.
Review active vulnerabilities on the go
Analysts can review vulnerability groups assigned to them to determine the next steps; view remediation status, assignment group, individual vulnerable items, and more. Then, create a Change Request and resolve the vulnerability or opt to defer until a later date.

Service Desk Agent

Manage my work
View new incidents coming in, incidents currently at risk, and even breached incidents on the My Work tab. Agents can easily add comments, reassign, resolve, and approve incidents and tasks. Agents can also view and manage shifts to ensure adequate support coverage. The app includes a visual calendar view that shows your schedule as well as the shifts they cover for other agents. Request shift coverage or even request time off right through the app.

Manage the team’s work
Agents use the My Team dashboard to direct resources no matter where they are. Review unassigned work and take ownership of a critical incident. You can also assign it to a member of their team based on availability for the fastest incident response. Manage the team’s shift coverage using the app to ensure that all-time slots are adequately covered. Agents can use their phone’s native voice-to-text capabilities to provide the input to the Virtual Agent to see all schedules where they may have an upcoming coverage gap.

Engage in major incidents as if you’re in the office
Agents use the Mobile Agent to engage and respond to major incidents. They can see proposed major incident candidates and quickly review and promote them. Collaborate by kicking off a conference call right from the app to serve as an in-person war room for the team.

For more information on incident management using mobile applications, see Mobile experience with ITSM Mobile agent

Learn more about the Mobile Agent app
Use the following links to learn more about how to use and configure the Mobile Agent app.

• Mobile Agent app
• Building and configuring mobile apps

Onboarding Mobile app
Enable your new hires to complete onboarding to-dos, view relevant media sections, view relevant banners, chat with an agent, and more using the Mobile Onboarding app.

Demonstrates how employees can use ServiceNow mobile apps to perform work tasks remotely.

Note:
The Mobile Onboarding app is being deprecated!
With the Now Platform Rome release in September 2021, we started phasing out support for the Mobile Onboarding. Customers may no longer activate it, and we are not offering enhancements or non-critical bug fixes. Mobile onboarding features are available in the Now Mobile app for HR Service Delivery."
Features of the Mobile Onboarding app

The Mobile Onboarding app works with the HR Service Delivery application to prepare your new hires for work. Use the Mobile Onboarding app to:

- Engage new employees before they start.
- Guide to-dos across all departments, including IT and HR.
- Provide your new hires with relevant media, chat options, and more.

Note: For more information, see Base system functionality for ServiceNow mobile.

Download the app

Download the Mobile Onboarding app for Apple iOS or Google Android from the Apple App Store or the Google Play store.

Access the demo

Take a look at Mobile Onboarding in the app demo. The demo is a curated experience designed to show how your new hires can prepare for work. For more information, see Now Mobile app demo.

Note: To use the Mobile Onboarding app, you must have the HR Service Delivery application installed on your instance. For more detail on this product, see HR Service Delivery.

Considerations before implementation

Use these guidelines and considerations when implementing ServiceNow® mobile on your instance.

Base system functionality for ServiceNow mobile

Learn about the base system functionality included in with ServiceNow® mobile apps.
Base system functionality for Mobile Agent app

Take advantage of the Mobile Agent app by activating the base system applications available from ServiceNow. For a list of plugins available for Mobile Agent app, see Mobile plugins for Mobile Agent app.

To use the Mobile Agent app, you must install additional plugins, which provide capability such as customer service, asset receiving, or IT service management.

Base system functionality for Now Mobile

Employees can use the Now Mobile app to do these tasks:

**Search**
Find answers using global search and knowledge base.

**Popular services**
View and request items from the service catalog.

**Popular articles**
Search and view company resources and useful articles.

**For Me tab**
Log in to the app to view your home page. Tap different items to see more information.

Use **My To-Dos** to see if you have anything on your to-do list. For example, approve an item or finish an onboarding task.

Use **My Requests** to track the status of your requests and update request details. If one of your requests is taking longer than you expect, you can ask for an update here.

**Services tab**
View things you’ve ordered and things you own.

Use **My Locker** to view the company items that you own, such as your laptop or phone.
Go here to report an issue if one of your company items isn’t working properly.

**Information tab**
Search for or browse company articles. Go here to look for a
solution to a problem or to learn company information.

**Notifications tab**
Enable or disable notifications here. Go here to view all of your notifications in one place.

**Siri shortcuts**
If you’re an iOS user, you can use Siri to open some pages in the app. These Siri shortcuts are available:

- Open a chat window.
- Browser items and services.
- Open your tasks.
- Open your requests.

Add functionality to the Now Mobile app by installing plugins. For a list of the available plugins, see [Mobile plugins for Now Mobile](#).

**Base system functionality for Mobile Onboarding**

⚠️ **Note:**

The Mobile Onboarding app is being deprecated!

With the Now Platform Rome release in September 2021, we started phasing out support for the Mobile Onboarding. Customers may no longer activate it, and we are not offering enhancements or non-critical bug fixes. Mobile onboarding features are available in the [Now Mobile app for HR Service Delivery](#).
New hires can use the Mobile Onboarding app to:

**View HR Requests**
Employees can view their HR requests by navigating to the **For Me** tab and tapping **See All** from the My Requests section. Opening an HR request takes them to the HR ticket page, where they can learn more about the details of their case.

**Request Help**
Employees can submit HR requests by navigating to the **Services** tab and searching or browsing for HR services. They can also tap the plus icon, and then tap **Create HR case** to create a general inquiry HR case.

**Complete HR tasks**
Employees can view and complete their HR tasks, such as signing electronic documents, uploading attachments, and watching videos.

**View relevant banners**
Employees can view relevant banners. See **Configure mobile content** for further information.

**Chat with a virtual agent**
If enabled, employees can chat with a virtual agent for answers to common questions.

For a list of plugins available for the Mobile Onboarding app, see **Mobile plugins for Mobile Onboarding**.

**Mobile Publishing**
The ServiceNow® Mobile Publishing application enables you to publish secure and branded mobile applications. These mobile apps use your unique company identity and management method.

The following video provides an overview of how you can brand your apps with Mobile Publishing.

Video describing the mobile publishing process.

Use Mobile Publishing to create a visually distinct version of ServiceNow mobile apps. Tailor your apps to your unique company identity. Configure each of your mobile apps with a unique name, splash screen, icon, Mobile Device Management (MDM) vendor, and theme colors.
The ServiceNow publishing program complies with the suggested practices from Google and Apple for branded app releases. The program gives you a guided path to achieve your branding goals.

You can order a branded version of any ServiceNow mobile app through your production instance. With branding, you can make the following customizations to your mobile apps:

- App icon
- App name
- App splash screen
- In app theme colors
- Add MAM (Mobile App Management) support

For more information on this process, see Request a branded mobile app.

Mobile themes

Use mobile themes to change the color scheme of your mobile apps. Themes can further enhance the appearance of your apps by matching your color scheme with the splash screen and icons. Splash screens are the first screen of the app that displays when the app is launched. For more detail on mobile themes, see Mobile themes.

How to request branded apps

To request a branded app, you must have the Mobile Publishing plugin activated on your ServiceNow instance. Mobile Publishing is a paid plugin that ServiceNow personnel must activate on your production instance. For more information on requesting mobile branded apps, see Request a branded mobile app.

Mobile app distribution

After you use the Mobile Publishing plugin to request a branded mobile app, the branded mobile app can be distributed privately or publicly. For information on your distribution options, see Distributing your mobile app.

Prerequisites for Mobile Publishing

Before submitting your first branded app request with Mobile Publishing, it is important to set up some prerequisite tools.

ServiceNow instance requirements

To use Mobile Publishing on any instance, an admin must have the branding plugin (com.glide.sn-mobile-whitelabel) installed on your production instance. The Mobile Publishing plugin is a paid plugin that can be installed from the ServiceNow® Store once your license is activated. For more details on the ServiceNow Store, see Install a ServiceNow Store application.

Note that Apple deprecated their legacy binary protocol, which affects customers who use custom push notifications. Custom push notifications for customers with mobile branding stopped working as of November 2020. This deprecation only affects customers using New...
York and Orlando instances who have apps customized with Mobile Publishing. To prevent the push issue, you can do one of the following:

- Orlando customers using Mobile Publishing customized apps can upgrade to Orlando Patch 5.

Paris customers are not affected.

For more information, see the Mobile Publishing - HTTP/2 Push Notifications support [KB0830082] article in the Now Support Knowledge Base.

**Apple tools required**

If you want to create your app for the iOS platform, the following tools are required from Apple.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple Business Manager (ABM)</td>
<td>Required to distribute iOS branded applications privately by using MDM providers. For more information, navigate to the Apple Business Manager User Guide.</td>
</tr>
<tr>
<td></td>
<td><strong>Important:</strong> Be sure to enable the “Custom Apps” setting in your ServiceNow instance. This enables you to publish the iOS branded apps into ABM.</td>
</tr>
<tr>
<td>Apple Developer Account</td>
<td>Required to distribute iOS branded applications publicly. For more information, navigate to the Apple Developer documentation.</td>
</tr>
<tr>
<td>XCode</td>
<td>Required to compile the iOS branded applications for public distribution. For more information, navigate to the Apple Developer documentation.</td>
</tr>
</tbody>
</table>

**Android tools required**

If you want to create your app for the Android platform, the following tools are required.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Firebase account</td>
<td>Required to set up Push Notifications for Android branded applications. For more information, navigate to the Google Firebase documentation.</td>
</tr>
</tbody>
</table>

**Distributing your mobile app**

Learn about making your mobile apps available for download to Android and iOS devices. ServiceNow® Mobile Publishing enables you to distribute apps to users through different channels according to Apple and Google guidelines. You can distribute apps privately on
your own websites or on public app stores provided by your operating system vendors. The decision depends on your enterprise requirements.

**Privately distributed apps**

Private or enterprise distribution can be used to distribute Android and iOS branded applications. Private distribution includes all means of distributing apps outside of public app stores.

**iOS branded apps for private distribution**

The ServiceNow® branding program complies with the suggested practices from Apple for branded app releases. The program gives you a guided path to achieve your branding goals.

Depending on your distribution model for an iOS deployment, you must have signed up for Apple Business Manager to use the Apple VPP program or have a valid Apple Customer DEP ID, or both. This enables you to use your ServiceNow instance to publish an app to your Apple Business Manager on your behalf. Apple then allows you to distribute the app to your employees with your Mobile Device Management (MDM) vendor or by using Apple Business Manager store redemption codes.

When you use Mobile Publishing for iOS private distribution, you can then test the mobile app using Testflight. ServiceNow also resolves any rejections from Apple when the application is reviewed.

The following image explains the workflow for privately distributing an app. It might take 2-4 weeks to build the iOS branded application and to pass the Apple review process.

---

**Android branded apps for private distribution**

The ServiceNow branding program complies with the suggested practices from Google for branded app releases. Using Mobile Publishing, you can request a branded Android application and receive an APK file. This file can then be uploaded to your MDM portal or your Managed Google Play account. The file can also be hosted on a website so that Android users can download the application directly onto their devices.

The following image explains the workflow for privately distributing an app. It might take 1–2 weeks to build the Android branded application.

---

**Publicly distributed apps**

Public distribution can be used to distribute iOS or Androidbranded applications on public app stores.

**iOS branded apps for public distribution**

The ServiceNow® branding program complies with the suggested practices from Apple for branded app releases. Mobile Publishing enables you to request a branded iOS app that
can be distributed on the public Apple Store. After you complete the request form, your ServiceNow instance builds the app and provides a link from where you can download an XCode archive file (.xcarchive). Then you download the XCode archive file, update it, and re-sign it to generate an iOS application archive file (.ipa). After the .ipa file is generated, you upload it to Apple Connect for testing. When testing is completed, you approve the build and can deploy your branded app with the public Apple Store.

The following image summarizes the workflow. It might take 1–2 weeks to build the iOS app for public distribution.

### iOS branded app publishing process for public distribution

1. Submit an iOS mobile branded app request
2. ServiceNow builds app & provides XCode archive file download link
3. Customer downloads XCode archive and generates IPA file
4. Customer uploads IPA file to Apple Connect for testing
5. Customer approves build & deploys app with Apple Store

### Android branded apps for public distribution

The ServiceNow branding program also complies with the suggested practices from Google for branded app releases. Use Mobile Publishing to request a branded Android app that you can upload and distribute with the Google Play Store. After you complete the branded app request form in Mobile Publishing, your ServiceNow instance builds the app. Then it provides a link from where you can download an Android App Bundle file (.aab). Download the AAB file and test it. If the tests are successful, you approve the build and can deploy your app with the Google Play Store.

The following image summarizes the workflow. It might take from 1 to 2 weeks to build the Android app for public distribution.

### Android branded app publishing process for public distribution

1. Submit an Android mobile branded app request
2. ServiceNow builds app & provides AAB file download link
3. Customer downloads AAB file for testing
4. Customer approves the build and deploys app with Google Play Store

### Request a branded mobile app

Publish secure custom mobile apps with your unique company identity by requesting branded versions of ServiceNow® mobile apps.

### Request a new branded Android app for private distribution

Request and publish a custom Android app that has your unique company identity. Then make the app available for private distribution.

### Before you begin

Role required: admin

### About this task

Mobile Publishing enables you to request a branded Android app that you can distribute privately. After the request form is completed and the build is successful, you can download an Android Package file (.apk). You can then upload this APK file to your MDM portal or Managed Google Play account. The APK file can also be hosted on a website for Android users to download the application directly onto their devices. Ensure that images or text in icons you use are at least 550 pixels from the edge to account for the rounded corners in Android icons.
Procedure
1. Navigate to All > System Mobile > Request Branded App.
2. Select New Custom Branding.
3. On the form, fill in the fields.

**Branding Setup form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Distribution</strong></td>
<td></td>
</tr>
<tr>
<td>Distribution Type</td>
<td>Type of distribution. Select <strong>Private/Enterprise.</strong></td>
</tr>
<tr>
<td><strong>Platforms</strong></td>
<td></td>
</tr>
<tr>
<td>Deploy app on</td>
<td>Mobile operating system for your mobile application. Select <strong>Android.</strong></td>
</tr>
<tr>
<td></td>
<td>Selecting an option reveals the branding setup fields for that operating</td>
</tr>
<tr>
<td></td>
<td>system.</td>
</tr>
<tr>
<td><strong>Requester’s Contact Information</strong></td>
<td></td>
</tr>
<tr>
<td>First Name</td>
<td>First name for the requester of the branded application.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Last Name</td>
<td>Last name for the requester of the branded application.</td>
</tr>
<tr>
<td>Email</td>
<td>Email address that you use to receive notifications about the build status of your request.</td>
</tr>
<tr>
<td><strong>Brand Your App</strong></td>
<td></td>
</tr>
<tr>
<td>Long Name</td>
<td>Unique name for your app. The name cannot exceed 30 characters and must be unique. This name displays for your app when asking for device permissions such as location or push notifications.</td>
</tr>
<tr>
<td>Short Name</td>
<td>Name of the app icon. This name appears under the icon on your mobile device. The name cannot exceed 13 characters. This name displays below the app icon on mobile devices.</td>
</tr>
<tr>
<td>MAM Support</td>
<td>Type of MAM (Mobile App Management) vendor that you want to use with your apps. Choices are as follows:</td>
</tr>
<tr>
<td></td>
<td>• None: App does not use MAM.</td>
</tr>
<tr>
<td></td>
<td>• Blackberry: App uses Blackberry MAM.</td>
</tr>
<tr>
<td></td>
<td>• Microsoft Intune: App uses Intune MAM.</td>
</tr>
<tr>
<td>In-app Color Branding</td>
<td>Mobile color theme on your instance. For information on configuring a color theme for your app, see Mobile themes.</td>
</tr>
<tr>
<td>Native client</td>
<td>Native client to which the color theme is assigned.</td>
</tr>
<tr>
<td>Client theme</td>
<td>Color theme that is assigned to the selected native client. In-app color branding uses the configured client theme from the instance. See Mobile themes.</td>
</tr>
<tr>
<td>Universal Links Host</td>
<td>This is the same URL as your instance. Universal Links allow your users to follow links to content inside your app.</td>
</tr>
<tr>
<td>EULA/Privacy Policy</td>
<td>End User License Agreement (EULA) and privacy policy options. You can use the default ServiceNow EULA and privacy policy or you can use a custom version. The ServiceNow version is selected by default.</td>
</tr>
<tr>
<td>EULA URL</td>
<td>URL where app users can view your custom end user license agreement from the mobile app settings. This option only displays when you select Use custom EULA and Privacy Policy for the previous field.</td>
</tr>
<tr>
<td>Privacy Policy URL</td>
<td>URL where app users can view your custom privacy policy from the mobile app settings. This option only displays when you select Use custom EULA and Privacy Policy for the previous field.</td>
</tr>
<tr>
<td><strong>Android Branding Setup</strong></td>
<td></td>
</tr>
<tr>
<td>Firebase app ID</td>
<td>App ID of the Android app in your Google Firebase project. The ID is the unique identifier for the Firebase app across all Firebase projects. App IDs always contain the name :android: within the ID.</td>
</tr>
</tbody>
</table>

⚠️ **Note:**

- Each Firebase app ID is unique to the branded app request. If you request multiple apps, each app must have its own unique Firebase app ID.
- The package name of the branded app defaults tocom.servicenow.b2b.<app_name_long_no_spaces>.<client_type>.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Firebase API Key</strong></td>
<td>API key of the Android app in your Google Firebase project. This key is a string that is used when calling Firebase APIs. For more information, see the <a href="https://firebase.google.com">Google Firebase documentation</a>.</td>
</tr>
<tr>
<td><strong>Firebase project ID</strong></td>
<td>ID of the Google Firebase project. This is a user-assigned unique identifier for your Firebase project. For more information, see the <a href="https://firebase.google.com">Google Firebase documentation</a>.</td>
</tr>
<tr>
<td><strong>Trusted certificates authorities (CAs)</strong></td>
<td>Option that enables you to opt in your app to the trusted user-added certificate authority. By default, apps that target API level 24 do not honor user-supplied CAs. Select the check box to reduce the application attack surface and to encourage consistent handling of network and file-based application data.</td>
</tr>
<tr>
<td><strong>Theming</strong></td>
<td>The colors and icons that your app uses. Custom apps can either have the default ServiceNow branding or their own custom branding. If custom branding is selected, you must provide the theming details. For example, icon images, splash screen image, and branding colors.</td>
</tr>
<tr>
<td><strong>Adaptive Foreground Launcher Icon</strong></td>
<td>Foreground launcher icon for your app. The image file must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>• File must be in the PNG format.</td>
</tr>
<tr>
<td></td>
<td>• Image size must be 2048 px by 2048 px.</td>
</tr>
<tr>
<td></td>
<td>• Image must be an icon of a company logo with no text. The icon must take up most of the image.</td>
</tr>
<tr>
<td></td>
<td>• Image must be on a transparent background.</td>
</tr>
<tr>
<td><strong>Adaptive Background Launcher Icon</strong></td>
<td>Background launcher icon for your app. The image file must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>• File must be in the PNG format.</td>
</tr>
<tr>
<td></td>
<td>• Image size must be 2048 px by 2048 px.</td>
</tr>
<tr>
<td></td>
<td>• Image must have no logo or text.</td>
</tr>
<tr>
<td><strong>Android Notification Icon</strong></td>
<td>Notification icon for your app. This icon is used as the push notification icon on the notification screen. The image file must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>• File must be in the PNG format.</td>
</tr>
<tr>
<td></td>
<td>• Image size must be 2048 px by 2048 px.</td>
</tr>
<tr>
<td></td>
<td>• Image must contain the company logo with no text on a transparent background.</td>
</tr>
<tr>
<td><strong>Android Splash Screen</strong></td>
<td>Splash screen for your app. The image file must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>• File must be in the PNG format.</td>
</tr>
<tr>
<td></td>
<td>• Image size must be 2048 px by 2048 px.</td>
</tr>
<tr>
<td></td>
<td>• Image must contain the company logo and name on a transparent background.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android Splash Screen Background Color</td>
<td>Colored box where you select a color for the splash screen background color for your app.</td>
</tr>
</tbody>
</table>

4. Select **Submit**.

**Results**

You have successfully submitted a request for a new branded app. You will receive email notifications about the status of the build.

**Note:** You can only submit one build at a time. You cannot submit another build until the current build is canceled, failed, or completed.

**What to do next**

The ServiceNow® instance begins creating your app. To view the status of the build, navigate to **System Mobile > Request Branded App** and view the Latest Builds section below each app. Changes to your build status are sent to the email address in the **Email Address for Notifications** field.

After the build successfully completes, an email notification is sent to the requester contact for the build. The next step is to complete testing for the provided Android branded application.

The application file is provided as a download link within the Latest builds section. When you click the link, the APK file is downloaded to your local system for testing.

When you are ready to deploy the Android branded application, select the **Approve** button. This completes your request.

**Important:** The APK file that was used during testing should also be used for deployment.

**Request a new branded iOS app for private distribution**

Request and publish a custom iOS app that has your unique company identity. Then make the app available for private distribution.

**Before you begin**

Role required: admin

**About this task**

Mobile Publishing enables you to request a branded iOS app that you can distribute privately. After the request form is completed and the build is successful, you can use **TestFlight** to test the build. After testing is completed and you have approved the build of your app in your ServiceNow® instance, the app is published to your Apple Business Manager account. Then you can distribute the app to your employees with your Mobile Device Management (MDM) vendor or by using Apple Business Manager store redemption codes.

Images that do not follow the guidelines listed below might appear cut off in your branded app as shown in this example.
Procedure

1. Navigate to All > System Mobile > Request Branded App.
2. Select New Custom Branding.
3. On the form, fill in the fields.

### Branding Setup form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution</td>
<td>Type of distribution. Select <strong>Private/Enterprise</strong>.</td>
</tr>
<tr>
<td>Distribution Type</td>
<td>Mobile operating systems that you can select for your mobile applications.</td>
</tr>
<tr>
<td></td>
<td>Select <strong>iOS</strong>.</td>
</tr>
<tr>
<td></td>
<td>Selecting an option reveals the branding setup fields for that operating</td>
</tr>
<tr>
<td></td>
<td>system.</td>
</tr>
</tbody>
</table>

### Requester's Contact Information

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td>First name for the requester of the branded application.</td>
</tr>
<tr>
<td>Last Name</td>
<td>Last name for the requester of the branded application.</td>
</tr>
<tr>
<td>Email</td>
<td>Email address that you use to receive notifications about the build status of</td>
</tr>
<tr>
<td></td>
<td>your request.</td>
</tr>
<tr>
<td>Add Testers</td>
<td>Testers for the iOS app. For iOS app branding, users are required to receive</td>
</tr>
<tr>
<td></td>
<td>notifications from Apple TestFlight. To add a tester for your application,</td>
</tr>
<tr>
<td></td>
<td>select the add icon (➕). Your testers receive notifications from TestFlight</td>
</tr>
<tr>
<td></td>
<td>to test the app. For more information on TestFlight, see the Apple documentation.</td>
</tr>
</tbody>
</table>

**Note:** To access completed iOS apps, you must use the email address associated with your Apple ID.

### Brand Your App

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long Name</td>
<td>Unique name for your app. The name cannot exceed 30 characters and must be unique. This name displays for your app in the Apple B2B app store.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Short Name</td>
<td>Name of the app icon. The name cannot exceed 13 characters. This name displays below the app icon on your mobile device.</td>
</tr>
</tbody>
</table>
| MAM Support            | Type of MAM (Mobile App Management) vendor that you want to use with your apps. Choices are as follows:  
  - **None**: App does not use MAM.  
  - **Blackberry**: App uses Blackberry MAM.  
  - **Microsoft Intune**: App uses Intune MAM.                                                                                                                                                                                                                                     |
| In-app Color Branding  | Mobile color theme on your instance. For information on configuring a color theme for your app, see [Mobile themes](#).                                                                                                                                                                                                                           |
| Native Client          | Native client to which the color theme is assigned.                                                                                                                                                                                                                                                                                           |
| Client Theme           | Color theme assigned to the selected native client. In-app color branding uses the configured client theme from the instance. See [Mobile themes](#).                                                                                                                                                                                                 |
| Universal Links Host   | Universal links enable users to follow links to content inside your app. The displayed URL is for the production instance where your app is hosted.  
  - To change the existing URL, refer to the Global Settings section of the Brand Your Mobile App configuration page. Navigate to [System Mobile > Request Branded App](#).  
  - For more information about iOS Universal Links, see the [iOS Universal Links](#) [KB0831629](#) knowledge article in the Now Support Knowledge Base.  
  - For more information on how to configure apple-app-site-association files on a customer's instance, see the [iOS Universal Links Support](#) [KB0831613](#) knowledge article in the Now Support Knowledge Base. |
<p>| EULA / Privacy Policy  | End User License Agreement (EULA) and privacy policy options. You can use the default ServiceNow EULA and privacy policy or you can use a custom version. The ServiceNow version is selected by default.                                                                                                                                                      |
| EULA URL               | URL where app users can view your custom end user license agreement from the mobile app settings. This option only displays when you select <a href="#">Use custom EULA and Privacy Policy</a> for the previous field.                                                                                                                       |
| Privacy Policy URL     | URL where app users can view your custom privacy policy from the mobile app settings. This option only displays when you select <a href="#">Use custom EULA and Privacy Policy</a> for the previous field.                                                                                                                       |
| <strong>iOS Branding Setup</strong> |                                                                                                                                                                                                                                                                                                                                             |
| Apple DEP ID/ Organization ID | Organization ID is available in the settings of your Apple Business Manager (ABM) account. This ID is used to publish your branded iOS app to your ABM account.                                                                                                                                 |
| Organization Name      | Organization name that is available in the settings of your Apple Business Manager account. This name is used to publish your branded iOS app to your ABM account.                                                                                                                                                           |
| <strong>Important:</strong>         | This field value is case sensitive and entering it incorrectly results in a failed build request.                                                                                                                                                                                                                                           |
| Theming                | The colors and icons for your app. Custom apps can either have the default ServiceNow branding or their own custom branding.                                                                                                                                                                                                                 |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>If custom branding is selected, you must provide the theming details. For example, icon images, splash screen image, and branding colors.</td>
<td></td>
</tr>
<tr>
<td><strong>App Icon</strong></td>
<td>Foreground launcher icon for your app. The image file must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>• File must be in the PNG format.</td>
</tr>
<tr>
<td></td>
<td>• Color space must be sRGB.</td>
</tr>
<tr>
<td></td>
<td>• Layers must be flattened.</td>
</tr>
<tr>
<td></td>
<td>• Image must not have any transparency.</td>
</tr>
<tr>
<td></td>
<td>• Image must be a 1024 px by 1024 px square, with no rounded corners.</td>
</tr>
<tr>
<td><strong>iOS Splash Screen</strong></td>
<td>Splash screen for your app. The image file must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>• File must be in the PNG format.</td>
</tr>
<tr>
<td></td>
<td>• Image size must be 2048 px by 2048 px.</td>
</tr>
<tr>
<td></td>
<td>• Image must contain the company name and logo on a transparent background.</td>
</tr>
<tr>
<td><strong>iOS Splash Screen Background Color</strong></td>
<td>Colored box where you select a color for the splash screen background color for your app.</td>
</tr>
</tbody>
</table>

4. Select **Submit**.

**Results**
You have successfully submitted a request for a new branded app. You will receive email notifications regarding the status of the build.

⚠️ **Note:** You can only submit one build at a time. You cannot submit another build until the current build is canceled, failed, or completed.

**What to do next**
The ServiceNow instance begins creating your app. To view the status of the build, navigate to **System Mobile > Request Branded App** and view the Latest Builds section below each app. Changes to your build status are sent to the email address in the **Email Address for Notifications** field.

After the build successfully completes, an email notification is sent to the requester contact for the build. Then you must complete testing for the provided iOS branded application.

If **Tester** information was entered in the branding setup form, you should receive an invitation from Apple TestFlight. You can use this invitation to test your iOS branded app.

When you are ready to deploy the iOS branded application, select the **Approve** button. Your request is completed. A process is initiated so that your ServiceNow instance publishes the app to your Apple Business Manager account.

**Request a new branded Android app for public distribution**
Request and publish a custom Android app that has your unique company identity. Then make the app available for public distribution on the Google Play Store.

**Before you begin**
Role required: admin
About this task
Mobile Publishing enables you to request a branded Android app that can be distributed on the public Google Play Store. After you complete the request form, your ServiceNow® instance builds the app and provides a link from where you can download an Android App Bundle file (.aab). Then you download the AAB file and test it. When testing completes successfully, you approve the build and can deploy your branded Android app with the public Google Play Store. For information about AAB files, see the Google documentation.

⚠️ **Note:** It might take 1–2 weeks to build the Android branded application for public distribution.

Ensure that the images or text in the icons are at least 550 pixels from the edge. This distance accounts for the rounded corners in Android icons.

Images that do not follow this guideline may appear cut off, as shown in this example.

---

**Procedure**
1. Navigate to All > System Mobile > Request Branded App.
2. Select **New Custom Branding**.
3. On the form, fill in the fields.
### Branding Setup form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Distribution</strong></td>
<td></td>
</tr>
<tr>
<td>Distribution Type</td>
<td>Type of distribution. Select <strong>Public</strong>. A dialog box displays with additional terms for public Mobile Publishing. After reading the terms, select <strong>I Accept</strong> to continue with your branded app request.</td>
</tr>
<tr>
<td><strong>Platforms</strong></td>
<td></td>
</tr>
<tr>
<td>Deploy app on</td>
<td>Mobile operating system for your mobile application. Select <strong>Android</strong>. Selecting an option reveals the branding setup fields for that operating system.</td>
</tr>
<tr>
<td><strong>Requester’s Contact Information</strong></td>
<td></td>
</tr>
<tr>
<td>First Name</td>
<td>First name for the requester of the branded application.</td>
</tr>
<tr>
<td>Last Name</td>
<td>Last name for the requester of the branded application.</td>
</tr>
<tr>
<td>Email</td>
<td>Email address that you use to receive notifications about the build status of your request.</td>
</tr>
<tr>
<td><strong>Brand Your App</strong></td>
<td></td>
</tr>
<tr>
<td>Short Name</td>
<td>Name of the app icon. This name appears under the icon on your mobile device. The name cannot exceed 13 characters. This name displays below the app icon on mobile devices.</td>
</tr>
<tr>
<td>MAM Support</td>
<td>Type of MAM (Mobile App Management) vendor that you want to use with your app. Choices are as follows:</td>
</tr>
<tr>
<td></td>
<td>* None*: App does not use MAM.</td>
</tr>
<tr>
<td></td>
<td>* Blackberry*: App uses Blackberry MAM.</td>
</tr>
<tr>
<td></td>
<td>* Microsoft Intune*: App uses Intune MAM.</td>
</tr>
<tr>
<td>In-app Color Branding</td>
<td>Mobile color theme on your instance. For information on configuring a color theme for your app, see <strong>Mobile themes</strong>.</td>
</tr>
<tr>
<td>Native Client</td>
<td>Native client to which the color theme is assigned.</td>
</tr>
<tr>
<td>Client Theme</td>
<td>Color theme that is assigned to the selected native client. In-app color branding uses the configured client theme from the instance. See <strong>Mobile themes</strong>.</td>
</tr>
<tr>
<td>Universal Links Host</td>
<td>URL of your instance. Universal Links allow your users to follow links to content inside your app.</td>
</tr>
<tr>
<td>EULA / Privacy Policy</td>
<td>End User License Agreement (EULA) and privacy policy options. You can use the default ServiceNow EULA and privacy policy or you can use a custom version. Your custom version is selected by default.</td>
</tr>
<tr>
<td>EULA URL</td>
<td>URL where app users can view your custom end-user license agreement from the mobile app settings. This option appears only when <strong>Use custom EULA and Privacy Policy</strong> is selected from the <strong>EULA / Privacy Policy</strong> field.</td>
</tr>
<tr>
<td>Privacy Policy URL</td>
<td>URL where app users can view your custom privacy policy from the mobile app settings. This option appears only when <strong>Use custom EULA and Privacy Policy</strong> is selected from the <strong>EULA / Privacy Policy</strong> field.</td>
</tr>
</tbody>
</table>

© 2022 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>Firebase app ID</td>
<td>App ID of the Android app in your Google Firebase project. The ID is the unique identifier for the Firebase app across all Firebase projects. App IDs always contain the name :android: within the ID.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>• Each Firebase app ID is unique to the branded app request. If you request multiple apps, each app must have its own unique Firebase app ID.</td>
</tr>
<tr>
<td></td>
<td>• The package name of the branded app defaults to com.servicenow.public.&lt;app_name_short_no_spaces&gt;..&lt;client_type&gt;.</td>
</tr>
<tr>
<td></td>
<td>For more information on Firebase, see the <a href="#">Google Firebase documentation</a>.</td>
</tr>
<tr>
<td>Firebase API Key</td>
<td>API key of the Android app in your Google Firebase project. This key is a string that is used when calling Firebase APIs. For more information, see the <a href="#">Google Firebase documentation</a>.</td>
</tr>
<tr>
<td>Firebase project ID</td>
<td>ID of the Google Firebase project. This ID is a user-assigned unique identifier for your Firebase project. For more information, see the <a href="#">Google Firebase documentation</a>.</td>
</tr>
<tr>
<td>Trusted certificates authorities (CAs)</td>
<td>Option that enables you to opt in your app to the trusted user-added certificate authority. By default, apps that target API level 24 do not honor user-supplied CAs. Select the check box to reduce the application attack surface and to encourage consistent handling of network and file-based application data.</td>
</tr>
<tr>
<td>Theming</td>
<td>The colors and icons that your app uses. Custom apps can either have the default ServiceNow branding or their own custom branding. If you select custom branding, you must provide the theming details. For example, icon images, splash screen image, and branding colors.</td>
</tr>
<tr>
<td>Adaptive Foreground Launcher Icon</td>
<td>Foreground launcher icon for your app. The image file must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>• File must be in the PNG format.</td>
</tr>
<tr>
<td></td>
<td>• Image size must be 2048 px by 2048 px.</td>
</tr>
<tr>
<td></td>
<td>• Image must be an icon of a company logo with no text. The icon must take up most of the image.</td>
</tr>
<tr>
<td></td>
<td>• Image must be on a transparent background.</td>
</tr>
<tr>
<td>Adaptive Background Launcher Icon</td>
<td>Background launcher icon for your app. The image file must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>• File must be in the PNG format.</td>
</tr>
<tr>
<td></td>
<td>• Image size must be 2048 px by 2048 px.</td>
</tr>
<tr>
<td></td>
<td>• Image must have no logo or text.</td>
</tr>
<tr>
<td>Android Notification Icon</td>
<td>Notification icon for your app. This icon is used as the push notification icon on the notification screen. The image file must meet the following requirements:</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>• File must be in the PNG format.</td>
</tr>
<tr>
<td></td>
<td>• Image size must be 2048 px by 2048 px.</td>
</tr>
<tr>
<td></td>
<td>• Image must contain the company logo with no text on a transparent background.</td>
</tr>
<tr>
<td>Android Splash Screen</td>
<td>Splash screen for your app. The image file must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>• File must be in the PNG format.</td>
</tr>
<tr>
<td></td>
<td>• Image size must be 2048 px by 2048 px.</td>
</tr>
<tr>
<td></td>
<td>• Image must contain the company logo and name on a transparent background.</td>
</tr>
<tr>
<td>Android Splash Screen Background Color</td>
<td>Colored box where you select a color for the splash screen background color for your app.</td>
</tr>
</tbody>
</table>

4. Select **Submit**.

**Results**
After you complete the request form and submit it, your ServiceNow instance builds the app. When it completes, you receive an email notification about the status of the build. A link is provided from where you can download an Android App Bundle file (.aab).

ℹ️ **Note:** You can only submit one build at a time. You cannot submit another build until the current build is canceled, failed, or completed.

**What to do next**
After the build successfully completes, an email notification is sent to the requester contact for the build. Download the AAB file and test it. If the testing completes successfully, you can approve the build and deploy your new custom branded app with the Google Play Store.

For information about deploying an app with the Google Play Store, see the [Google documentation](https://developer.android.com). 

**Request a new branded iOS app for public distribution**
Request and publish a custom iOS app that has your unique company identity. Then make the app available for public distribution on the Apple store.

**Before you begin**
Role required: admin

**About this task**
Mobile Publishing enables you to request a new branded iOS app that can be distributed on the public Apple Store. After you complete the request form, your ServiceNow® instance builds the app and provides a link from where you can download an XCode archive file (.xcarchive). Then you download the XCode archive file, update it, and re-sign it to generate an iOS application archive file (.ipa). After the .ipa file is generated, you upload it to Apple Connect for testing. When testing is completed, you approve the build and can deploy your branded app with the public Apple Store.

ℹ️ **Note:** It might take 1 to 2 weeks to build the iOS branded application for public distribution.
Images that do not follow the guidelines listed below might appear cut off in your branded app, as shown in this example.

**Procedure**
1. Navigate to All > System Mobile > Request Branded App.
2. Select New Custom Branding.
3. On the form, fill in the fields.

**Branding Setup form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Distribution</strong></td>
<td></td>
</tr>
<tr>
<td>Distribution Type</td>
<td>Type of distribution. Select <strong>Public</strong>. A dialog box displays with additional terms for public Mobile Publishing. After reading the terms, select I Accept to continue with your branded app request.</td>
</tr>
<tr>
<td><strong>Platforms</strong></td>
<td></td>
</tr>
<tr>
<td>Deploy app on</td>
<td>Mobile operating systems that you can select for your mobile applications. Select <strong>iOS</strong>. Selecting an option reveals the branding setup fields for that operating system.</td>
</tr>
<tr>
<td><strong>Requester’s Contact Information</strong></td>
<td></td>
</tr>
<tr>
<td>First Name</td>
<td>First name for the requester of the branded application.</td>
</tr>
<tr>
<td>Last Name</td>
<td>Last name for the requester of the branded application.</td>
</tr>
<tr>
<td>Email</td>
<td>Email address that you use to receive notifications about the build status of your request.</td>
</tr>
<tr>
<td><strong>Brand Your App</strong></td>
<td></td>
</tr>
<tr>
<td>Short Name</td>
<td>Name of the app icon. This name appears under the icon on your mobile device. The name cannot exceed 13 characters. This name displays below the app icon on your mobile device.</td>
</tr>
<tr>
<td>MAM Support</td>
<td>Type of MAM (Mobile App Management) vendor that you want to use with your apps. Choices are as follows:</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>In-app color branding</td>
<td>Mobile color theme on your instance. For information on configuring a color theme for your app, see Mobile themes.</td>
</tr>
<tr>
<td>Native Client</td>
<td>Native client to which the color theme is assigned.</td>
</tr>
<tr>
<td>Client Theme</td>
<td>Color theme assigned to the selected native client. In-app color branding uses the configured client theme from the instance. See Mobile themes.</td>
</tr>
<tr>
<td>Universal Links Host</td>
<td>Universal links enable users to follow links to content inside your app. The displayed URL is for the production instance where your app is hosted.</td>
</tr>
<tr>
<td></td>
<td>- To change the existing URL, refer to the Global Settings section of the Brand Your Mobile App configuration page. Navigate to System Mobile &gt; Request Branded App.</td>
</tr>
<tr>
<td></td>
<td>- For more information about iOS Universal Links, see the iOS Universal Links [KB0831629] knowledge article in the Now Support Knowledge Base.</td>
</tr>
<tr>
<td></td>
<td>- For more information on how to configure apple-app-site-association files on a customer's instance, see the iOS Universal Links Support [KB0831613] knowledge article in the Now Support Knowledge Base.</td>
</tr>
<tr>
<td>EULA / Privacy Policy</td>
<td>End User License Agreement (EULA) and privacy policy options. You can use the default ServiceNow EULA and privacy policy or you can use a custom version. Your custom version is selected by default.</td>
</tr>
<tr>
<td>EULA URL</td>
<td>URL where app users can view your custom end user license agreement from the mobile app settings. This option only displays when you select Use custom EULA and Privacy Policy for the previous field.</td>
</tr>
<tr>
<td>Privacy Policy URL</td>
<td>URL where app users can view your custom privacy policy from the mobile app settings. This option only displays when you select Use custom EULA and Privacy Policy for the previous field.</td>
</tr>
</tbody>
</table>

## iOS Branding Setup

### Theming

The colors and icons for your app. Custom apps can either have the default ServiceNow branding or their own custom branding.

If custom branding is selected, you must provide the theming details. For example, icon images, splash screen image, and branding colors.

### App Icon

Foreground launcher icon for your app. The image file must meet the following requirements:

- File must be in the PNG format.
- Color space must be sRGB.
- Layers must be flattened.
- Image must not have any transparency.
- Image must be a 1024 px by 1024 px square, with no rounded corners.

### iOS Splash Screen

Splash screen for your app. The image file must meet the following requirements:
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>• File must be in the PNG format.</td>
<td></td>
</tr>
<tr>
<td>• Image size must be 2048 px by 2048 px.</td>
<td></td>
</tr>
<tr>
<td>• Image must contain the company name and logo on a transparent background.</td>
<td></td>
</tr>
<tr>
<td>iOS Splash Screen Background Color</td>
<td>Colored box where you select a color for the splash screen background color for your app.</td>
</tr>
</tbody>
</table>

4. Select Submit.

Results
After you complete the request form, your ServiceNow instance builds the app and provides a link from where you can download an XCode archive file (.xcarchive).

What to do next
Download the XCode archive file, update it, and re-sign it to generate an iOS application archive file (.ipa). After the .ipa file is generated, you upload it to Apple Connect for testing. When testing is completed, you approve the build and can deploy your branded app with the public Apple Store.

For information about updating the .xcarchive file and creating the .ipa file, see the How to update XCARCHIVE and create IPA branded iOS app for public distribution [KB0997467] article in the Knowledge Base.

Note: It might take 1 to 2 weeks to build the iOS branded application for public distribution.

Update a branded app
Request and publish an update to a ServiceNow® mobile app that has been branded with your unique company identity. Updates can only be requested for approved, completed builds of branded apps.

Before you begin
Role required: admin

About this task
Mobile Publishing enables you to request an updated version of your branded iOS or Android app that you can distribute publicly or privately. You can request an update for the following branded app options:

• New version of the ServiceNow mobile app.
• Requester contact information such as first name, last name, and email address.
• Change whether you want to use your custom End User License Agreement (EULA) and privacy policy or the default ServiceNow EULA and privacy policy.
• For iOS apps that use private distribution, you can add or delete testers for TestFlight notifications.

Form options that are grayed out cannot be changed. If you must change those options, you must request a new branded app with Mobile Publishing.
Procedure

1. Navigate to All > System Mobile > Request Branded App.

2. Select Update Custom Branding.
   A dialog box displays that offers different builds from which to choose. The builds are identified with the request date.

3. Select a build from Choose a build, and then select Continue.
   The form pre-fills all the options except for the options that you can update.

4. On the form, fill in the fields.

Request new branding form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requester's Contact Information</td>
<td></td>
</tr>
<tr>
<td>First Name</td>
<td>First name for the requester of the updated branded application.</td>
</tr>
<tr>
<td>Last Name</td>
<td>Last name for the requester of the updated branded application.</td>
</tr>
<tr>
<td>Email</td>
<td>Email address that you use to receive notifications about the build status of your updated branded app request.</td>
</tr>
<tr>
<td>Add Testers</td>
<td>Testers for the privately distributed iOS app. This field is available only for iOS privately distributed apps. For iOS app branding, users must receive notifications from Apple TestFlight. You can do any of the following:</td>
</tr>
<tr>
<td></td>
<td>• To add a tester for your updated application, select the add (ıld) icon. Your testers receive notifications from TestFlight to test the app.</td>
</tr>
<tr>
<td></td>
<td>• To remove a tester, select the delete (ğı) icon next to the tester's information. For more information on TestFlight, see the Apple documentation.</td>
</tr>
<tr>
<td>Note:</td>
<td>The email addresses for testers must be the email address associated with their Apple ID.</td>
</tr>
</tbody>
</table>

Brand Your App

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EULA / Privacy Policy</td>
<td>End User License Agreement (EULA) and privacy policy options for your updated branded app. You can use the default ServiceNow EULA and privacy policy or you can use a custom version.</td>
</tr>
<tr>
<td>EULA URL</td>
<td>URL where app users can view your custom end-user license agreement from the mobile app settings. This option appears only when Use custom EULA and Privacy Policy is selected from the EULA / Privacy Policy field.</td>
</tr>
<tr>
<td>Privacy Policy URL</td>
<td>URL where app users can view your custom privacy policy from the mobile app settings. This option appears only when Use custom EULA and Privacy Policy is selected from the EULA / Privacy Policy field.</td>
</tr>
</tbody>
</table>

5. Select Submit.

Results

After you complete the request form and submit it, your ServiceNow instance builds the updated branded app. When it completes, you receive an email notification about the status of the build. A link is provided from where you can download the app file.
Note: You can only submit one build at a time. You cannot submit another build until the current build is canceled, failed, or completed.

What to do next
After the build successfully completes, an email notification is sent to the requester contact for the build. Download the app file and test it. If the testing completes successfully, you can approve the build. Then you or your ServiceNow instance performs one of the following actions to publish the app:

- For privately distributed Android apps, you can deploy the tested, approved APK file to your private distribution site.
- For privately distributed iOS apps, after you select the Approve button, your ServiceNow instance publishes the app to your Apple Business Manager account.
- For publicly distributed Android apps, you can deploy the tested, approved AAB file to the Google Play Store.
  For information about deploying an app with the Google Play Store, see the Google documentation.
- For publicly distributed iOS apps, you can deploy your tested, approved app with the public Apple Store.
  For information about deploying an app with the Apple Store, see the Apple documentation.

Building and configuring in branded mobile apps
Learn about configuring push notifications for branded mobile apps.

Most app configuration processes for branded apps are the same as non-branded apps. Use the following sections to learn about configuration processes that are unique to branded mobile apps.

Push Notifications
To use push notifications for your branded apps, you must change your push application settings.
- For iOS apps, see Configure push applications for iOS branded apps.
- For Android apps, see Configure push applications for Android branded apps.

Mobile themes
Use themes on your mobile applications so that your customers and employees can recognize your unique company identity.
Use mobile themes to change the color scheme of your mobile apps. The colors for elements such as headers, links, buttons, and icons can be controlled using themes. You can configure each of your mobile apps with a unique theme.

For details on creating a theme, see Create a theme for your mobile applications.

To see how your selected theme colors affect the appearance of your mobile apps, see the Mobile theme color guidelines.

⚠️ Note: You can configure themes on your apps without custom branding.
Virtual Agent theming

Theming for virtual agent is handled separately from mobile theming. For details on configuring the appearance of Virtual Agent, see Configure chat branding and the chat menu.

Create a theme for your mobile applications

Use mobile themes to give your applications a unique appearance that works with your company identity.

Before you begin

Role required: admin

Procedure

1. Navigate to All > System Mobile > Mobile App Builder.
   The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.

2. Search for the application scope you are working in and then select the name of the application scope.
   The Mobile App Builder categories home screen displays.

3. Select All mobile records from the menu.

4. From the Record type field, select Mobile app theme [sys_sg_client_theme], and then select New.
   The new mobile app theme screen displays where you can create a new theme.

5. Complete the following fields as needed.

<table>
<thead>
<tr>
<th>Mobile app theme form</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Field</strong></td>
</tr>
<tr>
<td><strong>Properties section:</strong></td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td><strong>Palette section:</strong></td>
</tr>
<tr>
<td>Name/Value</td>
</tr>
<tr>
<td>Field</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>
</tbody>
</table>

6. Select **Save**.
   
   Your theme is ready to use on your instance.

**What to do next**

After you configure a mobile theme, you can apply the mobile theme to your mobile apps. For details, see [Assign a theme to your mobile application](#).

**Assign a theme to your mobile application**

After you configure a mobile theme that reflects your company identity, you can assign the theme to your mobile applications.

**Before you begin**

Role required: admin

**Procedure**

1. Navigate to **All > System Mobile > Mobile App Builder**.
   
   The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.

2. Search for the application scope you are working in and then select the name of the application scope.
   
   The Mobile App Builder categories home screen displays.

3. Select **Mobile app configs** from the menu.

4. Open the mobile app configuration where you want to apply your theme.

5. Scroll down to locate the Client Theme section, select **Choose**, and then select the desired mobile app theme record.
   
   For information about creating a mobile theme, see [Create a theme for your mobile applications](#).

6. Select **Save**.

**Mobile theme color guidelines**

Learn how the colors that you select in your mobile themes are applied on your mobile applications.

**Color choice considerations**

Consider the following information when you select colors for your mobile applications:
• How your color choices look with colors in other parts of the app, such as your applet icons and UI styles. Avoid using too many colors. Simple designs of two to three colors result in a cleaner, more accessible layout.

• How your color choices look with colors in other elements that may display within your app. Elements like knowledge articles and service portal pages may appear in your apps. Avoid color schemes that don’t work well with your existing elements.

• How higher contrast, complementary colors can guide your user’s focus. Use color selection to highlight areas of your apps that require your user’s attention.

• How color can be used to provide contrast for greater readability. In areas where text appears, ensure that your text and background colors have a high level of contrast to accommodate both low-vision and color-blind users.

Theme color locations in mobile apps

This example shows the colors that are used in the default mobile app theme. Use the numbers in this image to see where these colors appear in the different areas of your mobile apps.

<table>
<thead>
<tr>
<th>Color</th>
<th>Description</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Brand color - Gable Green - #293E40</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Primary - Tropical Rain - #1F8476</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Secondary - Evening Sea - #165C53</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Tertiary - Gable Green - #293E40</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>Positive - Positive-3 - #3B7F00</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Destructive - Mahogany - #C83C36</td>
<td></td>
</tr>
</tbody>
</table>

Mobile Features

Use the following figures to see how your color choices affect the elements within your mobile apps.

Application Launch Page
Navigation Bar
Business overview

Value
Risk & resiliency
Execution

Urgent notifications

2
Upcoming releases

2
Service degradation

2
Projects

Digital portfolios

For Me
Team
Saved
Support
More
19:15

Done

Now Virtual Agent

Just now

How can I help you?

You can use the buttons or type your request below.

Email Issues

What type of issue are you experiencing?

Can't send email

Here are a few things you can try:
1) Check if the email you are trying to send doesn't have any attachments that exceed 25MB.
2) If it doesn't, try closing and reopening your email client.
3) If closing and reopening your email client doesn't work, try removing your account and adding it back again.

Did these steps work for you?

Yes

Thank you for using our support chat.

Start new conversation

Search
Actions

iOS

Android
Form screen and components

Form screen

Checklist
ServiceNow

On your first day of employment, you'll participate in our New Hire Orientation program.

Prepare for Orientation

2. Complete new hire tasks

○ Update LinkedIn with your new job information

○ Plan the commute to your office location

○ Review orientation details

Activity stream
Links
E-signature
Applet templates
Employee Directory
Calendar
地图
Note: The Pin color field in the item stream record that is associated to your map applet determines the color of the pins on your maps.

URL template
Settings and notifications

Notifications
Offline mode settings
Geolocation settings

PIN settings
Siri shortcuts settings (iOS Only)

Modals

Android modals
iOS modals
**Button emphasis considerations**

Use button emphasis to define the appearance and color of function buttons throughout your mobile app. The appearance and color of function buttons helps users identify the types of actions they trigger.

Select the mobile theme colors for button emphasis throughout your mobile app. You can define different colors to compliment your branding guidelines and you can vary your selection according to different locations on your mobile app.

The available theme colors for button emphasis are primary, secondary, tertiary, positive, and destructive. The following diagram shows the default colors that are available. For more information, see [Mobile theme color guidelines](#).

![Default colors for button emphasis](image)

**Configure button emphasis according to function location**

Configure button emphasis in different locations within your mobile app to help users identify the type of functions they can activate.

**Before you begin**

Role required: admin

**About this task**

Button emphasis configuration is performed in Mobile Studio. For more information, see [Associate a function with a location in the app](#). Starting from San Diego additional colors are included in the release, and button emphasis is supported in additional locations. These colors are configured in the web-based UI for the listed screen locations.

**Procedure**

1. Navigate to All.
2. In the filter navigator, enter `sys_sg_button_instance.list`, to open the Function instances table.
3. Select either a new or existing record.
4. Select a button emphasis for the chosen function location type.
### Function location

<table>
<thead>
<tr>
<th>Function location</th>
<th>Action</th>
</tr>
</thead>
</table>
| Swipe action     | a. In the **Location** field, select either **Trailing swipe** or **Leading swipe**.  
|                  | b. In the **Button emphasis** field, select either **Destructive** or **Positive**. |
|                  | Note: If you select **None**, the color is dynamically assigned based on the swipe button location. This color can be defined as either primary, secondary, or tertiary. |
|                  | c. Configure the other fields in the table as required.               |
|                  | d. Select **Submit**.                                                 |

<table>
<thead>
<tr>
<th>Top menu action or Quick action function</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a. In the <strong>Location</strong> field, select either <strong>Top</strong> or <strong>Quick Action</strong></td>
</tr>
<tr>
<td></td>
<td>b. In the <strong>Button emphasis</strong> field, select either <strong>Destructive</strong> or <strong>Positive</strong>.</td>
</tr>
<tr>
<td></td>
<td>Note: If you select <strong>None</strong>, the color is the primary theme color.</td>
</tr>
<tr>
<td></td>
<td>c. Configure the other fields in the table as required.</td>
</tr>
<tr>
<td></td>
<td>d. Select <strong>Submit</strong>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Footer function or Media section</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a. In the <strong>Location</strong> field, select <strong>Footer</strong>.</td>
</tr>
<tr>
<td></td>
<td>b. In the <strong>Button emphasis</strong> field, select either <strong>Primary</strong>, <strong>Destructive</strong>, <strong>Secondary</strong>, or <strong>Positive</strong>.</td>
</tr>
<tr>
<td></td>
<td>Note: If you select <strong>None</strong>, the color is the primary theme color.</td>
</tr>
<tr>
<td></td>
<td>c. Configure the other fields in the table as required.</td>
</tr>
</tbody>
</table>
**Mobile security**

Learn about the security features of the ServiceNow mobile platform.

**ServiceNow mobile architecture**

ServiceNow mobile apps consist of ServiceNow server instance and native apps for iOS and Android. The apps use full native code and are not a hybrid approach. The mobile apps transmit and receive data with the server across the wireless network.
Overview of key features of ServiceNow mobile platform security

- The mobile apps rely on the secure ServiceNow platform and its APIs to provide a seamless mobile experience to its users.
- App/server interactions are protected through OAuth authorization framework.
- Most of the user interface on the ServiceNow app is driven through meta data delivered by the ServiceNow platform.
- The ServiceNow mobile apps fetch all their data from the ServiceNow platform and store it in a local cache on the app client layer.
- For government community cloud (GCC) ServiceNow instances, locally data stored is encrypted.
- For iOS apps, ServiceNow uses the OS level FIPS 140-2 validated disk encryption on Core Data, by forcing a device level PIN or Biometrics security.
- For Android apps, ServiceNow uses the SQLCipher SDK. This SDK provides encryption using FIPS 140-2 validated crypto module for all the app data stored in Room DB.

**App flow overview**

ServiceNow mobile apps start fetching the initial user experience after a successful sign-in. The mobile app fetches the metadata to render the landing home screen from the instance. The app then uses this metadata to render the home screen.
Data retrieval

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 instance directs the request to the relevant API to fetch the information.

4. The instance returns the information 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. The instance checks the access control list (ACL) rules.

4. If valid, the document is available to view.

**Write-backs for 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 to the instance. For example, the ID, or the field to update.

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.

**Write-backs for attaching files**

When attaching files, the following steps occur.

1. The mobile app prompts the user to attach a file, for example, an image.

2. The mobile app sends the file and token to the instance.

3. The instance places the file based on the relevant API.

4. The instance sends a response back to the mobile app.

**Mobile authentication**

ServiceNow mobile apps support platform authentication using OAuth 2.0. Authentication mechanisms include multi provider SSO, MFA, LDAP, Local DB, and Digest. ServiceNow mobile apps use an authentication methodology called AppAuth. AppAuth uses an external mobile browser to log the user in.

**Authentication flow**

When a user signs in to an app on their mobile device, the app uses the user’s credentials to negotiate an OAuth Token with the instance. The iOS Keychain stores the token for iOS devices. Android device use KeyStore. The keychain encryption is AES 256 in Galois/Counter Mode (GCM).

At first login, the instance gives the user an access token and a refresh token. These tokens are valid for an amount of time that can be configured on your
instance. When a user opens the mobile app, the client checks to see if the access token is valid. If valid, the user can 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.

**Access and refresh tokens**

- The mobile apps never stores the user password.
- The mobile apps do store the client ID, which is necessary for getting the OAuth token as part of the authentication flow.

**User termination**

- When an administrator deletes or removes a user from the instance, the access token is no longer valid, and any operation logs the user out.
Mobile authentication workflow

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.

Multifactor authentication

Users can access the instance via Multifactor Authentication using the MFA plugin [com.snc.integration.multifactor.authentication]. The mobile app directs users to their login page after selecting their instance in the mobile app.

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.

**Data security**

ServiceNow mobile apps use SSL/TLS over-the-air (OTA) communication encryption for data security. 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 apps do not store record data such as incidents and problems on the device unless your organization has specifically enabled offline syncing for field service. Record data stored during offline mode is encrypted with FIPS 140-2 validated modules. (iOS cryptographic modules and SQL Cipher for Android which uses this cryptographic module for encryption).

**Data at rest**

![Diagram showing data at rest for iOS and Android](image)

**Data in motion**

Data in motion is over a secure SSL/TLS channel and encrypted with FIPS 140-2 validated modules.

**Data loss prevention**

ServiceNow provides data loss prevention features without the need for the device and application to be managed by an enterprise mobility management (EMM) suite. These features includes restrict copy/paste, enforce PIN, block attachment, and/or blur functionality.

**Restrict copy/paste**

Copy/paste restrictions are defined by a property in the system properties table.

<table>
<thead>
<tr>
<th>System property field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>glide.sg.clear_pasteboard_when_background</td>
</tr>
<tr>
<td>Type</td>
<td>true</td>
</tr>
</tbody>
</table>
Require an app PIN

Require users to enter a six-digit PIN each time they sign in from their mobile device, or when the application has been inactive for five minutes. Requiring an app PIN is controlled a property in the system properties table.

<table>
<thead>
<tr>
<th>System property field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>glide.sg.require_mobile_application_pin</td>
</tr>
<tr>
<td>Type</td>
<td>true</td>
</tr>
<tr>
<td>Value</td>
<td>true</td>
</tr>
</tbody>
</table>

Disabling attachments on a mobile device

You can configure ACL rule to block attachments specifically on mobile devices. Use the isMobile method to check if a request comes from a mobile device. For example, you could add an ACL rule for the attachment [sys_attachment] table where the read and write scripted ACLs includes the following check.

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

You can also add this code to any existing ACL rules you have for the attachment table. If you have multiple attachment ACL rules, all will need to have Admin override option unchecked.

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 table.

<table>
<thead>
<tr>
<th>System property field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>glide.sg.blur_ui_when_backgrounded</td>
</tr>
<tr>
<td>Type</td>
<td>true</td>
</tr>
<tr>
<td>Value</td>
<td>true</td>
</tr>
</tbody>
</table>

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. For more detail on security testing, see KB0538598: Customer Instance Security Testing | Policy and Procedure.

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.
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

**Device security for ServiceNow Mobile apps**

This document applies to current ServiceNow apps for iOS and Android for San Diego. This document may be subject to change for future mobile releases.

**Components and architecture**

The ServiceNow mobile apps consist of the ServiceNow server instance and native apps for iOS and Android. The apps use fully native code and are 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**

**Apps for iOS**

The ServiceNow apps for iOS are fully native iPhone and iPad applications. The applications 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.

**Apps for Android**

The ServiceNow apps for Android is a fully native applications for Android phone. They 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 only supports apk file distribution to customers for branded versions of Android mobile apps. For details on branded applications, see [Mobile Publishing](#).

**Identity and access management**

Learn about user authentication, third party authentication, and user session termination for mobile applications.

**User authentication for ServiceNow mobile apps**

ServiceNow mobile apps support platform authentication using OAuth 2.0. Authentication mechanisms include multi provider SSO, MFA, LDAP, Local DB, and Digest.

**AppAuth authentication**

The ServiceNow mobile apps use 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**

![Diagram of the OAuth 2.0 protocol flow](image)

**Single sign-on**

ServiceNow mobile apps require multi-provider single sign-on in order to use external authentication. 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 this plugin, see [External single sign-on (SSO)]().

For more information on configuring multi provider SSO, see [Mobile single sign on].

**Multifactor authentication**

Users can access the instance via Multifactor Authentication using the MFA plugin [com.snc.integration.multifactor.authentication]. Users are directed to their login page after selecting their instance in a mobile app.
For details on configuring Multi-factor Authentication, see Multifactor authentication system properties

**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 an 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 256 in Galois/Counter Mode (GCM).

The ServiceNow mobile apps never store 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
ServiceNow mobile apps use 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 a 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 re-authenticate.

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

Mobile data flow for ServiceNow mobile apps
Data can be retrieved, downloaded from, and written back to a mobile device.

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 ServiceNow mobile apps.

Updating fields
When a user updates a field in a 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 ServiceNow mobile apps 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 files, or any other unpublished app to customers as it breaches the Apple Enterprise Developer License Agreement.

Mobile app distribution providers:

- AirWatch
- BlackBerry
- Intune
- IBM
- MobileIron

**Data security for ServiceNow mobile apps**

ServiceNow mobile apps use SSL/TLS for Over-the-Air (OTA) communication encryption for data security. The OAuth authorization endpoints are HTTPS.

**Data stored in your mobile apps**

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

Information stored in mobile apps

- **Databases**
  - User defined instances
  - Favorite application IDs
  - Push Notifications
  - Geolocation updates
  - Offline data
- **Preferences stored in mobile apps**
  - 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

Choose specific applets and actions to be enabled offline from with Studio. On the mobile device, your users can select offline and choose to “cache data” from Settings. The offline flows that you designate are downloaded and cached to the device.

You can encrypt offline cached data by using native encryption. This encryption expires at a specified period of time. The default is 48 hours or when a user signs out of the mobile app.

Offline data is protected by local-auth and the app PIN that can be optionally enabled by administrators. When enabled, users are required to enter a PIN on login, or when the application is inactive for five minutes.

Disabling mobile attachments

You can disable attachments for mobile apps by using access control rules. For more details on this process see Disable attachments in mobile apps.

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.sg.clear_pasteboard_when_background: Clears the copy/paste clipboard when a 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, or when an app has been inactive for five minutes. 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.

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 a mobile app when not in focus on a mobile device using the following system property in the system properties [sys_properties] table.

• glide.sg.blur_ui_when_backgrounded

For more information, see Configure the blur app option

Block rooted and jailbroken devices

To improve security on your mobile device, block the logging in of mobile apps when a rooted or jailbroken device is used. Use the following security property:

• glide.sg.allow_rooted_jailbroken_devices system

For more information, see Configure the status for rooted and jailbroken devices.

Penetration testing

ServiceNow engages a third party to perform penetration testing of a 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**

ServiceNow mobile apps do not specifically collect any user data.

Any user transactions or usage within an 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**

ServiceNow mobile apps communicate with a third party software for app crash reporting. No customer information is shared.

- iOS:
  - Uses Crashlytics for crash reporting.
- Android:
  - Uses Crashlytics for crash reporting.

**Incident reporting**

ServiceNow mobile app issues should be reported through the standard support channels. You can report incidents by contacting Customer Service and Support.

**Edge Encryption for ServiceNow mobile**

Users can view and edit data protected with Edge Encryption within their mobile device. The data appears in readable form on the mobile device but is encrypted in the database.

ServiceNow® Edge Encryption encrypts sensitive data on your company premises before sending it over the Internet to your instance (encrypted in flight). This encrypted data is then stored and protected within your database (encrypted at rest).

The diagram shows an example of how Edge Encryption operates within the mobile platform. A field is configured and protected with Edge Encryption. When the user enters a value in an encrypted field on the mobile device, it remains in a readable format. However, in the database, the value is displayed as an encrypted value.
For more information, see Edge Encryption.

**Mobile GovCommunityCloud (GCC) Compliance**

Learn about how ServiceNow mobile apps comply with security standards for the GovCommunityCloud environment.

**Note:**

The Mobile Onboarding app is being deprecated!

With the Now Platform Rome release in September 2021, we started phasing out support for the Mobile Onboarding. Customers may no longer activate it, and we are not offering enhancements or non-critical bug fixes. Mobile onboarding features are available in the Now Mobile app for HR Service Delivery."
ServiceNow GovCommunityCloud (GCC) compliance is designed for U.S. Federal, State, and local government customers. This environment is FedRAMP High and DoD Impact Level 4 authorized and compliant. Each ServiceNow mobile app (Now Mobile, Mobile Agent, and Mobile Onboarding) use FIPS 140-2 certified encryption modules and are GCC-compliant.

As part of this accreditation, ServiceNow mobile apps include the following:

**Encryption**
ServiceNow uses FIPS 140-2 validated encryption when connecting to GovCommunityCloud instances.

**Enforced security feature enablement**
Enforced device PIN or biometric enablement when connecting to GovCommunityCloud instances.

**Encryption for locally stored data**
Locally stored app data such as user preferences and offline data are encrypted.

**Blur feature**
The blur feature is automatically enabled when the app is in the background.

---

**iOS FIPS 140-2 Compliance**
- On iOS devices, ServiceNow mobile apps use Apple’s validated cryptographic modules. These modules are available on all devices using iOS 11 and up.
- To enforce iOS FIPS 140-2 encryption, the ServiceNow mobile apps require that a user’s device has a pass code enabled when connecting to a GCC instance.
- All locally stored mobile app data such as user preferences and offline data use FIPS 140-2 validated encryption when pass code enablement is confirmed.

For more information on Apple’s validated cryptographic modules, see [https://support.apple.com/en-us/HT202739](https://support.apple.com/en-us/HT202739)

**Android FIPS 140-2 Compliance**
- On Android devices, ServiceNow mobile apps are integrated with a third party SDK that uses a FIPS 140-2 validated certificate.
- With this SDK, Android versions of ServiceNow mobile apps are FIPS 140-2 complaint for data at rest. All locally stored app data such as user preferences and offline data use the same level of encryption.
• ServiceNow mobile apps also require that a device pass code is enabled when a user connects to a GCC instance.

**Note:** This feature requires Android version 7.0 Nougat and up.

For more information on the certificate used with the third party SDK, see https://csrc.nist.gov/projects/cryptographic-module-validation-program/Certificate/1747

**Mobile system properties related to GCC compliance**

**Enforcing FIPS 140-2 Encryption**

Use the glide.sg.device_encryption_enabled system property to enforce encryption and require that a device pass code is configured. This system property is automatically added and defaults to true for GCC instances.

For non-GCC instances, this property defaults to false. Enable this property on these instances to take advantage of encryption and device pass code enablement.

**Disabling offline mode**

On GCC instances, offline mode is disabled by default when the offline mode plugin is installed. To enable offline mode on a GCC instance, an administrator must create the glide.sg.offline.enabled system property on the [sys_properties] table, and set the value of this property to true.

For non-GCC commercial instances, offline mode is enabled by default when the offline mode plugin is installed. To disable offline mode on a non-GCC instance, an administrator must create the glide.sg.offline.enabled system property on the [sys_properties] table, and set the value of this property to false.

For more information on offline mode, see [Offline mode](#).

**Screen blur on background**

Use the glide.sg.blur_ui_when_backgrounded system property to blur the app screen when in background. This property was introduced in the Madrid release. Beginning in the Paris release, this property is automatically enabled with a value of true.

**Note:** This property is not overridden for existing customers who upgrade to the Paris release.

**FedRAMP**

The Federal Risk and Authorization Management Program (FedRAMP) creates a set of processes to ensure cloud security for the government. For more detail on this program, see https://www.fedramp.gov/.

**Enterprise mobility management (EMM)**

Use an EMM suite to distribute ServiceNow mobile apps or the ServiceNow Classic mobile app.
Enterprise mobility management (EMM)

Apply your corporate app protection policies to ServiceNow mobile apps by either using your EMM suite, or, with an embedded mobile application management (MAM) SDK for personal devices. ServiceNow only supports Intune and BlackBerry SDKs.

AppConfig

AppConfig is a standard approach for configuring mobile apps using key-value pairs. AppConfig was created by leading EMM providers like MobileIron, SAP, IBM, and VMWare. For more information on application configuration, please read your MDM product documentation. ServiceNow supports two app configurations:

- Pre-configure the default instance
- Change the default browser

For details on these configurations, see AppConfig for Mobile Apps.

Mobile application management (MAM) integration

Use Microsoft Intune or BlackBerry Dynamics to secure and protect sensitive information in mobile applications, even in cases where customers use their own mobile devices. For more detail on MAM integration, see Mobile application management (MAM) integration.

Unsupported MDM/MAM features

The following MDM and MAM features are not supported by ServiceNow.

- Application distribution
- App protection policies
- GA (general availability) binary files (ipa and apk) will not be distributed.
- App wrapping and/or resigning are not allowed.
- Troubleshooting mobile gateway or proxy rules.
- App Config is supported for on-prem MDM but creating plist file is not supported.

Mobile application management (MAM) integration

Use Microsoft Intune or BlackBerry Dynamics to secure and protect sensitive information in mobile applications, even in cases where customers use their own mobile devices.

Use Microsoft Intune or BlackBerry Dynamics mobile management tools to control, secure, and enforce policies for ServiceNow mobile apps. These tools provide a central point of control for securing your data on mobile apps, even in scenarios where you are not the owner of the mobile device.

These apps are available in the Apple App store, Google Play, and BlackBerry marketplace stores. Get started right away using these publicly available apps, or request custom branded versions of these apps with your unique company identity.

Available apps by version

<table>
<thead>
<tr>
<th>MAM Platform</th>
<th>Mobile Agent Availability</th>
<th>Now Mobile Availability</th>
<th>Mobile Onboarding Availability</th>
</tr>
</thead>
</table>
Note:
The Mobile Onboarding app is being deprecated!

With the Now Platform Rome release in September 2021, we started phasing out support for the Mobile Onboarding. Customers may no longer activate it, and we are not offering enhancements or non-critical bug fixes. Mobile onboarding features are available in the Now Mobile app for HR Service Delivery.

Microsoft Intune

Take advantage of the policy enforcement and management features of Microsoft Intune for your ServiceNow mobile apps.

ServiceNow provides versions of the Mobile Agent, Now Mobile, and Mobile Onboarding apps designed to integrate with your Microsoft Intune mobility management tools.

Using these apps, you can set and manage policies for each app using your organization's Intune administration portal. The following mobile application management features are supported:

- App protection without device enrollment
- Restrict copy/paste
- Attachment Control
- Remote wipe
- MSFT conditional access

BlackBerry Dynamics

Manage your ServiceNow mobile apps using your organization's BlackBerry Dynamics mobility management tools.

ServiceNow provides versions of the Mobile Agent, Now Mobile, and Mobile Onboarding apps designed to integrate with your BlackBerry Dynamics mobility management tools. BlackBerry versions of ServiceNow are available in the BlackBerry marketplace store.
Using these apps, you can set and manage policies for each app using the BlackBerry Dynamics SDK. The following mobile application management features are supported:

Your branded apps are integrated with the BlackBerry Dynamics SDK to help manage policy enforcement and employee access to your data. The following mobile application management features are supported:

• App protection without device enrollment
• Restrict copy/paste
• Attachment Control
• Remote wipe
• Dynamic VPN

Package and Bundle IDs

To enforce app policies your mobile application management software may require the bundle or package ID associated with the mobile apps. For information on these IDs see Bundle and Package IDs for ServiceNow mobile apps.

Bundle and Package IDs for ServiceNow mobile apps

To enforce app policies your mobile application management software may require the bundle or package ID associated with the mobile app. See the following reference for these IDs.

Mobile app bundle and package IDs for Microsoft Intune

<table>
<thead>
<tr>
<th>iOS bundle IDs</th>
<th>Bundle ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>App</td>
<td></td>
</tr>
<tr>
<td>ServiceNow® Agent - Intune</td>
<td>com.servicenow.intune.fulfiller</td>
</tr>
<tr>
<td>Now® Mobile - Intune</td>
<td>com.servicenow.intune.requestor</td>
</tr>
<tr>
<td>ServiceNow® Onboarding - Intune</td>
<td>com.servicenow.intune.onboarding</td>
</tr>
</tbody>
</table>
Android package IDs

<table>
<thead>
<tr>
<th>App</th>
<th>Package ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceNow® Agent - Intune</td>
<td>com.servicenow.fulfiller.mam.intune</td>
</tr>
<tr>
<td>Now® Mobile - Intune</td>
<td>com.servicenow.requestor.mam.intune</td>
</tr>
<tr>
<td>ServiceNow® Onboarding - Intune</td>
<td>com.servicenow.onboarding.mam.intune</td>
</tr>
</tbody>
</table>

Mobile app bundle and package IDs for BlackBerry Dynamics

iOS bundle IDs for BlackBerry Dynamics

<table>
<thead>
<tr>
<th>App</th>
<th>Bundle ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceNow® Agent - BlackBerry</td>
<td>com.servicenow.blackberry.fulfiller</td>
</tr>
<tr>
<td>Now® Mobile - BlackBerry</td>
<td>com.servicenow.blackberry.requestor</td>
</tr>
</tbody>
</table>

Android package IDs for BlackBerry Dynamics

<table>
<thead>
<tr>
<th>App</th>
<th>Package ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceNow® Agent - BlackBerry</td>
<td>com.servicenow.fulfiller.mam.blackberry</td>
</tr>
<tr>
<td>Now® Mobile - BlackBerry</td>
<td>com.servicenow.requestor.mam.blackberry</td>
</tr>
</tbody>
</table>

Control specific app usage

To support your organization’s authentication policies, admins can control which mobile apps can log in to ServiceNow instances.

To control which mobile apps can log in to ServiceNow instances, admins can choose either basic app allowance lists or advanced app allowance. The advanced allowance mode enables admins to add links to authorized apps. These two modes are described in the following sections.

**Important:** The basic and the advanced app allowance modes should not both be configured on the same instance. If you configure both modes on the same instance, the system always prioritizes the advanced mode configuration over the basic mode configuration.

Basic app allowance lists

By configuring a system property, admins can create a list of mobile apps that can connect to ServiceNow instances. If a user tries to connect to an instance with an app that isn’t on the list, the user receives an Unable to log in … message as shown in the following image.
Advanced app allowance with links to permitted apps

Using Scripted Extension Points, admins can create a list of apps that can log in to ServiceNow instances. Admins can change the Dismiss link text to route the user to the appropriate mobile app. Using the link, end users can connect to an instance with an authorized mobile app.

Configure a basic allowed mobile app list

As an admin, configure a system property that contains a list of mobile apps that can log in to ServiceNow instances.

Before you begin
Role required: admin
About this task
To support your organization’s authentication policies, you can configure a list of mobile apps that can log in to instances. When a mobile app that is not granted access attempts to log in to your instance, a message appears. This message informs users that their app cannot access the instance.

Procedure
1. Navigate to All > sys_properties.list.
2. Make sure you are in the global application scope.
   To check if you are in the global application scope, select the globe icon (🌍) in the banner. The application scope that you are in is displayed.
   If you aren’t in the global application scope, then select Application Scope > Global.
4. In the Name field, enter glide.sg.allowed_mobile_apps.
   Enter this name exactly with no extra characters. This field is also case-sensitive, so enter the name with all lower-case characters.
5. In the Application field, make sure the field is set to Global.
6. In the Type field, select string.
7. In the Value field, enter a comma-separated list of the iOS Bundle IDs and of the Android Package Names for the apps that can log in to your instance.
   The following example shows a configuration that only allows end users with the Intune Now Mobile app to log in to the instance. All other apps are restricted and unable to log in.
   ```
   com.servicenow.intune.requestor,com.servicenow.requestor.mam.intune
   ```
8. Select Submit.

Configure advanced app allowance with links to permitted apps
Configure a scripted extension point with a list of mobile apps that can log in to the ServiceNow instance. When a user attempts to log in with an unauthorized app, the admin can add a link that redirects the user to an authorized app.

Before you begin
Role required: admin

About this task
Only use the advanced app allowance configuration if you must configure a link that redirects end users to a mobile app that is authorized to connect to the instance. Otherwise, you should use the basic app allowance list configuration. For more information, see Configure a basic allowed mobile app list.

If both basic and advanced app allowance modes are configured on the same instance, then the system prioritizes the advanced app allowance configuration and always uses it. The advanced app allowance configuration can cause performance issues if the scripted extension point is not written to maximize system performance.

Procedure
1. Navigate to All > Scripted Extension Points.
2. Search for the global.CustomPreAuthProperties API Name and select it.
   The CustomPreAuthProperties record appears. Do not change any field except the Example field.
3. In the Example field, configure the two following properties:
• **allowed_mobile_apps**
  Add a comma-separated list of iOS Bundle IDs and Android Package Names. Use this list to add mobile apps that can connect to your instance. For example:

```javascript
::
customProperties['allowed_mobile_apps']='com.servicenow.fulfiller';
::
```

• **blocked_mobile_apps_redirect**
  Add a redirect URL for the button that appears when an end user attempts to log in with an unauthorized app. For example:

```javascript
::
::
```

For an example of a completed script, see Advanced app allowance example script.

4. To save the record, select **Update**.

5. Select **Create implementation**.

**Results**

A new script include is created and registered as an extension point instance.

**Advanced app allowance example script**

Users with the admin role can use the example JSON script to configure a scripted extension point. Admins can use these scripted extension points to limit which mobile apps can log in to ServiceNow instances. These scripts can also be used to specify a redirect link to authorized mobile apps.

**Example script**

The following example script blocks all apps except the Mobile Agent app and the Now Mobile app.

All other apps are restricted. This script also uses the blocked_mobile_apps_redirect property. When an end user attempts to log in with an unauthorized app, an error message appears with a redirect button. That button redirects the end user to log in to a mobile app that is authorized to connect to the ServiceNow instance.

```javascript
var CustomPreAuthProperties = Class.create();
CustomPreAuthProperties.prototype = {
  initialize: function() {},

  /** *
   * Returns a JSON object keyed by the custom property names.
   */
  getProperties: function(input) {
    var customProperties = {};
    if (input.clientType == "agent") {
      customProperties['allowed_mobile_apps'] = 'com.servicenow.fulfiller';
      if (input.deviceType == 'android') {
      }
    } else {
      // Other cases...
    }
  }

  // Additional methods...
};
```

© 2022 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.
This sample script uses the following JSON objects:

- `input.clientType` which determines the mobile app type.
- `input.deviceType` which determines the operating system type.

### AppConfig for Mobile Apps

Use AppConfig to define a default instance and browser for your managed mobile apps.

#### Pre-configure the default instance

When you distribute ServiceNow mobile apps through an EMM suite or an embedded MAM SDK app, you can use AppConfig to pre-configure the default instance URL.

**Key**  
**Value**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNDefaultInstanceURL</td>
<td>URL for your instance (ex: <a href="https://instancename.service-now.com">https://instancename.service-now.com</a>)</td>
</tr>
</tbody>
</table>

#### Prevent end users from creating connections to new instances

You can use the following AppConfig key/value pair to prevent end users from creating connections to new instances for managed mobile apps. When the following key is set to true, end users do not see the plus sign (+) on the instance list page of their mobile app. Without the plus sign, end users cannot add instances that they are not authorized to add.

**Key**  
**Value**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNBlockInstanceCreate</td>
<td>&quot;true&quot;</td>
</tr>
</tbody>
</table>

#### Change the default browser

Because ServiceNow uses AppAuth for authentication, the mobile apps use the default browser on the mobile device. iOS uses the default in-app browser, while Android uses the default operating system browser. You may have browser security requirement where their
app protection policy only allows their MDM managed browser or a specific browser. A common use case is the support for per-app VPN.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Key</th>
<th>Value</th>
<th>Browser</th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS</td>
<td>SNAuthenticationBrowseriOS</td>
<td>Safari</td>
<td>Apple Safari</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Chrome</td>
<td>Google Chrome</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Firefox</td>
<td>Mozilla Firefox</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Edge</td>
<td>Microsoft</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WorkspaceONE</td>
<td>AirWatch VMWare</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Workspace ONE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WebAtWork</td>
<td>MobileIron</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Web@Work</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BlackBerry Access</td>
<td>BlackBerry Access</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MaaS360</td>
<td>IBM MaaS360</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Citrix</td>
<td>Citrix Secure</td>
</tr>
<tr>
<td>Android</td>
<td>SNAuthenticationBrowserAndroid</td>
<td>Chrome</td>
<td>Google Chrome</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Firefox</td>
<td>Mozilla Firefox</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Samsung</td>
<td>Samsung Internet Browser</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WorkspaceONE</td>
<td>AirWatch VMWare</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Workspace ONE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Edge</td>
<td>Microsoft</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BlackBerry Access</td>
<td>BlackBerry Access</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MaaS360</td>
<td>IBM MaaS360</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Citrix</td>
<td>Citrix Secure</td>
</tr>
</tbody>
</table>

**Note:**
- Keys are case sensitive.
- AppConfig key-values are still supported for non-managed devices if the user installs a ServiceNow app with a MAM SDK.

**Supported mobile device management vendors**

Learn how to configure mobile device management software from supported vendors to distribute ServiceNow mobile apps.

**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.

Before you begin
Role required: admin

Procedure
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.

Application 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>https://&lt;instance name&gt;.service-now.com/</td>
</tr>
</tbody>
</table>
13. Click Add.
14. Click Save & Publish.
15. Click Publish.

Results
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 app for Android to AirWatch
Configure the ServiceNow app for Android for AirWatch distribution.

Before you begin
Role required: admin

Procedure
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>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Server URL</td>
</tr>
<tr>
<td>Description</td>
<td>The full ServiceNow instance URL</td>
</tr>
<tr>
<td>Value</td>
<td>https://&lt;instance name&gt;.service-now.com/</td>
</tr>
</tbody>
</table>

13. Click Add.

14. Click Save & Publish.

15. Click Publish.

© 2022 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.
Results
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 (Unified Endpoint Management) as a mobile device management system to distribute and manage the ServiceNow mobile app on user devices.

Get the BlackBerry UEM registered versions of ServiceNow mobile apps

| ServiceNow has BlackBerry integrated versions of the Now Mobile and Mobile Agent apps. These apps integrate with the BlackBerry UEM portal. These apps are available in the BlackBerry Marketplace at https://marketplace.blackberry.com/apps |
| Note: ServiceNow does not support the legacy Blackberry Good Control management system. |

Managing ServiceNow mobile apps with BlackBerry UEM
Learn how to manage ServiceNow mobile applications in the BlackBerry UEM (Unified Endpoint Management).

Before you begin
Role required: admin

Procedure
1. Sign into the BlackBerry UEM.
2. In the menu bar to the left, click Apps.
You see a list of available apps tied to your BlackBerry UEM.

3. In the search bar above the list of apps, type ServiceNow.

4. Click the app you want to register.
5. When the app opens, click on the blue text labeled **App Config With Default Values** to create the app configuration for this app.

6. In the app configuration form, fill in the fields as needed.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Instance URL</td>
<td>Your instance URL. For example, <a href="https://INSTANCENAME.service-now.com">https://INSTANCENAME.service-now.com</a>.</td>
</tr>
<tr>
<td>iOS Authentication Browser</td>
<td>The browser your app will use for authentication on iOS devices.</td>
</tr>
<tr>
<td>Android Authentication Browser</td>
<td>The browser your app will use for authentication on Android devices.</td>
</tr>
</tbody>
</table>

7. Click **Save** to save the app configuration.
8. Click **Save** to save the app.

**Note:** If using the instructions above to setup the BlackBerry Access browser as your authentication browser, you must also have an app configuration setup for the BlackBerry Access app. This is necessary to successfully authenticate to ServiceNow integrated apps.

**Configure BlackBerry Access for mobile apps**
Configure the BlackBerry Access browser to authenticate ServiceNow mobile apps.

**Before you begin**
Role required: admin

**About this task**
The following steps are only necessary if you are using the BlackBerry Access browser as the authentication browser for your ServiceNow mobile apps. The BlackBerry Access browser must be configured in order to authenticate with your ServiceNow mobile apps.

**Procedure**
1. Sign into the BlackBerry UEM.
2. In the menu bar to the left, click **Apps**.
You see a list of available apps tied to your BlackBerry UEM.

3. In the search bar above the list of apps, type BlackBerry Access.

4. Click **BlackBerry Access** to open the app.

5. When the app opens, click on the blue text labeled **App Config With Default Values** to create the app configuration for this app.
6. Enable **Allow external apps** to open HTTP/HTTPS URLs through BlackBerry Access.

7. Enable **Enable 3rd Part Applications**.

8. Add these values: **snappauth,snempappauth**

   - **Note:** Do not use any spaces in the values above.

   - **Note:** The **snappauth** is required for the Mobile Agent app. The **snempappauth** value is required for the Now Mobile app.

9. Click **Save**.

**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.

**Before you begin**

Role required: admin

**Procedure**

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.
      The App Details fields are pre-populated with information related to the app.
   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.
   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 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.

What to do next
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.
About this task
Configuring the app for a default instance is only available for iOS devices.

Procedure
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.

```xml
<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.

Related information
- Citrix: Add apps
- Citrix: Android for Work
- Citrix: App configuration policy
- Citrix: VPN policy

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.

Before you begin
Role required: admin

Procedure
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.

Results
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 app for Android to the IBM MaaS360 portal

Configure the ServiceNow app for Android for IBM MaaS360 distribution.

Before you begin
Role required: admin
Procedure
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.

Results
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 an Intune integrated app to Microsoft Azure
Configure your ServiceNow app for distribution with Microsoft Intune.

Before you begin
Role required: admin

Procedure
1. Sign in to the Microsoft Azure portal.
2. In the search bar, enter Client Apps.
3. In the navigator menu on the left side of the screen, navigate to Manage > Apps.
4. Click Add.

5. Select the App type.

Note: Both iOS and Android store apps will go through the same process. When adding an iOS app, Azure offers a shortcut for you to find the app on the Apple App Store seen here:

If you are adding an Android store app, there is no shortcut to add in the apps. Android apps are found in the following locations.
• Mobile Agent Intune can be found [here](#).
• Now Mobile Intune can be found [here](#).
• Mobile Onboarding Intune can be found [here](#).

**Note:**
The Mobile Onboarding app is being deprecated!
With the Now Platform Rome release in September 2021, we started phasing out support for the Mobile Onboarding. Customers may no longer activate it, and we are not offering enhancements or non-critical bug fixes. Mobile onboarding features are available in the Now Mobile app for HR Service Delivery."

6. Click **Next**.
7. Add the specific group or all users who will be using this app.

8. Click **Next**.
9. Review the information you entered. When you are finished, click **Create**.

**Add an Intune integrated apps to a protection policy in Microsoft Azure**
Learn how to add your ServiceNow mobile apps to your existing Microsoft Azure protection policies.

**Before you begin**
Role required: admin
The following steps describe how to add your Intune integrated apps to an existing protection policy. For information on creating protection policies, refer to your Microsoft Azure documentation.

**Procedure**
1. Sign in to the Microsoft Azure portal.
2. In the search bar, enter Client Apps.

3. In the navigator menu on the left side of the screen, navigate to Manage > App protection policies.
4. Open the app protection policy where you want to add your apps.

5. In the left navigator, open Manage > Properties.
6. To the right of the **Apps** header click **Edit** to add an app.
7. In the Edit policy form, click **Select public apps**.

8. Add the selected apps, then click **Select**.

Configure the default authentication browser for ServiceNow mobile apps in Microsoft Azure

Learn how to configure the default authentication browser for your ServiceNow mobile apps managed by Microsoft Azure.

**Before you begin**
Role required: admin
The following steps describe how to add your Intune integrated apps to an existing configuration policy. For information on creating configuration policies, refer to your Microsoft Azure documentation.

**Procedure**
1. Sign in to the Microsoft Azure portal.
2. In the search bar, enter Client Apps.
3. In the navigator menu on the left side of the screen, navigate to Manage > App configuration policies.
4. Open the configuration policy where you want to add your ServiceNow mobile apps.
5. To the right of the Settings header, click **Edit**.
6. In the **General configuration settings** section, add the following **Name** and **Value** entries:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNDefaultInstanceURL</td>
<td><a href="https://INSTANCENAME.service-now.com/">https://INSTANCENAME.service-now.com/</a></td>
</tr>
<tr>
<td><strong>Note:</strong> Replace INSTANCENAME with the name of your instance.</td>
<td></td>
</tr>
<tr>
<td>SNAuthenticationBrowseriOS</td>
<td>Enter the value for the browser to use for apps on iOS mobile devices. For a list of values, see AppConfig for Mobile Apps</td>
</tr>
<tr>
<td>SNAuthenticationBrowserAndroid</td>
<td>Enter the value for the browser to use for apps on Android mobile devices. For a list of values, see AppConfig for Mobile Apps</td>
</tr>
</tbody>
</table>

7. Click **Review + save**
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.

Before you begin
Role required: admin

Procedure
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.

What to do next
After the instance is configured for the mobile app, configure the app settings to distribute the app to mobile devices in the scope.

Related information
Configure app 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 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.

**Before you begin**

Role required: admin

**Procedure**

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 a distribution level 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 key and value pairing.

<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. Set a distribution level 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.
Results
The ServiceNow app for iOS 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.

Before you begin
Role required: admin

Procedure
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. Set a distribution level 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. Add the name and configuration value for the default instance configuration.

<table>
<thead>
<tr>
<th>Configuration Setup fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Configuration value</td>
</tr>
</tbody>
</table>

12. Set a distribution level 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.
13. Click **Next**.

14. Click **Done**.

**Results**

The ServiceNow app for Android from the Google Play store 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, 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.
Before you begin
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.

Procedure
1. Navigate to the Google Play store.
2. Search for the ServiceNow mobile app.
3. Click Approve.

Tested devices and supported versions for ServiceNow mobile apps
Use the following information to ensure that your mobile devices and operating systems are ready to use ServiceNow mobile apps.

Operating system version support
ServiceNow supports latest two versions of Apple iOS, and the latest four versions of Google Android. ServiceNow reserves the right to further reevaluate support based on OS capabilities and the product development roadmap.

Supported operating system and instance versions
Use this table to find the minimum iOS and Android versions necessary to use ServiceNow mobile apps.

<table>
<thead>
<tr>
<th>ServiceNow mobile app version</th>
<th>Minimum iOS version</th>
<th>Minimum Android version</th>
<th>Maximum Instance version</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.x</td>
<td>iOS 11</td>
<td>5.0 Lollipop</td>
<td>Madrid</td>
</tr>
<tr>
<td>7.x</td>
<td>iOS 11</td>
<td>5.0 Lollipop</td>
<td>Madrid</td>
</tr>
<tr>
<td>8.x</td>
<td>iOS 11</td>
<td>5.0 Lollipop</td>
<td>New York</td>
</tr>
<tr>
<td>9.x</td>
<td>iOS 12</td>
<td>5.0 Lollipop</td>
<td>Orlando</td>
</tr>
<tr>
<td>10.x</td>
<td>iOS 12</td>
<td>6.0 Marshmallow</td>
<td>Paris</td>
</tr>
<tr>
<td>11.x</td>
<td>iOS 13</td>
<td>7.0 Nougat</td>
<td>Quebec</td>
</tr>
</tbody>
</table>
Supported operating system versions by app version (continued)

<table>
<thead>
<tr>
<th>ServiceNow mobile app version</th>
<th>Minimum iOS version</th>
<th>Minimum Android version</th>
<th>Maximum Instance version</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.x - 12.4</td>
<td>iOS13</td>
<td>7.0 Nougat</td>
<td>Rome</td>
</tr>
<tr>
<td>12.5 +</td>
<td>iOS13</td>
<td>8.0 Oreo</td>
<td>Rome</td>
</tr>
<tr>
<td>13.0</td>
<td>iOS 13</td>
<td>8.0 Oreo</td>
<td>San Diego</td>
</tr>
<tr>
<td>13.1</td>
<td>iOS 14</td>
<td>8.0 Oreo</td>
<td>San Diego</td>
</tr>
</tbody>
</table>

**Note:**

The latest mobile apps are backward compatible and can be connected to any instance version. However, the mobile apps are not forward compatible. For example, the 8.x client should only be used in New York and earlier instances. Avoid experience issues or crashes by following these maximum instance version guidelines.

**Mobile tested devices**

ServiceNow® mobile apps have been officially tested on the following devices, although additional devices might work as well. ServiceNow® mobile apps require the Madrid release of ServiceNow or later.

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Tested devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple® iPhone®</td>
<td>• iPhone 7&lt;br&gt;• iPhone 8 and 8 Plus&lt;br&gt;• iPhone XR&lt;br&gt;• iPhone XS&lt;br&gt;• iPhone XS Max&lt;br&gt;• iPhone 11&lt;br&gt;• iPhone 11 Pro&lt;br&gt;• iPhone 11 Pro Max&lt;br&gt;• iPhone 12 Pro&lt;br&gt;• iPhone 12 Pro Max</td>
</tr>
<tr>
<td>Apple® iPad®</td>
<td>• iPad&lt;br&gt;• iPad mini (5th gen)&lt;br&gt;• iPad Pro (2nd gen)&lt;br&gt;• iPad Pro (4nd gen)&lt;br&gt;• iPad Mini 5&lt;br&gt;• iPad Air 10.5 (3rd Generation)&lt;br&gt;• iPad 7.9” (5th Generation)&lt;br&gt;• iPad 9.7”&lt;br&gt;• iPad 10.2”</td>
</tr>
<tr>
<td>Operating system</td>
<td>Tested devices</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------</td>
</tr>
<tr>
<td></td>
<td>• iPad Pro 11&quot; (2nd Generation)</td>
</tr>
<tr>
<td></td>
<td>• iPad Pro 12.9&quot; (4th Generation)</td>
</tr>
<tr>
<td>Android™</td>
<td>• Asus ZenPad 3S 10</td>
</tr>
<tr>
<td></td>
<td>• Huawei P20</td>
</tr>
<tr>
<td></td>
<td>• OnePlus 8 Pro</td>
</tr>
<tr>
<td></td>
<td>• Samsung S8 and S8 Edge</td>
</tr>
<tr>
<td></td>
<td>• Samsung S9</td>
</tr>
<tr>
<td></td>
<td>• Samsung Galaxy Note10</td>
</tr>
<tr>
<td></td>
<td>• Samsung S10 and S10+</td>
</tr>
<tr>
<td></td>
<td>• Samsung S20 and S20+</td>
</tr>
<tr>
<td></td>
<td>• Samsung S20 Ultra 5G</td>
</tr>
<tr>
<td></td>
<td>• Samsung Galaxy Tab A</td>
</tr>
<tr>
<td></td>
<td>• Samsung Galaxy Tab 3</td>
</tr>
<tr>
<td></td>
<td>• Samsung Galaxy Tab S5e</td>
</tr>
<tr>
<td></td>
<td>• Samsung Galaxy Z Flip</td>
</tr>
<tr>
<td></td>
<td>• Google Pixel 2 and 2 XL</td>
</tr>
<tr>
<td></td>
<td>• Google Pixel 3 and 3a</td>
</tr>
<tr>
<td></td>
<td>• Google Pixel 3 XL and 3a XL</td>
</tr>
<tr>
<td></td>
<td>• Google Pixel 4</td>
</tr>
<tr>
<td></td>
<td>• Google Pixel 4 XL</td>
</tr>
<tr>
<td></td>
<td>• Google Pixel 5</td>
</tr>
<tr>
<td></td>
<td>• Xiaomi Mi 8</td>
</tr>
<tr>
<td></td>
<td>• Zebra TC51</td>
</tr>
</tbody>
</table>

Android tablet is not officially tested.

**Using the mobile apps**

Learn how use ServiceNow mobile apps to get your work done from anywhere using your mobile device.

**Log in to an instance with a mobile app**

Learn how to access an instance using a ServiceNow mobile app.

**Before you begin**

Role required: none

**About this task**

If you downloaded the mobile app from an enterprise mobility management (EMM) suite, enter your user credentials as prompted. For more detail on EMM, see Enterprise mobility management (EMM)

If you did not download the mobile app from an EMM suite, follow the subsequent steps.
Procedure

1. Tap the icon for one the ServiceNow mobile apps ( NOW ) on your device
2. In the instance address field, enter the instance address in one of two ways:
Enter the instance address or scan a QR code.

- example.service-now.com

Nickname (Optional)

Save and log in

<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.</td>
</tr>
<tr>
<td>Note:</td>
<td>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 (QR code), and then scan the QR code that was provided by your administrator. This QR code will automatically fill in the name of the instance.</td>
</tr>
</tbody>
</table>
Note: If you have already logged into another instance using the app, you can the plus button (➕) to open a different instance.

3. Tap Login.

4. In iOS versions of the apps, a window appears to remind you that you are connecting your mobile device to the server. Tap Continue.

5. In iOS versions of the apps, you see the prompt to connect to the server. Tap Allow.

Note: The preceding two steps are a requirement of the iOS operating system, are not configurable.

Related information
Mobile authentication

Mobile app structure
Learn how applets, screens, and settings are organized within ServiceNow® mobile apps to understand how to find the tools and data your looking for.
Components of a ServiceNow app

App Icons

Each ServiceNow app has its own icon on your mobile device. You can configure your apps to show a badge indicating the number of unread notifications you have in the app.

Note: If your company has provided you with branded ServiceNow mobile apps, your icons may not match the image shown here.
**Navigation bar**

Each ServiceNow app displays a navigation bar along the bottom of the screen. This bar displays icons called **tabs**, which you can use to access settings, notifications and commonly used applets or application launch pages.

Your administrator can choose which tabs appear in the navigation bar, and the order in which they appear. If a navigation bar has six or more icons, a **More** tab appears. Tap the **More** tab to see the additional tabs in a list format.

For more detail on navigation bars, see **Navigation bar**.

---

**Applet Launchers**

Applet launchers serve as landing pages or home pages. Using an applet launcher, you can access applets in various formats, as well as search, do quick actions, and find user information.

**Header**

The header of the applet launcher defines how the title of the screen appears and what information is shown in the header. For detail on configuring headers, see **Configure a launcher screen header**.

**Search**

Use global search to quickly search through defined search sources configured on your instance. For more details, see **Global search for mobile**.

**Sections**

Sections appear as containers within applet launchers. Within sections, you can see applets, charts, record information, and more. For more detail on these sections, see **Launcher screen UI sections**.

**Quick Actions**

Quick actions are a way to access commonly used functions, such as transferring.
records or using a template in the mobile app. For more detail on quick actions, see Quick actions.

For more information on applet launchers configuration, see Launcher screens.

## Settings

<table>
<thead>
<tr>
<th>3:26</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; More</td>
<td></td>
</tr>
</tbody>
</table>

### Header
- The header displays the name of the logged in user and last login time and date.

### Instance
- The ServiceNow instance your app is logged into.

### Version
- The version of the mobile app on your mobile device.

### Security
- Tap to display a configuration page for your security options.

### Privacy Policy
- Tap to view the apps privacy policy.

### Legal
- Tap to view the apps legal disclosures

### Log out
- Tap to log out of the connected ServiceNow instance.

### Analytics
- For more detail about analytics settings, see Analytics settings for mobile
Notifications

The notifications screen is accessed from the Notifications tab on the navigation bar. This screen displays the push notifications sent to you. Tap any notification to view its contents.

You can enable or disable all notifications on your mobile device from this tab. For more information on notifications, see Mobile push notifications.
Applets are collections of screens within your mobile apps, designed for a specific task. Each applet provides one or more screens, which you see when you select an applet. The template may include other optional screens that you can access for additional information. All screens display information that is based on records in a data item.

**Segments**

Applets can contain one or more segments, which display information from a specific data item. If an applet has more than one segment, you will be able to switch between segments using a tabbed interface.

**Embedded screens**

Within a form applet, you might see one or many embedded screens. An embedded screen could be a details screen, an activity stream screen, a related list screen, or an embedded list screen.

**Functions**

Applets can be associated with one or more functions. Access these functions through the top menu or swipe actions. Use functions to perform tasks relating to the current record, or navigate to another applets, applet launchers, or external resources.

For more information on applets and how to configure them, see [Mobile screens](#).

**Related information**

- Mobile hierarchy
- Navigation bar
- Launcher screens
- [Mobile screens](#)

**Mobile search**

Learn how to use global search to find information within search sources configured for your mobile apps.

ServiceNow apps have several configurable ways to search to help you find what you need, to get work done on your mobile device.

**Text search**

Use search in the applet launcher header to quickly find information within the defined search sources while on your mobile device.

Your administrator can configure global search within an applet launcher header and define a set of tables that can be searched through. For details on how an administrator can configure global search, see [Enable global search in your applet launcher](#).
Voice search

If enabled by your administrator, you can tap to talk on the microphone icon ( микрофон ) to the right of the search bar. You can then speak to your device and your voice is captured as search text. For details on how an administrator can configure voice search, see Enable voice search.
Photo search

If enabled by your administrator, you can tap on the camera icon (📸) to the right of the search bar to perform image-based searches. For details on how an administrator can configure photo search, see Mobile photo search.

Mobile text search

Find items, articles, and people by entering text in the search area.

Use search in the applet launcher header to quickly find information within the defined search sources while on your mobile device.

Your administrator can configure a search area within applet launcher headers and define a set of tables that can be searched through. There are two search engine types which your administrator can configure, Zing text indexing and search engine and AI Search. Zing search is available throughout all mobile platforms, whereas AI Search is available only on Now Mobile.

The appearance and capabilities of the Zing and AI Search results differ. You may notice this difference as admins can configure the different search engines on different applet launchers, throughout a single instance.
Both search engines provide search suggestions as you type. They also include navigation tab filters which streamline the results displayed to help you find your specific answer. Zing search engine is available for all mobile platforms and utilizes additional data sources not associated with AI Search, for example cases or tasks.

AI Search has the following additional capabilities and behaviors to assist you in your search.

- **Recent and popular searches** - When tapping the search bar, suggested search options display according to your personal search history and queries by all users on the instance.
- **Auto-correct typos** - Misspelt words are automatically corrected.
- **Synonym handling** - The search engine is able to look for similar words entered and produce results accordingly. For example, if you enter “holiday” the AI Search also searches for “vacation”.
- **Genius cards and action items** - Genius result cards display the most relevant answers in an attractive visual layout. They can also contain buttons for you to perform an action.

**Mobile voice search**

Find items, articles, and people using native speech recognition from an app on their mobile device.
Using voice search

Take advantage of the speech recognition feature of your mobile device. Use voice search by tapping the microphone icon (.uml-source:image:media-127.png) that appears on the right side of a search bar. Use your device's voice recognition interface to record your search query, which is then converted to text and entered as the search query in your app.

The first time you use the voice search feature, your mobile device may prompt you to allow your ServiceNow app to record audio.

Note: Your administrator must enable this feature. For details on enabling voice search on your instance, see Enable voice search.

The first time you use the voice search feature, your mobile device may prompt you to allow your ServiceNow app to record audio.

If you deny this permission, you cannot use the voice search feature. The microphone icon changes to a disabled microphone icon (.uml-source:image:media-127.png).

Third-party data usage

Depending on your mobile device's operating system, the voice recordings you create with the voice search feature are sent to Google or Apple to be processed into a text query. ServiceNow does not have control of the recording once it has been sent.

Mobile photo search

Configure photo search to give your users the ability to perform image-based searches using the objects around them.
Photo searches in your mobile apps

When photo search is configured on your instance, a photo icon appears in your applet launcher search bar. Your users can tap this icon to use the camera on their mobile device to take a picture. The picture is identified using the Google Vision API, which returns one or more results. Users can select a result, which is used as their search query.

Third-party data usage

Images you take using the photo search feature are sent to Google for identification using Google Vision API. ServiceNow does not have control of the image once it has been sent. For details on how Google Vision API handles your image, see https://cloud.google.com/vision/docs/data-usage.

Accessing and posting to the mobile activity stream

Use the activity stream to submit and list work notes, files, and images to a record within your mobile app.

Before you begin
Role required: admin

About this task
An activity stream enables you to communicate with other users about the work done on a record.
Procedure

1. Access the activity stream segment for a record by opening a record and tapping the Activity Stream tab.

   *Note:* The default tab name is Activity Stream, but your system administrator might have changed it to a name like Updates, History, or Notes.

2. Tap an icon displayed within the ribbon in the record title bar to post items to the activity stream.
### Activity stream item

<table>
<thead>
<tr>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Tap the <strong>Add comment</strong> icon.</td>
</tr>
<tr>
<td>b. On the Compose screen, enter your text in the work notes area.</td>
</tr>
<tr>
<td>c. Enable the <strong>Share Comments</strong> button if you want all users on the instance to see your comment.</td>
</tr>
<tr>
<td>d. Tap <strong>Post</strong> to save your message in the activity stream.</td>
</tr>
</tbody>
</table>

### Text

<table>
<thead>
<tr>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Tap the file icon ( ).</td>
</tr>
<tr>
<td>b. Browse to the file you want to upload.</td>
</tr>
<tr>
<td>c. Tap on the file. The file automatically uploads to the activity stream.</td>
</tr>
</tbody>
</table>

### File

<table>
<thead>
<tr>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Tap the image icon ( ).</td>
</tr>
<tr>
<td>b. Browse to the image or video.</td>
</tr>
<tr>
<td>c. Tap on the image or video. The image or video automatically uploads to the activity stream.</td>
</tr>
</tbody>
</table>

### Image

<table>
<thead>
<tr>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Tap the camera icon ( ).</td>
</tr>
<tr>
<td>b. Select to either take a photo or record a video and then use your mobile device to perform one of those actions.</td>
</tr>
<tr>
<td>c. Save the photo or video. The photo or video automatically uploads to the activity stream.</td>
</tr>
</tbody>
</table>

### Photo or video

### Filtering and sorting results in mobile

Use mobile filter capabilities within list and map screens to help you search for specific data and streamline your results. Sorting lets you define how to view the filtered results.

### Before you begin

Role required: user

### Procedure

1. Tap the filter icon ( ) to filter your records.
2. Apply filtering and sorting options to your list by performing the following actions.
Select the category or categories you want to filter. Your administrator decides which categories are available.

**Note:** If only one category is defined, the instance does not present the option to select categories.

To see the filter conditions and sorting options that are common between all the displayed categories, tap All.

Select **Sort by** to display a list of options by which you sort your filtered results.

Select how you want to view your filtered results. Your administrator sets the default value. After you make your selection:

- Tap the back button (←) to return to the filtering options.
- Tap **Apply** to apply your filtering and sorting selections.
For date fields, tap the date field to display a calendar interface. The following actions are available:

- Swipe to the left or right to view previous or future months.
- To select or de-select a date, tap the day.
- To search a date range, tap a day, then tap a second day. The filter searches for dates matching that range.

For fields where you must select a value rather than enter a value, the available options display as check boxes. Tap one or more options to select or de-select, and then tap **Apply**.
Select **Apply**, after you define your filtering and sorting options.

### 3. Optional: Bookmark your filter and sorting selections for easy access. See, Mobile saved views.

**Mobile saved views**

Use saved views to gain quick access to frequently used screens and views.

Saved views enable you to save any screen or web page within your mobile app.

You can bookmark any screen on the mobile platform, including the following items:

- List
- Maps
- Forms
- Charts
- Calendars
- Embedded web pages (Cabrillo JS)

For lists and maps, you can save values, for example, a list of all critical incidents or a specific map location.

**Note:**

If configured by your administrator, you can access your saved views while working in offline mode. However, you cannot save, rename, or remove views in offline mode.
Bookmarking and using mobile saved views

Create bookmarks for mobile views like forms, maps, and charts. Decide how you want to display these saved views, and deselect them when you no longer need them.

Before you begin
Role required: user

Procedure

1. Tap the saved view icon (✓) in the screen header.

2. In the Save item name field, either accept the default name for the saved view or enter a name of your choice.

   Note: You can rename the saved view later by swiping left on the item in the Saved Items list, tapping Rename, entering a new name, and confirming the change.

3. Tap Save item name.

   The saved view icon changes to a filled status (✓) indicating that the page is bookmarked and available in the Saved items list.

4. Tap the Saved icon (Saved) in the navigation bar to access all your saved views from the Saved items list.

5. Deselect a saved view either from a bookmarked screen or from the Saved items list.

<table>
<thead>
<tr>
<th>Option</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>From a bookmarked screen</td>
<td>a. Locate the bookmarked screen you want to deselect.</td>
</tr>
<tr>
<td></td>
<td>b. Tap the saved view icon (✓).</td>
</tr>
<tr>
<td></td>
<td>The icon's appearance changes to an unfilled state (✓), indicating that</td>
</tr>
<tr>
<td></td>
<td>the screen is no longer saved.</td>
</tr>
<tr>
<td>From the Saved items list</td>
<td>a. Tap the Saved icon (Saved) in the navigation bar.</td>
</tr>
<tr>
<td></td>
<td>b. Swipe to the left on the saved item.</td>
</tr>
<tr>
<td>Option</td>
<td>Procedure</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>c. Tap Remove</td>
<td><img src="image" alt="Saved Items" /></td>
</tr>
</tbody>
</table>

The saved view is no longer listed in the Saved items list and the page is no longer bookmarked.

6. **Optional**: Arrange how you want your saved items listed.

a. Tap the Saved icon (Saved) in the navigation bar.

b. Tap on the arrow next to the heading above your listed saved views.
c. From the saved items menu, select how to display your saved views, for example, alphabetically or most recent to least recent.

**Mobile platform dashboards**

Use a mobile dashboard to display data in graphical format. Adding reports and Performance Analytics widgets helps users more easily identify trends and turning points through indicator scores and visual representation.

A mobile dashboard is an applet launcher that enables you to see preconfigured reporting and analytical views via trend lines and score visualizations. Tap chart UI sections in your applet launcher to open reports and analytical information displaying the status of instance data. For example, you can review the number of incidents opened per week or the average resolution time of projects.

For more information about adding chart UI sections to your applet launcher, see Configure chart UI sections.

For more information about accessibility features supported in charts, see Change mobile analytics views for accessibility.

**Mobile dashboard applet launcher showing Performance Analytics and reporting data**

Click the names of the Performance Analytics and Reporting charts listed in the sections below for general, non-mobile specific information.

**Performance Analytics charts**

Performance Analytics charts display views of key performance indicators over time, for example, the percentage mean time to resolve incidents.

The Performance Analytics chart available on the mobile platform is latest score.

**Reporting charts**

Reporting charts display views of current state instance data.

Four types of Reporting charts are available on the mobile platform: time series trend-by charts, score reporting, bar and donut. These charts display information such as the number of high priority work orders or the number of pending approvals over a defined period.

The available score report is single score.

The available bar report is vertical bar.
The available pie and donut report is donut.

The available time series trend-by charts are:

- Area
- Line
- Spline
- Step line

For more information

For general information on the benefit and use of Performance Analytics and Reporting concepts (not specifically for mobile), see the following documentation topics:

Understanding Performance Analytics

Getting started with reports

Performance Analytics compared to Reporting

Displaying Performance Analytics charts for mobile dashboard

Use Performance Analytics charts to help identify trends and turning points through indicator scores and visual representation. These charts are used when reporting on KPIs and metrics. The mobile platform supports the Performance Analytics latest score widget, which shows a single indicator score and the change in that score compared to a previous period. For example, you can configure the widget to show the number and percentage difference of high priority incidents over a time period.

Performance Analytics chart

Score - latest score widget

Latest score widgets show a single indicator score and the change in that score compared to a previous period. The latest score widget in the applet launcher.
displays the single score value, the Performance Analytics calculation, the time of the change, and the compare-to date.

Tap the widget to display a detailed chart of the Performance Analytics data. Alternatively, you can navigate to a chart screen via a navigation button. To magnify or contract the chart view, spread or pinch your fingers on the screen. A small sparkline chart at the bottom of the screen highlights the magnified section of the chart. Tap the zoom-out button (✓) to return the screen to its original magnification.

>Note: Unlike on the web-based UI, you cannot tap on an x axis for details or records on a specific area of a report.

Available Performance Analytics features

Mobile dashboards support the following Performance Analytics features from the web-based UI:

- Data configuration: Direction property; target and gap calculations in chart screen
- Style configuration: Change color

Unavailable Performance Analytics features

The following Performance Analytics features from the web-based UI are not available in mobile dashboards:

- Date configuration: Value when nil field; "compare to" date score
- Style configuration: Templates, score color

For more information

For general information about Performance Analytics widgets and when to use them, see Understanding Performance Analytics.
For instructions about enabling charts to be displayed on mobile apps, see Configure chart UI sections.

Displaying reporting charts for mobile dashboard

Use Reporting charts to help identify trends and turning points through indicator scores and visual representation. These charts display the current state of instance data.

The mobile platform supports the following reporting charts.

Time series reports

Time series reports show you data points over a period of time, for example, the number of cases resolved within the first response. For more information, see Time series reports for mobile.
Single score reports

Single score reports display a single key value, for example, the number of critical incidents opened by the QA team. For more information, see Single score reports for mobile.
**Donut reports**

Donut reports show how one grouping relates to the total amount, for example, the number of tasks completed by a team compared with the status of other task definitions. For more information, see Donut reports for mobile.
Bar reports

Bar reports compare two or more values. Use a stacked bar report to show information in segments that are proportional to the values they represent. A bar report, for example, can display the number of incidents assigned to employees while a stacked bar report can segment the incidents into categories. For more information, see Bar reports for mobile.

---

For more information

For general information about reports and when to use them, see Getting started with reports.

For instructions about enabling charts to be displayed on mobile apps, see Configure chart UI sections.

Time series reports for mobile

Time series reports display a series of data points over successive intervals of time, for example, the number of incidents opened over a certain period. The mobile platform supports the following time series reports, area, line, spline, and step line.
Tap a time series dashboard preview in an applet launcher to display a detailed screen of the report. Alternatively, you can navigate to a chart screen via a navigation button. Depending on the configuration settings, you can then tap on the report screen to see a complete list of records relating to the report. To magnify or contract the chart view, spread or pinch your fingers on the screen. A sparkline at the bottom of the screen highlights the magnified section of the chart. Tap the zoom-out button (🔍) to return the screen to its original magnification.

For time series charts that support multiple lines, select lines from the legend that you want to include or omit from the chart.

To list records corresponding the point highlighted in the chart line, tap the navigation arrow (➤) in chart header. Tap the list icon (⋮) in the screen header to display a list of all the records contained within the time series chart.

Available time series report types

Mobile dashboards support the following time series report types:

- Area
- Line
- Spline
- Step line

Available time series features

Mobile dashboards support the following time series report features from the web-based UI:

- Data configuration: Multiple line reports; accessibility line patterns
- Style configuration: Chart colors; chart titles; y-axis settings
- Decimal precision in y-axis (for line reports only). Enter 0 to show integers.

Unavailable time series features
The following time series features from the web-based UI are not available in mobile dashboards:

- Drill-down views; data labels on charts
- Legend configuration by admin
- Grouping by data set for more than one level of grouping
- Styling (other than chart colors)

Note the following display behavior within chart applets and charts:

- For some time series charts, the title for the y axis is not visible in the applet launcher preview of a report.

**Single score reports for mobile**

A single score report is a single value which you can use to share metrics or scores that are key to your business needs, for example the number of high priority incidents opened by your team.

Tap on the single score to see a complete list of records relating to the report. You can then tap on individual records for additional information.

**Available single score features**

Mobile dashboards support the following single score report features from the web-based UI:

- Data configuration: Aggregation type
- Style configuration: Default colors; color rules; chart titles
Unavailable single score features

The following single score features from the web-based UI are not available in mobile dashboards:

- Drill-down views
- Title styling options

Donut reports for mobile

Donut reports enable you to see the portions and percentages between different priorities at a single glance. For example, a donut report showing open incidents by priority, enables you to quickly see whether incident counts of different priorities are within acceptable ranges.

To display a donut report, tap on the dashboard preview in an applet launcher. Alternatively, you can navigate to a chart screen via a navigation button. The percentage figure displayed in the center of the chart corresponds to the percentage of records for that selected segment compared to the remaining segment records within the chart.

To display the number of records contained in that group, tap on a donut segment. To list records related to the selected segment, tap the segment, then tap the navigation arrow in the chart header. Tap the list icon in the screen header, to display a list of all the records contained within the donut chart.

Use the legend at the bottom of the chart to include or omit segments from the donut. As you add or remove filters, the percentage in the center of the donut report changes to correspond to the records currently displayed in the donut report.

Available donut report features

Mobile dashboards support the following donut report features from the web-based UI:

- Data configuration: aggregation type
- Style configuration: Chart colors; set palette; chart titles

Unavailable donut report features

The following donut report features from the web-based UI are not available in mobile dashboards:
• Grouping by data set for more than one level of grouping
• Legend configuration by admin
• Title (other than chart title)
• Styling (other than chart colors and palette)

Note: Admins can configure both donut and pie reports in the web-based UI. Both these reports have a donut visualization in the mobile platform.

Bar reports for mobile

Use bar reports to compare individual or aggregate scores across data dimensions. Each bar represents a specific category of data. A stacked bar report helps you differentiate between each segment within a category. Mobile dashboards support vertical bar graphs.

To display a bar report, tap a dashboard preview of a bar report in an applet launcher. Alternatively, you can navigate to a chart screen via a navigation button.

Tap on a bar or a segment of a bar to display the number of records contained in that selection. To view the records contained within a bar or a segment of a bar, tap the navigation arrow ( ) in the chart header. To display a list of all the records contained within the bar report, tap the list icon ( ) in the screen header.

To magnify or contract the chart view, spread or pinch your fingers on the screen. A sparkline at the bottom of the screen highlights the magnified section of the chart.

To display a list of all the records contained within the bar report, tap the navigation arrow ( ) in the chart header. To display a list of all the records contained within the bar report, tap the list icon ( ) in the screen header.

To magnify or contract the chart view, spread or pinch your fingers on the screen. A sparkline at the bottom of the screen highlights the magnified section of the chart.

A maximum of four bars are displayed when viewing the chart in a vertical position. Use the horizontal position to view additional bars.

For stacked bars, use the legend at the bottom of the chart to either include or omit segments from the bar report.

Available bar report features

Mobile dashboards support the following bar report features from the web-based UI:

• Data configuration: Aggregation type, stacked bar
• For stacked bars visualizations, support of one group by and stack by option
• Style configuration: Chart colors; set palette; chart title
• Decimal precision in y-axis. Enter 0 to show integers.

Unavailable bar report features
The following bar report features from the web-based UI are not available in mobile dashboards:

- Only vertical bar graphs are supported. Horizontal, pareto, and histogram bar reports are not supported.
- Legend configuration by admin
- Title (other than chart title)
- Styling (other than chart colors and palette)

**Handling mobile alerts**

Mobile alerts are an overlay pop-up window that displays over a form screen. The mobile alert displays an important message and a button to redirect you to an alternative screen.

**Before you begin**

Role required: admin

**About this task**

Mobile alerts are a way that administrators can highlight an important message and provide a call to action. In addition to the overlay pop-up window, a mobile alert icon also displays in the screen header. This icon gives you access to the alert even after closing the overlay pop-up window.
Procedure

1. Select either the action button or the dismiss button, after reading the mobile alert message.
   - Tap the dismiss button, named "Close" in the image, to close the overlay banner.
     You can view the alert again when you revisit the form screen that initially activated the overlay banner.
   - Tap the action button, named "View" in the image, for the instance to redirect you to the designated screen.

2. Optional: To view the overlay banner again, do one of the following.
   - Revisit the form screen that initially activated the overlay banner.
   - Tap the Alert icon in the screen header. The red spot indicates that you have not performed the call to action.
Mobile app feedback

Learn how to provide feedback regarding your ServiceNow in-app experience. ServiceNow strives to continuously improve the Mobile Platform, and therefore it is important to gather valuable feedback from you. While ServiceNow is not directly responsible for your in-app experience or the functionality available for your role, we aim to guide you toward directing your feedback and concerns to the appropriate stakeholder who can take direct action.

Starting with version 11.3 of the native client, anyone using a ServiceNow® mobile app can provide feedback regarding their in-app experience.

ℹ️ Note: This functionality only exists for ServiceNow General Availability applications. This functionality is not available in ServiceNow MAM (Intune and BlackBerry) apps.

Providing feedback

After completing three successful actions, the mobile app displays a prompt asking if you are enjoying the experience.

This prompt appears only once throughout all your sessions, to avoid additional interruptions.
If you are satisfied with the in-app experience, tap **Yes** to open the review page in the store. Here you rate the app and provide feedback about the benefits of the app and the functionality provided.
If you are not satisfied with the in-app experience, tap **Not really**. A message displays guiding you to direct your concerns with the appropriate stakeholder who can take the necessary action.

**Note:** Because feedback is gathered in different and unique ways for each environment, a generic message displays, directing you to the appropriate stakeholder.

If you want to provide feedback at any time, tap **Settings** from the navigation tab and select **Give feedback**.

### Accessibility features in mobile

To make the user interface of the ServiceNow Mobile Platform accessible to users with disabilities, ServiceNow includes features that support several specifications in the Web Content Accessibility Guidelines.
For more information
For general information on how ServiceNow is working with accessibility, see the following documentation topics:
Product accessibility
Product documentation for accessibility

Change mobile analytics views for accessibility
You have the option to change chart analytics views from color segments to black and white patterns. This option can be used for accessibility purposes.

Before you begin
Role required: user

About this task
The Chart accessibility button gives you the option to view charts in either black and white patterns or colored segments, for all chart analytic types. Using black and white patterns assists people with accessibility requirements.

Procedure
1. Tap More (≡) in the navigation bar.
2. Tap Settings (⚙).
3. On the Settings page, enable the Chart accessibility button to view charts in black and white patterns. Do not enable the button to view charts as colored segments.

   Note: If you do not see the Chart accessibility button, refer to your admin to enable this feature.

Results
Select the display option best suited for your requirements.
Comparison of graph with and without the accessibility option enabled
Defining font size settings for mobile devices

Adjust the size of text labels in the main screens of your mobile applications to enhance the readability of texts. This typography option can be used for accessibility purposes.

Before you begin
Role required: user

About this task
Specify the size of text by configuring the text-size definition in your mobile device settings.

Note: Text-label typography within the ServiceNow Mobile Platform isn’t yet fully available for mobile and legacy cards, map screens, Virtual Agent, dynamic segments, and mobile banners (iOS only).

Procedure
1. In your mobile device, go to the Settings option or Settings icon.
2. Tap the Accessibility option from the menu.
3. Set the font setting according to your requirements.

Note: If you do not see a change in the font settings, refer to your admin to enable this feature.
Results
The ServiceNow Mobile Platform displays text in your mobile device according to the text-label typography selection made.

Mobile app settings
Use the settings in your mobile applications to manage accessibility, notifications, location tracking, and more.

App PIN settings for mobile
Improve the security of your ServiceNow mobile apps by setting a security PIN.

Using a PIN with your mobile app
A security PIN is a 6-digit code you must enter for added security. When this feature is enabled, you are prompted to enter your pin:

- When logging into your app
- When the app has been inactive for five minutes
- When the app is closed using your operating systems force quit or force close option.
- When accessing the security settings for the app

If your mobile device uses faceID, touchID, or similar biometric security, you can use biometric authentication in place of the PIN.

For information for administrators on PIN settings for your instance, see Require an app PIN for the mobile app.

Create a PIN for a mobile app
Create a PIN for your mobile app.

Before you begin
Role required: admin

Procedure
1. Log into a ServiceNow mobile app.
2. On the navigation bar, tap the Settings.
4. Tap Require an app PIN.
5. When prompted Enter a 6 digit PIN for your app.
6. When prompted, re-enter the same 6 digit PIN. Your app is now configured with a security PIN.

7. Optional: Enable Unlock with Fingerprint Sensor to use your device’s fingerprint biometric security instead of your PIN.

Change your PIN for a mobile app
Change the pin for your ServiceNow mobile apps.

Before you begin
Role required: admin

Procedure
1. Log into a ServiceNow mobile app.
2. On the navigation bar, tap the Settings.
4. Tap Change app PIN.
5. At the prompt, enter your current app PIN.
6. When prompted, enter a new 6 digit PIN.
7. When prompted, re-enter the same 6 digit PIN. Your app is now configured with a new security PIN.

Remove your PIN from a mobile app
Remove the pin from a ServiceNow mobile app.
Before you begin
Role required: admin

Procedure
1. Log into a ServiceNow mobile app.
2. On the navigation bar, tap the Settings.
4. Tap Disable app PIN.
   Your app no longer configured with a security PIN.

Analytics settings for mobile
Enable analytics tracking of your user journey in a mobile application.

When launching a mobile application for the first time, European users are prompted to select whether to consent to analytics tracking. Non-EU/non-US users see a notice of enabled analytics tracking.

All users can enable/disable tracking in the mobile application at Settings > Analytics > Enable Analytics.

For detail on how an administrator can configure and track analytics for mobile, see User Experience Analytics.

Enable notification settings for mobile
Enable push notifications on your mobile device to receive information and messages directed to you.

Before you begin
Role required: admin

Procedure
1. Log in to the ServiceNow mobile app.
2. On the navigation bar, tap the Settings button.
3. On the settings page, tap notification preferences and then enable the Enable notifications toggle.

Location tracking for mobile
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.

The location setting on your mobile device takes precedence over the location setting in the mobile app. Grant permission on your mobile device to enable the app to track your location.

Note: To access geolocation settings on your mobile device, your administrator must install the geolocation plugin and enable location tracking on your user record. For more detail on geolocation configuration, see Mobile location tracking configuration.
Access by tapping the **Settings** tab in the navigation bar, and selecting **Location Tracking**.

On the **Geolocation** page, enable **Geolocation tracking**.

*Note:* You may see a prompt from your mobile device to grant location tracking permission to the mobile app. Select **OK** to enable geolocation tracking.

Use the **Track For** field to select the duration for which the app tracks your location. Once you have set a duration, you can see a time stamp indicating when location tracking ends on your **Settings** page.

### Using location tracking

When enabled, geolocation keeps a record of where you are when you last work on an instance. Location tracking continues even when there is no internet connection. For more information on location tracking, see [Location tracking](#).

---

### Display mobile screen performance data

Display performance-related data, like SQL counts, business rules, and REST API responses in a debug drawer at the bottom of the screen. This data enables administrators to measure the performance of each mobile screen and to identify any performance issues.

**Before you begin**

Role required: **admin**

**About this task**

Enable the debug drawer option to show the performance and timing related to each API call, when a page is loaded or an action is completed. Administrators use this information to assess network, server, and web performance.

*Note:* Administrators are not able to configure any settings of this feature.
**Procedure**

1. Log in to a ServiceNow mobile app.
2. On the navigation bar, tap the **Settings** button.
3. Choose from the following option to either enable or disable the debug drawer option.

<table>
<thead>
<tr>
<th>Option</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable debug drawer</td>
<td>a. On the settings page, tap <strong>Version</strong> rapidly five times.</td>
</tr>
<tr>
<td></td>
<td>The <strong>Debug Drawer</strong> option displays in the settings page.</td>
</tr>
<tr>
<td></td>
<td>b. Enable <strong>Debug Drawer</strong>, to display the debug drawer on each screen.</td>
</tr>
</tbody>
</table>

© 2022 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.
### Disable debug drawer

- On the settings page, tap `Version` rapidly five times.
- The **Debug Drawer** option no longer displays.

### Offline mode for mobile

Access and submit actions to records even if you do not have an internet connection.

Watch this three-minute video on how offline mode works. Learn how to download data, enable and disable offline mode, synchronize your outbox, and resolve synchronization errors. Demonstrates how offline mode works in the ServiceNow mobile app.

Plan ahead when you use offline mode. If 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 synchronize your device with the instance. The cached changes in your outbox update to the instance.

### Enable offline mode

Enable offline mode in your **Settings** tab. Tap **Offline** 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 synchronize to the server.
Disable offline mode and synchronize outbox

To return online in the mobile app, navigate to Settings > Offline Mode. On the offline mode screen, toggle off Offline Mode.
You can synchronize your changes while in Offline Mode in one of the following ways:

- From the Offline Mode screen, toggle off the Offline Mode button.
- From the Outbox screen, tap Sync All.

**Note:** After the synchronization 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 synchronize 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 synchronize your cache before it expires. To avoid losing your data due to a cache expiration, always synchronize your cache before and after going offline.
Resolve synchronization errors

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

You cannot synchronize changes that contradict changes made by other users while you were offline. For example, you may receive an error message if you try to synchronize 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. You can resolve any of the listed issues directly from your outbox.

You can either tap **Resolve** to fix the error or tap **Delete** to remove the issue from the list.
Scheduled offline caching

Enable scheduled offline caching to automatically download your cache according to your work schedule. Scheduled caching works in the background, so you are able to continue to use the app while the download completes. You can enable or disable this feature in your app settings.

**Note:** Enable Wi-Fi only to allow downloads only when you are connected to Wi-Fi.

Refresh mobile app meta-data

See the latest updates your administrators have made to your ServiceNow mobile apps by refreshing an applet launcher page.

**Before you begin**

Role required: none

Some changes to ServiceNow mobile app made by your administrator require a refresh of your app metadata before you can see the updated information. Changes that may require a refresh include:
Changes to push notification properties
Changes to applet launchers
Changes to the navigation bar

Avoid inconsistencies in your apps by refreshing your app when instructed by your administrator.

Procedure

1. Navigate to any applet launcher on your mobile app.
2. Scroll to the top of the applet launcher page.
3. Tap the screen and drag down until you see the refresh (_spinner) icon, then release.

Example: Refreshing an applet launcher
Accessing an instance on a mobile device web browser

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.

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

© 2022 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.
**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.

**Mobile plugins and upgrades**

Learn about upgrading to ServiceNow mobile, and how to install plugins to get your users started with the ServiceNow mobile platform.
Mobile plugins

Learn about the plugins used to enable ServiceNow mobile on your instance.

Core plugins

The following mobile plugins include the core functionality of ServiceNow mobile and the ability to interact with ServiceNow mobile apps. Plugins marked as base system plugins are automatically installed on your instance.

Note:
The Mobile Onboarding app is being deprecated!

With the Now Platform Rome release in September 2021, we started phasing out support for the Mobile Onboarding. Customers may no longer activate it, and we are not offering enhancements or non-critical bug fixes. Mobile onboarding features are available in the Now Mobile app for HR Service Delivery.

Core plugins

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>Description</th>
<th>Base System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile Studio</td>
<td>com.glide.sg-studio</td>
<td>Plugin for Mobile Studio and Mobile Agent</td>
<td>Yes</td>
</tr>
<tr>
<td>ServiceNow NowMobile App</td>
<td>com.glide.mobile-employee</td>
<td>Application and configurations required to setup the Now Mobile app.</td>
<td>No</td>
</tr>
<tr>
<td>Human Resources Scoped App: Mobile Employee Onboarding</td>
<td>com.sn_hr_onboarding</td>
<td>Activates the Mobile Onboarding app for HR.</td>
<td>No</td>
</tr>
<tr>
<td>Human Resources Scoped App: Mobile</td>
<td>com.sn_hr_mobile</td>
<td>Activates the HR mobile pieces for Now Mobile</td>
<td>No</td>
</tr>
</tbody>
</table>

Supporting plugins

These plugins are not required for ServiceNow mobile, but include extended functionality, such as offline mode, geolocation, and access to Virtual Agent.

Supporting plugins

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>Description</th>
<th>Base System</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG Offline support</td>
<td>com.glide.sg.offline</td>
<td>Provides offline support forServiceNow mobile.</td>
<td>No (This plugin is automatically installed with the Field Service Mobile plugin)</td>
</tr>
<tr>
<td>Service Management Geolocation Mobile</td>
<td>com.snc.service_management_m</td>
<td>Provides offline support forServiceNow mobile.</td>
<td>No (This plugin is automatically installed with the Field Service Mobile plugin)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Provides offline support forServiceNow mobile.</td>
<td>No (This plugin is automatically installed with the Field Service Mobile plugin)</td>
</tr>
</tbody>
</table>

© 2022 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.
Supporting plugins (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>Description</th>
<th>Base System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Management Geolocation</td>
<td>com.snc.service_management.geolocation</td>
<td>Enables geolocation capabilities for Service Management</td>
<td>No (This plugin is automatically installed with the <em>Field Service Mobile</em> plugin)</td>
</tr>
<tr>
<td>Geolocation</td>
<td>com.snc.geolocation</td>
<td>Core geolocation capabilities</td>
<td>Yes</td>
</tr>
<tr>
<td>Glide Virtual Agent</td>
<td>com.glide.cs.chatbot</td>
<td>Virtual Agent platform and other necessary plugins</td>
<td>No</td>
</tr>
</tbody>
</table>

Activating plugins on your instance

Install a plugin by searching for the plugin name in your plugins list. For information on the plugin activation process, see [Activate a plugin](#).

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Mobile plugins for Mobile Agent app

Use these plugins to extend functionality for the Mobile Agent app.

Base system applications for Mobile Agent app

<table>
<thead>
<tr>
<th>Application</th>
<th>Description</th>
<th>Availability</th>
<th>Paid</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITSM Mobile Agent</td>
<td>Delivers mobile-first experiences designed for agents to triage, act on, and resolve incidents. Agents can also view schedules, check who is on-call, respond to major incidents, and more.</td>
<td>Store</td>
<td>No</td>
</tr>
<tr>
<td>Asset Receiving Mobile</td>
<td>Provides the capabilities for receiving personnel to receive purchased assets in Mobile Agent app.</td>
<td>Instance</td>
<td>Yes</td>
</tr>
<tr>
<td>PPM Mobile</td>
<td>Project Portfolio Suite Mobile user experience. This provides access to project status and</td>
<td>Instance</td>
<td>Yes</td>
</tr>
</tbody>
</table>
## Base system applications for Mobile Agent app (continued)

<table>
<thead>
<tr>
<th>Application</th>
<th>Description</th>
<th>Availability</th>
<th>Paid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>project status report in tMobile Agent app.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer Service Mobile</td>
<td>Customer service mobile user experience.</td>
<td>Instance</td>
<td>Yes</td>
</tr>
<tr>
<td>Field Service Mobile</td>
<td>Manages the Field Service mobile components on an iOS or an Android mobile device. Field service agents and dispatchers can execute work order tasks, manage assets, and close work order tasks on their mobile device in online or offline mode.</td>
<td>Instance</td>
<td>Yes</td>
</tr>
<tr>
<td>GRC: Mobile</td>
<td>Task assignments, requests, approvals, and other follow-up actions for GRC applications directly from the Mobile Agent app app.</td>
<td>Store</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Receive timely notifications for current alerts, as well as risk and compliance status for your critical assets, vendors, and impacted essential business services.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Status Report</td>
<td>Project status report app delivers out of the box mobile experience for project managers, program managers, and portfolio managers to track their projects on the go. Use this app to track and update the factors that impact the health of the project such as risks, issues, decisions,</td>
<td>Store</td>
<td>No</td>
</tr>
</tbody>
</table>
### Base system applications for Mobile Agent app (continued)

<table>
<thead>
<tr>
<th>Application</th>
<th>Description</th>
<th>Availability</th>
<th>Paid</th>
</tr>
</thead>
</table>
| Security Incident Response Mobile  | Receive notifications when security incidents or security response tasks are assigned to you or to your assignment groups and begin remediation using the Mobile Agent app.  
Monitor the security incidents the Security Operations Center (SOC) or your assignment groups are responding to with search criteria that you enter. | Store        | Yes  |
| Vulnerability Response Mobile      | Receive notifications when vulnerability groups or vulnerability items are assigned to you or to your assignment group. Begin remediation on your most critical vulnerabilities using the Mobile Agent app. | Store        | Yes  |

### Mobile plugins for Now Mobile

Use these plugins to extend functionality for the Now Mobile app.

### Base system applications for Now Mobile

<table>
<thead>
<tr>
<th>Application</th>
<th>Description</th>
<th>Availability</th>
<th>Paid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Resources Scoped App: Mobile Employee Onboarding</td>
<td>Mobile Onboarding module for HR. Combined with Mobile Employee Experience Native Application plugin, items related to HR Onboarding will be shown in the Now Mobile app.</td>
<td>Instance</td>
<td>Yes</td>
</tr>
<tr>
<td>Application</td>
<td>Description</td>
<td>Availability</td>
<td>Paid</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
<td>--------------</td>
<td>-----</td>
</tr>
<tr>
<td>Safe Workplace suite</td>
<td>The ServiceNow Safe Workplace suite application allows customers to easily download all Safe Workplace applications at once, eliminating the need install the them individually.</td>
<td>Store</td>
<td>Yes</td>
</tr>
<tr>
<td>Emergency Response Management for Now Mobile</td>
<td>The Emergency Response Management for Now Mobile application provides a mobile experience for the Emergency Response apps. In the initial version, users will be able to report on their health status. Managers will also be able to view the health status of their employees, report status on behalf of their employees as well as the ability to reassign tasks which are currently assigned to their direct reports.</td>
<td>Store</td>
<td>No</td>
</tr>
<tr>
<td>Mobile Time Sheets</td>
<td>Create, submit, edit, and approve time sheets on the go with the new Mobile time sheets app. Time sheet users will be able to create and submit timecards and time sheets from their mobile device. Managers will be able to approve, reject, and recall timecards and time sheets from their mobile device. The app is supported on both iOS and Android devices.</td>
<td>Store</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Base system applications for Now Mobile (continued)

<table>
<thead>
<tr>
<th>Application</th>
<th>Description</th>
<th>Availability</th>
<th>Paid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walk-up Experience</td>
<td>Walk-up Experience for pre-built tech lounges. This plugin enables your IT organization to set up a walk-up contact channel to support online and on-site check in.</td>
<td>Instance</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Mobile plugins for Mobile Onboarding

Use these plugins to extend functionality for the Mobile Onboarding app.

ℹ️ Note:

The Mobile Onboarding app is being deprecated!

With the Now Platform Rome release in September 2021, we started phasing out support for the Mobile Onboarding. Customers may no longer activate it, and we are not offering enhancements or non-critical bug fixes. Mobile onboarding features are available in the Now Mobile app for HR Service Delivery."

Base system applications for Mobile Onboarding

<table>
<thead>
<tr>
<th>Application</th>
<th>Description</th>
<th>Availability</th>
<th>Paid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Resources Scoped App: Mobile Employee Onboarding</td>
<td>Mobile Onboarding module for HR. Combined with Mobile Employee Experience Native Application plugin, items related to HR Onboarding will be shown in the Now Mobile app.</td>
<td>Instance</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Migrate from the ServiceNow Classic mobile app to the ServiceNow Mobile Platform

Migrate from the Mobile Classic app to the Mobile Platform to take advantage of features such as rapid development, offline capabilities, and native mobile device feature integrations.

Understand the benefits of migration to ServiceNow® mobile

ServiceNow Mobile Platform is a native, mobile-first design that introduces new capabilities that were not available in the ServiceNow Classic mobile app: Migrate from the ServiceNow Classic mobile app to the ServiceNow Mobile Platform

- Rapid, low code application development using Mobile Studio and Mobile Card Builder.
- Base system applications including, but not limited to Customer Service Management, IT Service Management, and others as listed later in the topic.
• 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 Configuration and Navigation.

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 the ServiceNow Classic mobile app components into the Mobile Platform. 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. ServiceNow offers many base system store applications or plugins, some of which include experiences for:

• Customer Service Management
• Employee Experience Suite
• Field Service Management
• Incident Management
• IT Operations Management
• IT Asset Management
• IT Service Management
• Security Operations
• Risk
• HR
• Finance

Use these applications right away, or customize them to suit your business needs using Studio.

Mobile experience for Customer Service Management

Manage customer service cases from a mobile device with the Customer Service Management mobile application. Stay connected and access information in real time to complete tasks quickly. For more detail, see Mobile experience for Customer Service Management.

Mobile experience for Employee Experience Suite

Approve catalog requests, requested items, or change requests from anywhere using the ServiceNow Now Mobile app. For more detail, see Now Mobile app.

Mobile experience for Field Service Management

Manage your field service tasks anywhere using the Field Service Management 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

Create your own applications

Use Studio to make new applications, and give your users the ability 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 Building and configuring mobile apps.

Understand the current limitations for ServiceNow® mobile

General

- Limited support for client scripts. Basic functionality can be addressed with Mobile UI Rules. See, Mobile UI Rules.
- Geolocation and Offline mode is not supported in the Now Mobile app.

Offline mode

- Mobile apps do not automatically go into offline mode when internet connection is lost.

Mobile migration from Madrid to New York and later releases

Migrate your mobile applications in New York or later releases to take advantage of the improved features and continue editing within Studio.

Changes made during your upgrade

Note:
The Mobile Onboarding app is being deprecated!

With the Now Platform Rome release in September 2021, we started phasing out support for the Mobile Onboarding. Customers may no longer activate it, and we are not offering enhancements or non-critical bug fixes. Mobile onboarding features are available in the Now Mobile app for HR Service Delivery.

During the upgrade from Madrid to New York or later releases, the instance updates to the new mobile hierarchy by activating the Mobile Agent Native Client [com.glide.sg.agent_native_client] plugin. This installation creates the following changes:

Native clients

Adds the Native Clients [sys_sg_native_client] table. Records on this table represent the available native clients; Mobile Agent app, Now Mobile, and Mobile Onboarding.

Navigation bar

Adds the navigations [sys_sg_navigation] table. Records on this table represent a navigation bar for each of the native clients. Records on this table during the migration have their Legacy application [legacy_application] field enabled.

Notifications tab

Adds the notifications tabs [sys_sg_notifications_tab] table. Records on this table represent a tab for notifications on each navigation bar.

Settings tab
Adds the settings tabs \([\text{sys\_sg\_settings\_tab}]\) table. Records on this table represent a tab for settings on each navigation bar.

This upgrade includes new features such as application launchers and a configurable navigation bar. Any unmodified base system mobile applications installed on your instance are automatically updated to work with the new design, and can be used with Studio right away. For more detail on the mobile hierarchy used in New York and later, see Mobile hierarchy.

Modified base system applications, and applications that you have created in Madrid will continue to work after the upgrade. These applications will not be configurable in Studio until after you have run the mobile migration script.

Post-upgrade considerations

After an upgrade, consider the following information to confirm that your mobile implementation is working as expected, and ensure that mobile migration script runs.

**Modified base system applications**

Document any changes you have made to mobile applications provided by ServiceNow, as well as any applications you have created. Test each of these applications to ensure that they continue to function as you expect.

**Use the Debug Upgrade feature**

The debug upgrade feature can help you to quickly diagnose upgrade issues. For information on this feature, see Debug Upgrade.

A video training course on this tool is available. To view this course, see Using Debug Upgrade

**Review skipped records**

To prevent overriding your customizations, the upgrade process does not update records that you have modified. Instead, the upgrade process notes this skipped record in the upgrade logs. For more detail on handling skipped records, see Process the skipped records list.

A video training course on resolving skipped records is available. To view this course, see Upgrade Skipped Records.
**Review functionality after upgrade**

Once you have upgraded your instance and run the migration script, regression testing can help ensure that your users can continue to work as expected after an upgrade. A regression test is a review of your applets, screen UI policies, and functions to make sure that they are working as intended.

**Running Mobile migration script**

This script converts your custom applications and any modified base system application to the new mobile schema available in the New York release. The script only changes the current scope when it runs. If you have more than one scoped mobile application, you must run the script for each scope.

After an upgrade, the option to run the migration script appears when you first access a custom application, or a base system application that you have modified. For example, when opening a modified or custom applet record. You can also see the migration prompt when accessing the applet picker in Studio by browsing to Mobile Studio > Applets and clicking the pop-out icon ( ). The migration prompt displays if any of the applets shown the picker require migration.

![Migrate to New York Version](image)

After the script completes, you may be prompted to resolve collisions detected by the migration process. Collisions are records created by ServiceNow that you have modified, and are not automatically upgraded. Collisions can only occur when you have modified a base system application before your upgrade to New York or later releases.

![Migration Collisions](image)

Click the View Collisions to resolve these collisions. For detail on this process, see Troubleshooting mobile migration script results.

**Changes made by the mobile migration script**

Click Migrate to start the migration script for the current scope. The migration script migrates all records within the scope, not just the applet you have opened.

**Applications and folders transition to applet launchers**

The legacy Madrid schema used mobile applications and folders to organize your applets. The Now Mobile schema, uses applet launcher screens, which are divided into UI sections. Applet launcher is accessed by tapping on tabs in the navigation bar which appears at the bottom of your app screens.
The migration script creates an applet launcher for each mobile application record. The script converts each folder in the original mobile application to a new horizontal icon section within that applet launcher. The script then creates an icon in the icon section for each applet with the folder. Hidden screens do not appear in the icon section. The script then adds a tab to the navigation bar for each of the new applet launchers.

The example image shows how the incidents application appears after the migration process. The original folders (My Incidents and Group Incidents) display as UI sections in the Incidents applet launcher. These UI sections can scroll horizontally to show as many applets as needed. The Incidents application is accessible by tapping the Incidents tab in the navigation bar.

After migration, the script removes the legacy Folder [sys_sg_folder] and Mobile Application [sys_sg_application] records.

For more detail on the navigation bar, applet launchers and their UI sections, see Navigation bar, and Launcher screens.

**Form migration**

The Form applet replaces the master detail screens used to view record forms in the Madrid release. The migration creates a form screen [sys_sg_form_screen] record. The script creates segments for each embedded screen in the original
master detail screen. Any button [sys_sg_button] records associated to the original master detail screen change to associate with the new form applet.

Map migration

Map applets did not use an item view to display fields in map cards in the Madrid release. The migration script creates an item view [sys_sg_item_view] record for each map applet using the Title, Tag, Sub-title, and Info fields from the original map applet.

Calendar migration

The migration script creates time span item stream [sys_sg_time_span_item_stream] records for each calendar, and associates the calendars original data item to the new item stream. The migration script also creates a form applet [sys_sg_form_screen] record, and migrates the buttons from the calendars original embedded screen to the new form.

Item streams and Item configurations

The migration script creates an item stream [sys_sg_item_stream] record for each screen in the scoped application. The original data item record associated with the legacy application changes to associate with the new item stream record. The script creates time span item stream [sys_sg_time_span_item_stream] records for each calendar screen, and location item stream [sys_sg_location_item_stream] records for map screens. These two
tables extend from the item stream table, but are used specifically for these screen types.

**Screen Cleanup**

The following fields are no longer used in Screen records. The script removes these fields from call records on the Screen [sys_sg_screen] table.

- User Roles [application_roles]
- Order [order]
- Parent [parent]
- Parent table [parent_table]
- Data Item [sys_sg_data_item]
- Hidden [hidden]

In addition, the script also removes values from the following fields on Map screen [sys_sg_map_screen] records:

- Data item table [data_item_table]
- Title [title]
- Sub-title [subtitle]
- Info [info]
- Location [location]
- Tag [tag]
- Tag font color [tag_font_color]
- Tag background color [tag_background_color]
- Tag Style [tag_style]
- Phone [phone]
- Pin color type [pin_color_type]
- Pin color [pin_color]

The script removes values from the following fields on Item configuration [sys_sg_master_item] records:

- Table [table]
- Screen [screen]
- Condition [condition]
- Condition Order [condition_order]

The script removes the value in the Item View [item_view] field of Details screen [sys_sg_details_screen] records.

The script removes the value in the Item View [item_view] field of List screen [sys_sg_list_screen] records.

The script removes the value in the Data Item [data_item] field of Item View [item_view] records.
More Resources
For more information on the migration process, see the Mobile Migration Guide for New York on the ServiceNow community site. https://community.servicenow.com/community?id=community_article&sys_id=f5121a33dba7f788ff8a345ca961957

Run the mobile migration script
Run the mobile migration script to convert Madrid mobile applications you have created or modified to use the new mobile hierarchy.

Before you begin
Role required: admin
Mobile applications created in the Madrid release still work in the San Diego release, but cannot be edited in Studio. To continue editing in Studio, and to take advantage of new features available in the San Diego release, run the mobile migration script.

Procedure
1. Navigate to All > System Applications > Studio.
2. Open a scoped application that you have created, or a base system application that you have modified.
3. Click Upgrade when prompted to start the migration

Troubleshooting mobile migration script results
Find solutions to common issues after running the mobile migration script.

Log error messages
The mobile migration script adds entries to the Log [syslog] table when it encounters and error. You can review these logs by navigating to System Logs > System Log > All. Listed here are errors the mobile migration script may add to the logs.

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid instance scope provided</td>
<td>If you see this message, the migration script was aborted. Run the migration script again to complete the migration. You can do rerun the migration script by reopening Studio and selecting the scope.</td>
</tr>
<tr>
<td>Please activate com.glide.mobile-employee first before migration.</td>
<td>The ServiceNow NowMobile App Screens and Applet Launcher [com.glide.mobile-employee] plugin must be active to run the migration script. Ensure that this plugin is active. If you see this message, the migration script was aborted. Run the migration script again to complete the migration. You can do rerun the migration script by reopening Studio and selecting the scope.</td>
</tr>
</tbody>
</table>

© 2022 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.
## Error Messages (continued)

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cannot perform migration task on customized record.</td>
<td>The record causing this error appears immediately after this message. A customization on this record prevented the migration script from changing this file. The migration script skips this file, and continues to run. The named file is inaccessible in Studio.</td>
</tr>
</tbody>
</table>

## Collisions

Collisions can occur in base system applications that you have customized before the upgrade to New York or later versions. If the migration script detects any collisions, it prompts you to review them after the script execution completes.

Click the **View Collisions** button to view a filtered list of upgrade details [sys_upgrade_history_log] records. This list shows the records within the current scope that the upgrade process has skipped. To resolve a conflict, click a record on this list to open the record, then click the **Resolve Conflicts** button.

The **Resolve Conflicts** form shows the base system version of the record alongside the customized version Fields that are different between versions are highlighted with a darker background.
Use the arrow buttons (>) to move values from one version to the other. After making your changes, click the **Save Merge** button to save your changes. You can also click the **Revert to Base System** button to discard your customizations and use the unmodified version of the record.

**Note:** Use an update set to capture the changes you make while resolving conflicts. You can use this update set to apply your changes in other instances. For details on using update sets see [System update sets](#).

A common collision issue is Master detail [sys_sg_master_detail_screen] records. Master details records are no longer a part of the mobile schema as of the New York release. These records are replaced with new list [sys_sg_list_screen] and form [sys_sg_form_screen] screen records. They are normally deleted as part of the upgrade process, but if they have been customized, the script does not automatically delete them. If you have, for example, renamed a base system application, this kind of collision can occur.

To resolve the issue, check studio to make sure that your applet is still available and working as expected. Once you have confirmed that the applet is available, you can delete the master detail record.

### Common migration issues

#### An applet is missing

After migration, your applets should be visible in the **Applications** tab in the navigation bar. If the applets do not appear, you can manually migrate these applets.

1. In Studio, open **Mobile Studio > Application Menu** in the application explorer, and select the app where you are missing an applet.
2. In the **Navigation Tabs** related list, click the **Applications** navigation tab.
3. Note the **Applet Launcher** associated with the **Applications** tab.
4. In Studio, open **Mobile Studio > Applet Launchers**, and open the applet launcher noted in the last step.
5. In the applet launcher form, select a UI section with the same name as the folder the missing applet was located in before the migration.
6. Find the missing applet in the **All Applets** list, and move it to the **Selected Applets** list.
7. Click **Save**.

#### A related list is missing from an applet

This issue may be the result of an outdated reference on the related lists mapping [sys_sg_related_list_map] table. You can re-associate the **Destination Screen** for your related list to resolve the issue.

1. In Studio, navigate to **Mobile Studio > Applets**, and select the applet with the missing segment.
2. Click the **Form Screen** tab.
3. Click **Body**.
4. Click the **Related Lists** button.
5. Check the list for items that have an empty **Destination Screen** value.
6. Click the list item, and select a value in the **Destination Screen** field.
Users are not prompted to enter input parameters in Field Service Management or ITSM applets

Normally parametrized applets prompt your users for a value. If you are no longer seeing this prompt after a migration, use these steps to correct the issue.

1. Open the UI Parameter [sys_sg_ui_parameter] list by entering sys_sg_ui_parameter.list in the filter navigator for your instance.
2. Find the parameter which is not generating a prompt for your users.
3. Check the value of the Screen field. If this field appears empty, it may be pointing to a Master-detail screen [sys_sg_master_detail_screen] record.
4. Update the field by selecting the applet [sys_sg_screen] record used by this parameter.

Incorrect results for a customized Field Service Management or ITSM applet.

This issue can occur if you have added a customer parameter to a base system.

1. Open the Screen Parameters mappings [sys_sg_screen_param_map] list by entering sys_sg_screen_param_map.list in the filter navigator for your instance.
2. Find the record with the Item Parameter field matching the item parameter you have added to your data item.
3. Check the value of the Screen field. If this field appears empty, it may be pointing to an unused Master-detail screen [sys_sg_master_detail_screen] record.
4. Update the field by selecting the applet [sys_sg_screen] record used by this parameter.

Migration from New York and later releases

Learn about migration from the New York to later releases.

Upgrading from the New York release to later releases of ServiceNow mobile does not require any special configuration steps. The upgrade process will have no impact on your existing mobile applications.

Building and configuring mobile apps

Build and configure a mobile experience for any of the three ServiceNow mobile apps.

Requirements

ServiceNow mobile is available for instances upgraded to Madrid or later releases.
The ServiceNow mobile platform, like any other application, is built using records on tables within your instance. While configuration can be done through the platform user interface, the ServiceNow mobile platform has its own experience built out in Studio. With Mobile Studio, you can use to quickly and easily create mobile experiences.

Mobile tutorials

Use tutorials to see how you can build or customize mobile applications suited to your business needs. These tutorials include step-by-step examples for creating mobile applications for common business cases.

- For a list of available tutorials on ServiceNow mobile, see Mobile tutorials.
Design considerations for ServiceNow mobile

Before you begin building and customizing your applications for ServiceNow mobile, take a look at the design considerations documentation. This content contains information from mobile designers and product managers to help you design an application optimized for the mobile environment. View the design considerations documentation at Design considerations for mobile apps.

Mobile Hierarchy

⚠️ Note:

The Mobile Onboarding app is being deprecated!

With the Now Platform Rome release in September 2021, we started phasing out support for the Mobile Onboarding. Customers may no longer activate it, and we are not offering enhancements or non-critical bug fixes. Mobile onboarding features are available in the Now Mobile app for HR Service Delivery.

The ServiceNow mobile platform is designed using a hierarchy structure. At the top, are the three different mobile apps, Now Mobile, Mobile Agent, and Mobile Onboarding. Familiarize yourself with is hierarchy to deliver an exceptional mobile experience. Before getting into configuration, find more details about the mobile fundamentals and hierarchy at Mobile hierarchy.

Mobile Customization

Use Studio to modify base system mobile app components, or create your own components. Studio provides an environment where you can perform most customization and creation tasks that relate to mobile apps. For more details on Studio, see Mobile Studio.

To test your configuration, you can use the app that is installed on your mobile device. To get the mobile apps, go to the Apple App Store or the Google Play store and search for ServiceNow to download the available mobile apps.

ServiceNow mobile terminology changes

Starting from the Rome version ServiceNow mobile has updated the mobile terminology. The updated wording helps you better understand which items you are configuring within a mobile app, how the configurations affect the on-screen view, and how the items relate to one another.

⚠️ Note: The name changes apply from Rome and later versions. All versions earlier than Rome use the earlier terminology.

Terminology changes for the Rome version

The table compares the terminology used from the Quebec version and earlier with the replaced terms used from the Rome version and later.

<table>
<thead>
<tr>
<th>Terms used from Quebec and earlier</th>
<th>Terms used from Rome and later</th>
</tr>
</thead>
<tbody>
<tr>
<td>applet</td>
<td>screen</td>
</tr>
<tr>
<td>applets</td>
<td>screens</td>
</tr>
<tr>
<td>applet launcher</td>
<td>launcher screen</td>
</tr>
<tr>
<td>applet launcher section</td>
<td>launcher section mapping</td>
</tr>
<tr>
<td>applet launcher tab</td>
<td>launcher tab</td>
</tr>
</tbody>
</table>

© 2022 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>Terms used from Quebec and earlier</th>
<th>Terms used from Rome and later</th>
</tr>
</thead>
<tbody>
<tr>
<td>applet tab</td>
<td>screen tab</td>
</tr>
<tr>
<td>browser applet</td>
<td>mobile web screen</td>
</tr>
<tr>
<td>calendar applet</td>
<td>calendar screen</td>
</tr>
<tr>
<td>chart applet</td>
<td>chart screen</td>
</tr>
<tr>
<td>chart</td>
<td>analytics preview</td>
</tr>
<tr>
<td>chart section</td>
<td>analytics section</td>
</tr>
<tr>
<td>chart UI section</td>
<td>analytics section</td>
</tr>
<tr>
<td>client theme</td>
<td>mobile app theme</td>
</tr>
<tr>
<td>content section M2M item stream container</td>
<td>content section M2M stream container</td>
</tr>
<tr>
<td>destination applet launcher</td>
<td>destination launcher screen</td>
</tr>
<tr>
<td>dynamic form screen segment</td>
<td>dynamic record screen segment</td>
</tr>
<tr>
<td>dynamic segment item stream</td>
<td>dynamic segment stream</td>
</tr>
<tr>
<td>employee directory applet</td>
<td>employee directory screen</td>
</tr>
<tr>
<td>form applet</td>
<td>record screen</td>
</tr>
<tr>
<td>form screen</td>
<td>record screen</td>
</tr>
<tr>
<td>form screen segment</td>
<td>record screen segment</td>
</tr>
<tr>
<td>global search configuration</td>
<td>search configuration</td>
</tr>
<tr>
<td>global search M2M item configuration</td>
<td>search M2M item configuration</td>
</tr>
<tr>
<td>header title</td>
<td>launcher header title</td>
</tr>
<tr>
<td>header view config</td>
<td>header card</td>
</tr>
<tr>
<td>icon section</td>
<td>legacy icon section</td>
</tr>
<tr>
<td>icon UI section</td>
<td>legacy icon section</td>
</tr>
<tr>
<td>item configuration</td>
<td>list item configuration</td>
</tr>
<tr>
<td>item parameter</td>
<td>data parameter</td>
</tr>
<tr>
<td>item section</td>
<td>record section</td>
</tr>
<tr>
<td>item stream</td>
<td>list stream</td>
</tr>
<tr>
<td>item stream container</td>
<td>stream container</td>
</tr>
<tr>
<td>item stream container M2M item stream</td>
<td>stream container M2M stream</td>
</tr>
<tr>
<td>item stream M2M item configuration</td>
<td>list stream M2M item configuration</td>
</tr>
<tr>
<td>item stream M2M segment</td>
<td>list stream M2M segment</td>
</tr>
<tr>
<td>item UI section</td>
<td>record section</td>
</tr>
<tr>
<td>item view</td>
<td>legacy card</td>
</tr>
<tr>
<td>list applet</td>
<td>list screen</td>
</tr>
<tr>
<td>location item stream</td>
<td>map location stream</td>
</tr>
<tr>
<td>map applet</td>
<td>map screen</td>
</tr>
<tr>
<td>Terms used from Quebec and earlier</td>
<td>Terms used from Rome and later</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>mobile favorites</td>
<td>saved items</td>
</tr>
<tr>
<td>mobile favorites deny list</td>
<td>saved deny list</td>
</tr>
<tr>
<td>mobile section</td>
<td>UI section</td>
</tr>
<tr>
<td>name (on card and card template)</td>
<td>type (on card and card template)</td>
</tr>
<tr>
<td>native client</td>
<td>mobile app configuration</td>
</tr>
<tr>
<td>navigation</td>
<td>navigation bar</td>
</tr>
<tr>
<td>navigation section</td>
<td>icon section</td>
</tr>
<tr>
<td>navigation section destination</td>
<td>icon section destination</td>
</tr>
<tr>
<td>navigation section destination applet</td>
<td>icon section destination screen</td>
</tr>
<tr>
<td>navigation section destination applet launcher</td>
<td>icon section destination launcher</td>
</tr>
<tr>
<td>navigation section destination button</td>
<td>icon section destination function</td>
</tr>
<tr>
<td>navigation UI section</td>
<td>navigation section</td>
</tr>
<tr>
<td>parameter input</td>
<td>input</td>
</tr>
<tr>
<td>parameter page</td>
<td>input form page</td>
</tr>
<tr>
<td>parameter screen</td>
<td>input form screen</td>
</tr>
<tr>
<td>parameter section</td>
<td>input form section</td>
</tr>
<tr>
<td>parameter sections</td>
<td>input form section</td>
</tr>
</tbody>
</table>

**Note:** From San Diego and later versions this term is called "input form section M2M".

| quick actions menu map             | quick actions menu            |
| section                            | UI section                    |
| segment                            | screen segment                |
| sg incremental result (table)      | incremental offline result    |
| time span item stream              | calendar event stream         |
| use view config                    | use card                      |
| view config                        | card                          |
| view config element                | card element                  |
| view config element attribute      | card element attribute        |
| view config element attribute name | card element attribute type   |
| view template                      | card template                 |
| view template slot                 | card template element         |
| view template slot attribute       | card template element attribute |
**Mobile tutorials**

Use this reference to find all tutorials available in the ServiceNow mobile documentation.

**Mobile parameter tutorials**

Parameters are a way of creating a variable or a placeholder that is waiting for input from either the user or the database. The variable then queries the database or the user for more information. 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 could create a data item that allowed users to filter incidents by priority. The data item parameter would hold a place in the Priority field, so when the user opened the screen they would may select the priority. For more information on configuring a data item with parameters, see **Configure a parametrized data item**.

When you add parameters to an action function, that parameter looks for information from the user or the database before updating a record. For example, you could create an action function that allowed a user to update the assignee field from a swipe action. The action function parameter would hold 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**.

Use the following tutorials to create action and data items with parameters and associate them with their respective functions.

**Tutorial: Configure a data item with parameters**

Use a parameter to query a user for a priority value, then display a list of incident records matching that priority. This example demonstrates how you can use a parameter to get a value from user input and apply that value to an applet.

**Tutorial: Configure an action with parameters**

Configure a swipe action on a list to assign a record from the list to a user. This example demonstrates how you can use a parameter to get user input and apply that input to a record using an action.

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

Navigate from an incident record to the user record for the incident’s caller. In this example demonstrates how you can use a parameter to store information from a record (the caller), and use that to navigate to a specific record on another table.

**Tutorial: Configure a data item with parameters**

Follow this tutorial to create an applet that allows the user to filter by priority before viewing a list. Use this tutorial as a guideline to help you understand how data items with parameters work in the mobile application.

In this example, you use a parameter to prompt a user for a priority, then display a list of incidents matching that priority. The parameter accepts the selection from the user and passes that information to a data item which is used to filter the list of incidents. Setting up an applet that uses a parametrized data item includes several steps. In this tutorial you will:

- Create a data item with parameters.
- Configure the applet to use the parametrized 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 a mobile application using Guided Application Creator**.
Create a data item with parameters
Create a parametrized data item for a list of incidents.

Before you begin
Role required: admin

About this task
In these steps you will create a parametrized data item. This means creating a parameter within the data item form, and adding the parameter to your data item's conditions, so the value of the parameter is used to filter your data item.

Procedure
1. In Studio, navigate to **Mobile Studio > Data Items**.

2. Click the pop out icon (🗂) to open the data items list in a tab.

3. Click **Create a new data item**.

4. In the **New Data Item** form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a name for your data item. For example, Open Incidents</td>
</tr>
<tr>
<td>Table</td>
<td>Table from which the data items get its records. In this example, choose Incident (incident).</td>
</tr>
<tr>
<td>Description</td>
<td>Enter a description for your data item</td>
</tr>
<tr>
<td>Condition Type</td>
<td>Method used to create conditions for the data item. For this example, select Declarative.</td>
</tr>
</tbody>
</table>

5. In the Query condition, add the following conditions.
   - Active is true
   - State is one of **New**, In progress, or **On hold**

6. Click **Save**.
Example
Data item completed using the preceding steps.

7. In the Parameter Definition section, click the add icon (➕).
8. In the parameter definition form, fill in the fields.

Parameter definition form

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the parameter. For this example, use Priority.</td>
</tr>
<tr>
<td></td>
<td>Note: You can have multiple parameters with the same name. Choose a name that you can distinguish easily when creating parameters.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of parameter. For this example, select String.</td>
</tr>
</tbody>
</table>

9. On the parameter definition form, click Save.
10. In the Data Item tab, from the condition builder, click the AND button that appears to the left of the State condition.
11. From the Choose field list, select Priority.
12. In the same row as the Priority field in the condition builder, click the contextual reference value icon (🔗).
13. Add the item parameter you created in previous steps.
14. Click **Save**.

**Example**

Data item conditions using the parameter created in the preceding steps.

![Data Item Conditions](image)

Assign the data item to an applet

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

**About this task**

In these steps you will create a list applet. The list applet uses the parametrized data item you created in the preceding steps.

**Procedure**

1. In Studio, navigate to **Mobile Studio > Applets**.
2. Click the pop out icon ( ), to open the applets tab.
3. Click **Create an applet**.
4. On the Create an applet form, fill in the fields. Any fields not mentioned in the table can be left at their default values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the applet. For this example enter Open incidents.</td>
</tr>
<tr>
<td>Icon</td>
<td>Icon displayed for this applet. Select any icon.</td>
</tr>
<tr>
<td>Description</td>
<td>Description for your applet. Enter a description.</td>
</tr>
<tr>
<td>Choose a template</td>
<td>The template used for the applet. For this example, select <strong>List</strong>.</td>
</tr>
</tbody>
</table>

5. Click **Create New**.

6. In the Open incidents applet tab, from the Data Item list, select Open incidents.

7. In the Field Configurations section, use the list to add the following fields.
   - Number
   - Priority
   - Short description
- Assigned to
- Assigned to > Avatar
- State

8. Click the **Form Screen** tab.

9. Above the **Selected Fields** list, click the **Replicate from primary** button ( ).

10. Click **Save**.

11. Click the **List Screen** tab to return to the list configuration. The parameter settings section is now visible under the data item.

12. In the **User input parameter definition** section, click the add button ( ).

13. In the **User Input** pop-up form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for your user input. For this example enter Priority.</td>
</tr>
<tr>
<td>Input type</td>
<td>The type of input. Select <strong>Choice List</strong>.</td>
</tr>
<tr>
<td>Table</td>
<td>Table from which your choice list will be created. Select <strong>Incident [incident]</strong>.</td>
</tr>
<tr>
<td>Field name</td>
<td>Field from which your choice list will be created. The available fields are determined by the table selected in the <strong>Table</strong> field. Select <strong>Priority</strong>.</td>
</tr>
<tr>
<td>Input style</td>
<td>Style for the user input. Select <strong>Inline</strong>.</td>
</tr>
<tr>
<td>Default value type</td>
<td>Default value type for the input. Select <strong>None</strong>.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>Whether the user input is mandatory. Leave this option unselected.</td>
</tr>
<tr>
<td>Placeholder text</td>
<td>Text that appears in the input before the user makes a selection. Enter Priority.</td>
</tr>
<tr>
<td>Multi-select</td>
<td>Whether the user can select more than one value from the list. Leave this option unselected.</td>
</tr>
</tbody>
</table>

14. Click **Save**.

15. In the **Screen UI Parameter Mapping** section, click the add button ( ).

16. In the **Screen UI Parameter mapping** pop-up form, fill in the fields as needed.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item parameter</td>
<td>Select the parameter from the data item you created in previous steps.</td>
</tr>
</tbody>
</table>
### Field | Value
--- | ---
UI parameter | Select the UI parameter you created in this applet in the previous steps.

17. Click **Save**.

18. In the Applet form, click **Save**.

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

---

**Tutorial: Configure an action with parameters**

Follow this tutorial to create a swipe action that allows a user to assign an incident from a mobile list. Use this tutorial as a guideline to help you understand how actions with parameters work in a mobile app.

In this tutorial, you create a parametrized action item to enable your users to assign an incident from a list using a swipe action. Creating a parametrized action item requires several steps. In this tutorial you will do the following:

- 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.

**Before you begin**
Role required: admin

**About this task**
In these steps you will create an action item with parameters.
Procedure
1. In Studio, navigate to Mobile Studio > Action items.
2. Click the pop out icon ( đôi ) to open the Action items list in a tab.
3. Click the Create New button.
4. On the Action Item form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the action item. For this example, enter Assign to</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the action item</td>
</tr>
<tr>
<td>Type</td>
<td>Type of action item. Select Update.</td>
</tr>
<tr>
<td>Application</td>
<td>Scoped application for the action item. This field is automatically</td>
</tr>
<tr>
<td></td>
<td>populated with the current application.</td>
</tr>
<tr>
<td>Table</td>
<td>Table used for the action item. Select Incident [incident].</td>
</tr>
<tr>
<td>Use current record as condition</td>
<td>Whether the action item will use the current record as its condition. Leave this field selected.</td>
</tr>
<tr>
<td>Set field values</td>
<td>Field values that this action item will set. Leave this field blank for now. You will create field values in later steps.</td>
</tr>
</tbody>
</table>

5. Click Submit.
6. In the Item Parameters -> Writeback related list, click New.
7. On the Item Parameters -> Writeback form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the item parameter. For this example, enter Asignee</td>
</tr>
<tr>
<td>Type</td>
<td>Item parameter type. Select String.</td>
</tr>
</tbody>
</table>

8. In the Item Parameter tab, click Submit.
9. In the Action Item tab, from the Set field values condition builder, search for Assigned to.
10. In the same row as the Assign to field in the condition builder, click the contextual reference value icon ( đôi ).
11. Add the Assignee item parameter you created in previous steps.
12. Click **Update**.

**Create an action function**
After creating the action item with parameters, configure an action function to associate the action item with.

**About this task**
In these steps, you use the action item you created in previous steps in a function. You will use this function in a list applet to allow your users to assign incidents.

**Procedure**

1. In Studio, navigate to **Mobile Studio > Functions > Actions**.
2. Click the pop out icon ( ) to open the Action items list in a tab.
3. Click **Create New**.
4. In the **Function form**, fill in the fields.
   Fields not mentioned in the table can be left at their default values.

**Function form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for the function. For this example, enter Assign.</td>
</tr>
<tr>
<td>Description</td>
<td>Description for the function</td>
</tr>
<tr>
<td>Type</td>
<td>Type of function. Select <strong>Action Item</strong>.</td>
</tr>
<tr>
<td>Application</td>
<td>Scoped application for the function. This field is automatically populated with the current application.</td>
</tr>
<tr>
<td>Context</td>
<td>Context for the function. Select <strong>Record</strong>.</td>
</tr>
<tr>
<td>Field</td>
<td>Value</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Action Item</td>
<td>Action item used in the function. Select the <strong>Assign to</strong> action item created in the previous steps.</td>
</tr>
</tbody>
</table>

5. At the bottom of the form, open the **Condition** tab.

6. In the **Table** field, select **Incident [incident]**.

7. At the bottom of the form, open the **Messages** tab.

8. In the **Success message** field, enter **{{number}} has successfully updated**. 

9. Click **Submit**.

**Completed function form**

<table>
<thead>
<tr>
<th>Name</th>
<th>Assign</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Assign an incident</td>
</tr>
<tr>
<td>Type</td>
<td>Action Item</td>
</tr>
<tr>
<td>Context</td>
<td>Record</td>
</tr>
<tr>
<td>Action Item</td>
<td>Assign to</td>
</tr>
</tbody>
</table>

10. In the UI parameter related list, click **New**.

11. In the **UI Parameter** form, fill in the fields.

   Fields not mentioned in the table can be left at their default values.

**UI Parameter form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the UI Parameter. For this example, enter <strong>Assign</strong>.</td>
</tr>
<tr>
<td>Parameter type</td>
<td>The parameter type. Select <strong>Button</strong>.</td>
</tr>
<tr>
<td>Input Style</td>
<td>Determines where the input is handled inline or in a pop-up. Select <strong>Pop-up</strong>.</td>
</tr>
<tr>
<td>Button</td>
<td>The function used when the user taps or swipes to activate the function. This field is automatically filled with the function created in the previous steps.</td>
</tr>
<tr>
<td>Field</td>
<td>Value</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Input source</td>
<td>Input source for the action. Select <strong>User Input</strong>.</td>
</tr>
<tr>
<td>Input type</td>
<td>The method the user uses to enter input. Select <strong>Choice List</strong>.</td>
</tr>
<tr>
<td>Table Name</td>
<td>The table related used in this action. Select <strong>Incident [incident]</strong></td>
</tr>
<tr>
<td>Field Name</td>
<td>The field used for this action. Select <strong>Assigned to</strong>.</td>
</tr>
</tbody>
</table>

12. Click **Submit**.

13. In the **Assign** function record, 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**
14. From the Assign action function, click **Update**.

**Assign the action function to an applet**

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

**About this task**

Now that you have an action function, you a list applet to apply your action to. In these steps you will create a data item, list applet, and swipe function definition.

**Procedure**

1. Create a data item for open incidents.

   a. In Studio, navigate to **Mobile Studio > Data Items**.

   b. Click the pop out icon ( ) to open the data items list in a tab.

   c. In the Data Items tab, click **Create New**.

   d. In the **New Data Item** form, fill in the fields.

   **Data item form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Title for the data item. For this example, enter Open Incidents</td>
</tr>
<tr>
<td>Table</td>
<td>The name of the table you want the data item to pull information from. Select <strong>Incident [incident]</strong></td>
</tr>
<tr>
<td>Description</td>
<td>Optional additional details about the data item.</td>
</tr>
<tr>
<td>Condition Type</td>
<td>Whether the data item condition is declarative or scripted. Select <strong>Declarative</strong>.</td>
</tr>
</tbody>
</table>

   e. In the query condition, add the following condition.

   • `<State> <is one of> <New, In Progress, On Hold>`

   f. Click **Save**.
2. Create an applet for open incidents.

   a. In Studio, navigate to **Mobile Studio > Applets**.

   b. Click the pop out icon (⤢) to open the applets tab.

   c. Click **Create an applet**.

   d. Create a new applet using the list template.

   e. On the applet form, in the **Data Item** list, select **Open incidents**.

   f. In the field configurations section, add the following fields to the list header.
      - Number
      - Priority
      - Short description
      - Assigned to
• Assigned to > Avatar
• State

g. Click **Save**.

3. Add the action function to a swipe action for the Open incidents list.

   a. From the primary screen tab of your 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**.

**Results**

**Action function example**

© 2022 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.
Tutorial: Configure a navigation function with parameters

Follow this tutorial 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 the mobile application.

Creating a navigation function with parameters requires several steps. In this tutorial, you will do the following:

• 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 a mobile application using Guided Application Creator.

Create a data item with parameters

Create a data item containing your caller records.

Before you begin
Role required: admin

About this task

In these steps you create a data item that contains your caller records. The data item uses a parameter to filter the User [sys_user] list, so only the callers from your incident record appear.

Procedure

1. In Studio, navigate to Mobile Studio > Data Items.
2. Click the pop out icon ( ▼ ) to open the data items list in a tab.
3. In the Data Items tab, click Create New.
4. In the New Data Item form, fill in the fields.

Data item form

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Title for the data item. For this example, enter Caller</td>
</tr>
<tr>
<td>Field</td>
<td>Value</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Table</td>
<td>The name of the table you want the data item to pull information from. Select <strong>User [sys_user]</strong></td>
</tr>
<tr>
<td>Description</td>
<td>Optional additional details about the data item.</td>
</tr>
<tr>
<td>Condition Type</td>
<td>Whether the data item condition is declarative or scripted. Select <strong>Declarative</strong>.</td>
</tr>
</tbody>
</table>

5. In the query condition, add the following condition.
   - `<Active> <is> <true>`

6. Click **Save**.

7. In the Parameter Definition section, click the add icon (✚).
8. On the Parameter Definition form, fill in the fields.

### Parameter definition form

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the UI parameter. For this example, enter <strong>caller_sys_id</strong>.</td>
</tr>
<tr>
<td>Note:</td>
<td>You can have multiple parameters with the same name, so choose a name that you can distinguish easily.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of parameter. Select <strong>String</strong>.</td>
</tr>
</tbody>
</table>

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

10. In the Data Item tab, click **And** to add a new row to your condition builder.

11. From the Choose field list, select **Sys ID**.

12. In the same row as the Sys ID field in the condition builder, click the contextual reference value icon (🔍).

13. Add the item parameter you created.

14. Click **Save**.
Results

Caller data item

Create an applet to navigate to

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

About this task

In these steps you create the applet that your users will see when they use the navigation function.

Procedure

1. In Studio, navigate to Mobile Studio > Applets.
2. Click the pop out icon ( ) to open the ITSM application in a tab.
3. Click Create an applet.
4. Create a new applet using the list template.

New applet with the list template
5. In the Callers applet tab, from the Data Item list, select Caller.
6. Above the Item preview example, click Change Pattern.
7. In the List Item Patterns window, select pattern 9 in Pattern Set 1.

8. Click Done.
9. In the Field Configurations section, use the list to add the following fields to the list header.
   - Department
   - Name
   - Avatar
   - Title
10. Switch to the Form screen tab of the Callers applet.
11. Above the Selected Fields list, click the Replicate from primary button.
12. Click Save.
    A Parameter Setting section and sub sections appear under the Data item field.
13. In the User input parameters definition section, click the add icon.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the user input parameter. For this example, enter caller_sys_id.</td>
</tr>
<tr>
<td>Input Type</td>
<td>Type of input for the user input parameter. Select Text.</td>
</tr>
</tbody>
</table>

15. In the Screen UI Parameter Mapping section, click the add icon.
   - Data Item Parameter: The name of the parametrized data item 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, which should be caller_sys_id.
**Screen UI Parameter mapping form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item Parameter</td>
<td>Select the caller_sys_id item parameter created in the previous steps.</td>
</tr>
<tr>
<td>UI Parameter</td>
<td>Select the caller_sys_id UI parameter created in previous steps.</td>
</tr>
</tbody>
</table>

17. In the Screen UI Parameter mapping pop-up, click **Save**.

18. In the Callers applet, click **Save**.

**Completed caller applet form**

**Configure a navigation function with parameters**

After you create an applet to navigate to, create a navigation function.
Procedure

1. In Studio, in a mobile application, navigate to **Mobile Studio > Functions > Navigations**.

2. Click the pop out icon ( ![ ] ) to open the navigation function list in a tab.

3. In the **Navigations** tab, click **Create New**.

4. On the **New Navigation** form, fill in the fields.

### New navigation form

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for the navigation function. For this example, enter Navigate to caller</td>
</tr>
<tr>
<td>Destination Type</td>
<td>Whether the destination is an applet or applet launcher. Select <strong>Applet</strong>.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional description for the navigation function.</td>
</tr>
<tr>
<td>Destination</td>
<td>Destination for the application. Select the <strong>Callers</strong> applet you created in previous steps.</td>
</tr>
<tr>
<td>Context</td>
<td>Context for the navigation function. Select <strong>Record</strong>.</td>
</tr>
<tr>
<td>Table</td>
<td>Table for the navigation function. Select <strong>Incident [incident]</strong></td>
</tr>
</tbody>
</table>

5. Click **Save**.
   After saving, the **Parameter Setting** section appears on the form.

6. In the **Parameter Setting**, click the first row in the table.

### Parameter setting section

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>caller_sys_id</td>
<td>Field</td>
<td></td>
</tr>
</tbody>
</table>

7. From the Redirection Parameter table, fill in the fields as needed.

### Redirection parameter fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination UI parameter</td>
<td>Automatically filled with the name of the parameter created for the data item.</td>
</tr>
<tr>
<td>Type</td>
<td>Where the information for the parameter comes from. Since your users access this navigation from the <strong>Caller</strong> field on the incident form, select <strong>Field</strong>.</td>
</tr>
<tr>
<td>Source field</td>
<td>The field you want the user to navigate from. Select <strong>Caller</strong>.</td>
</tr>
</tbody>
</table>
8. Click Save.
9. In the main navigation tab, click Save.

**Associate the navigation function with an applet**
Now that you have a navigation function and destination applet, you need an incident applet to act as a starting point.

**About this task**
You can use any applet for these steps, including one from a previous tutorial. The first steps include instructions on creating an applet. Make sure that the Caller field is included as part of the Body on the details screen of the app. Otherwise, you do not have a field to assign the navigation to.

**Procedure**
1. Create a data item for open incidents.
   
   a. In Studio, navigate to Mobile Studio > Data Items.
   
   b. Click the pop out icon (opolitan icon) to open the data items list in a tab.
   
   c. In the Data Items tab, click Create New.
   
   d. In the New Data Item form, fill in the fields.

   **Data item form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Title for the data item. For this example, enter Open Incidents</td>
</tr>
<tr>
<td>Table</td>
<td>The name of the table you want the data item to pull information from. Select Incident [incident]</td>
</tr>
<tr>
<td>Description</td>
<td>Optional additional details about the data item.</td>
</tr>
<tr>
<td>Condition Type</td>
<td>Whether the data item condition is declarative or scripted. Select Declarative.</td>
</tr>
</tbody>
</table>

   e. In the query condition, add the following condition.
   
   - <State> <is one of> <New, In Progress, On Hold>
   
   f. Click Save.
2. Create an applet for open incidents.

   a. In Studio, navigate to **Mobile Studio > Applets**.

   b. Click the pop out icon (/sidebar) to open the applets tab.

   c. Click **Create an applet**.

   d. Create a new applet using the list template.

   e. On the applet form, in the **Data Item** list, select **Open incidents**.

   f. In the field configurations section, add the following fields to the list header:
      - Number
      - Priority
      - Short description
      - Assigned to
Field configuration for the list applet

3. From the Form Screen tab of the applet, select Body.
4. Use the list to add the Caller field to the applet.
5. Click Save.
6. In the Details tab, open the Functions section.
7. In the Field functions section, click the Add icon (+).
8. From the Field Functions window, in the Field list, select Caller.
9. In the Function list, select Navigation > Navigate to caller.
10. Click Save.

Results

Navigate to a caller example

Tutorial: Create a bar code scanning applet

Create an applet and parametrized data item that accepts a bad code scan as an input.

Using this example, you create a list applet that allows a user to scan a bar code or QR code for an asset.

Create a list applet and data item for bar code scanning

Begin by creating a list applet and a parametrized data item for your assets. The bar or QR code your users scan determine which assets appear in this list.
Before you begin
Role required: admin

Procedure
1. Navigate to All > System Applications > Studio to open Studio.
2. Select a scoped application where you want to create your applet.
3. In Application Explorer, navigate to Mobile Studio > Applets and select Applets.
4. Click the pop-out icon ( ) that appears to the right of Applets.
5. In the applet list, click Create an applet.
   The Create an Applet window appears on the screen.

   Create an Applet window
   Define the properties of the applet
   Name [ ] Icon [ ]
   Description [ ]
   Hide filter [ ]

   Choose a template
   List [ ] Form [ ]
   Employee Directory [ ] Map [ ]
   Group List [ ] Calendar [ ] URL [ ] Chart [ ]
   List [ ] Form [ ]
   Cancel [ ] Create new [ ]

6. In the Create an Applet window, enter a name for your applet, such as Asset QR Lookup, and select the List template.
7. Click Create New.
   The Create an Applet window closes. A new tab for your list applet appears in Studio.

8. In the Data item field, click the plus button ( ) to create a new data item.
9. In the New Data Item form, enter a name for your data item, such as Assets by QR.
10. In the Table field, select the Asset [alm_asset] table.
11. Click Save.
    After you click save, the Parameter definition section is visible.
12. In the Parameter definition section, click the plus button ( ) to create a new parameter.
13. In the Parameter definition pop-up window, enter Asset Tag in the Name field.
14. Click Save.
15. Under the heading All of these conditions must be met, select Asset Tag in the first list.
16. Click the reference value icon( ), and select the Asset Tag parameter you created in previous steps.

17. Click Save.

18. Return to the tab containing your list applet. In the Data Item field, select the data item you created in the previous steps.

19. Click Save.

20. In the Field Configurations section, select the pattern you want to use for your list.

21. Add the fields you want your users to see to the Selected Fields list.

Example of a pattern with selected fields. This example uses pattern 5 from pattern set 1.

22. Click Save.

Configure parameters for your list applet

Before you begin
Role required: admin
Procedure

1. In your list applet form in Studio, click the plus button (➕) in the **User Input Parameter Definition** section.

   ![User Input Parameter Definition](image)

   The **User Input** pop-up window displays.

2. In the **User Input** pop-up, enter **Asset Tag** in the **Name** field.

3. In the **Input type** field, select **QR/Barcode**.

4. Click **Save**.

5. In the **Screen UI Parameter Mapping** section, click the plus button (➕).

   ![Screen UI Parameter Mapping](image)

   The **Screen UI Parameter Mapping** pop-up window displays.

6. In the **Item parameter** field, select the **Asset Tag** parameter you created in the previous steps.

7. In the **UI parameter** field, select the **Asset Tag** parameter you created in the previous steps.

8. Click **Save**.

Results

After you add your applet to an applet launcher you can test the results. Upon opening the applet, you are prompted to scan a barcode. The list will return a list of records where the **Asset Tag** field matches the scanned in value.

Tutorial: Configure an applet to view knowledge articles within a ServiceNow mobile app

Use a modified form screen applet to enable your users to view, rate, and comment on knowledge articles from within your mobile apps.

Use this tutorial to create the mobile components needed to display knowledge articles within your mobile apps. Without this configuration, your knowledge articles are still available,
but they open in a browser rather than within your apps. After this tutorial, your mobile app can display knowledge articles as shown in this example.

The example screens shot show:

- List applet showing knowledge articles
- Form applet showing the article contents
- User interface in the Content tab form for rating the article
- User interface in the Comments tab for user feedback

![Knowledge Article Applet Screenshots]

Required plugin

To make use of the this tutorial. The ServiceNow NowMobile App - Knowledge Management Screens and Applet Launcher (com.glideapp.knowledge.mobile_requestor) plugin must be activated on your instance.

Note: If this plugin is not active on your instance, then will not appear on your plugin list. You can request that ServiceNow activate the plugin. For details on requesting plugin activation, see Request a plugin.

Configuration summary

To display knowledge records in a mobile app, you will need to configure these four components:

1. Data item
2. List screen item stream
3. List screen
4. List screen item stream segment

Create a data item for your knowledge records

Create the data item on the knowledge articles table to provide your app with the data it needs to display.

Before you begin

Role required: admin

Use these steps to create a data item for the Knowledge Article [sn_km_mr_st_kb_knowledge] table.
Note: The Knowledge Article [sn_km_mr_st_kb_knowledge] table is a new table installed with the ServiceNow NowMobile App - Knowledge Management Screens and Applet Launcher (com.glideapp.knowledge.mobile_requestor) plugin. This table contains records for each article in your Knowledge [kb_knowledge] table, and is automatically updated when you create new records on that table.

Procedure

1. In Studio, navigate to Mobile Studio > Data Items, and click the popout icon.
2. In the Data Items tab, click Create New.
3. Create a standard new data item using Knowledge Article [sn_km_mr_st_kb_knowledge] as the table. For more detail on this process, see Configure a standard data item.

Example:
This example shows a standard data item containing all records on the Knowledge Article [sn_km_mr_st_kb_knowledge]

Create an item stream for your list applet
Create an item stream to display the knowledge articles in your applet.

Before you begin
Role required: admin

Procedure

1. In the web-based UI, enter sys_sg_item_stream.list in the filter navigator to open the item streams list.
2. Click New.
3. In the item stream form, fill in the fields
**Item stream form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a name for your item stream, for example List of Knowledge articles Item Stream</td>
</tr>
<tr>
<td>Data item</td>
<td>Select the data item you created in the previous steps.</td>
</tr>
</tbody>
</table>

4. Right-click the form header and click **Save**.

5. In the **Item Stream M2M Item Configurations** list, double-click **Insert a new row...** under the **Item Configurations** field.

6. Select the **Knowledge articles Main Item** item configuration, then click the green check (✓) to save.

7. Click **Update**.

**Example:**

**Completed item stream form**

Create a knowledge article list applet

Create a list applet to display knowledge articles to your users.

**Before you begin**

Role required: admin

**Procedure**

1. In the web-based UI, enter `sys_sg_list_screen.list` in the filter navigator to open the list screens list.

2. Click **New**.

3. In the item list screen form, enter a name for your item stream, for example List of Knowledge articles.

4. In the **Icon** field, select an icon.

5. Right-click the form header and click **Save**.

6. In the **Item Stream Segments** list, double-click **Insert a new row...** under the **Name** field.

7. Enter Knowledge Item Stream Segment in the text pop-up, and click the green check (✓) to save.
8. Right-click the form header and click **Save** again.

9. In the **Item Stream Segments** list, click **Knowledge Item Stream Segment** to open the item stream record.

10. In the **Item Stream M2M Segments** list, double-click **Insert a new row**... under the **Item Stream** field.

11. Select the item stream you created in the previous steps, then click the green check (✔) to save.

12. Right-click the form header and click **Save** again.

**Results**

Your list applet has been successfully configured. Add your applet to an applet launcher to make it available in your mobile app.

**Design considerations for mobile apps**

Use mobile design and product principles to give your users the best mobile experience.

Before designing your mobile experience, take a look at the Building Mobile Apps with ServiceNow guide. The purpose of this document is to provide ServiceNow mobile app creators product principles and design considerations to create consumer grade mobile experiences.

Download the Building Mobile Apps guide at https://community.servicenow.com/community?id=community_article&sys_id=6daca508dbafc8905ed4a851ca961978

**Mobile hierarchy**

Learn the components of ServiceNow® mobile and how they work together to assist you in configuring, modifying, and creating applications.

**Components of the ServiceNow mobile framework**

This image represents the structure of the ServiceNow mobile framework. The next sections detail specific areas of the overall hierarchy, and descriptions of individual components.

ℹ️ **Note:**

The Mobile Onboarding app is being deprecated!

With the Now Platform Rome release in September 2021, we started phasing out support for the Mobile Onboarding. Customers may no longer activate it, and we are not offering enhancements or non-critical bug fixes. Mobile onboarding features are available in the **Now Mobile app for HR Service Delivery**.
A part of the ServiceNow® Mobile Platform, there are 2 mobile apps: the Now Mobile app and the Mobile Agent app. Each app focuses on a persona, which means the app experiences are tailored to support the tasks of specific roles in your organization. Each mobile app can have one or more mobile app configurations. For more information, see the following "Mobile app config" section.

**Mobile app config**

When users download a ServiceNow Mobile Platform application, they are prompted to log in with their credentials. When a user logs in to the instance, they are presented an experience that is created by a single mobile app config. The mobile app config that defines the user's experience depends on conditions defined in the config.

**Navigation bar**

Each mobile app config has a navigation bar that appears at the bottom of the screen. A navigation bar can have up to five icons, called navigation bar tabs. By default, navigation bars have a notification and settings navigation bar tab. For information about how to create a navigation bar, see [Configure the navigation bar](#).

**Navigation bar tabs**

Each tab in the navigation bar represents an applet or applet launcher. When you add more than five tabs to the navigation bar, a More tab appears. Tapping the More tab opens a list view showing additional tabs.

For information on how to create navigation bar tabs, see [Configure a launcher screen tab](#).

**Launcher screens**

Launcher screens serve as landing pages or home pages for your users. Launcher screens contain a configurable header, and sections to provide access to screens in several formats. You can also configure launcher screens with a search bar and quick actions, which give your users access to commonly used functions.

For more detail on launcher screens, see [Launcher screens](#).

**Screens**

Screens provide your users a method to view and modify data on your instance. Screens can display information as lists, maps, records, and other formats. You can find more detail on screen components in the next section.
screens

Screen segments
Screens contain one or more screen segments, which display information from your instances to your users. Screen segments represent the lists, calendars, maps, and records your users see within the app. If a screen has more than one screen segment, your users can switch between screen segments using a tabbed interface.

Screen segment records for lists, calendars, and maps are located on the screen segment [sys_sg_item_stream_segment] table. Screen segment records for record screens are on the record screen segment [sys_sg_form_segment] table.

Icons
Each screen has an icon. This icon represents the applet when it is displayed in a launcher screen or the navigation bar.

Icon records are located on the Icons [sys_sg_icon] table.

For more information on icons, see Mobile icons.

Item streams
An item stream is the source for the data shown in your screen. An item stream gets its data from a single source, called a data item. You can associate more than one item stream to a screen segment to include data from multiple tables. For example, you could create two item streams to display items from both the incident [incident] and request item [sc_req_item] tables in a single list.

Item streams are also associated with one or more item configurations. These item configurations provide a pattern controlling how the data appears in your screen segment.

Item stream records are located in the item stream [sys_sg_item_stream] table.

Data items
Data items provide the data presented in a screen. A data item is a dataset correlated with a table in an instance. A data item can include a filter condition to restrict what data the item returns. Associate data items with screens so that the screens can transform the dataset into human-readable information.

Data items are located on the data items [sys_sg_data_item] table.
For more detail on data items, see Data items.

Item configurations

Item configurations provide a pattern for data in your screen, and control how your data appears within a screen segment. For more detail on how an item configuration controls the appearance of your data, see the item configuration section.

Item configurations are located on the item configuration [sys_sg_master_item] table.

More information

For more information, See:

Screen segments

Screen types determine what an applet looks like and how your users are able to interact with it. You can create applets with these screen types:

- Calendar
- Chart
- Employee directory
- Form
• Grouped list
• List
• Map
• URL

In addition to these types, you can add the following screens to segments in your form applet:
• Details screen
• Activity stream screen
• Related list screen
• Embedded list screen

Functions

Your users can use functions to perform tasks in the mobile app such as assigning a task, or navigating between records. Actions can also interact with your mobile device to send emails, navigate using map software, or make a phone call.

Functions are located on the Function [sys_sg_button] table.

For more information on functions, see Mobile functions.

Mobile impersonation

Use mobile impersonation to impersonate authenticated users for testing purposes and view impersonation logs.
As an administrator or an impersonator, use impersonation to view a ServiceNow mobile app exactly as another user would see the app.

While impersonating another user, the app displays a banner at the top of the screen with the name of user being impersonated.

Impersonation logging and limitations for mobile are the same as those when using impersonation in the web-based UI. For details see Impersonate a user.

**Limitations of mobile impersonation**

In addition to the limitations listed in Impersonate a user, the following are not supported when using impersonation in ServiceNow mobile:

- Virtual Agent
- Offline mode.
- Security PIN

**Enable or disable impersonation in mobile apps**

Learn how to enable or disable impersonation in your ServiceNow mobile apps.

**Before you begin**

Role required: admin or impersonator
Note: An impersonator is a user who can impersonate all users except those with an admin role.

Procedure

1. In your ServiceNow mobile app, tap the Settings (⚙️) tab in the navigation bar.

   Note: If you do not see a Settings tab, tap More (≡) and then tap Settings from the list.

2. In the settings screen, tap the user card.

3. At the bottom on the screen, tap impersonate user.
4. Select a user to impersonate in the **Select User** list.
You are now impersonating the selected user. A banner appears at the top of the screen to indicate which user you are impersonating.
5. To end impersonation, return to the settings screen, tap the user card, and tap **End Impersonation**.

**Mobile App Builder**

The Mobile App Builder (MAB) is a configuration tool to build and manage screens and records that make up workflows within ServiceNow mobile apps. The organizational layout and navigation options in the Mobile App Builder facilitates a faster and more intuitive creation of ServiceNow mobile applications.

Introduction to Mobile App Builder

Benefits and uses of the Mobile App Builder include:
• Organization of records in a hierarchical layout for quick access and reference.
• Streamlining of mobile configurations by: allowing the editing of multiple records in a single screen, guiding users to appropriate record relationships, preventing invalid configuration choices, and auto-populating fields when creating relationships between records.
• Supports hierarchical tree navigation, enabling users to visualize how records relate to each other and tie into the overall mobile app configuration.
• Supports the editing of all ServiceNow base system mobile application scopes.

Mobile App Builder workflow

The building of mobile applications takes place within three screens. In the Mobile App Builder application scope selection screen, you select the application scope to work in. After choosing your application scope, select a category or specific record to create or edit in the Mobile App Builder categories home screen. The record you select then opens in the Mobile App Builder record screen. It is here where you perform most of your work, by populating record fields and creating relationships between different records.

Accessing the Mobile App Builder application scope selection screen

Access the Mobile App Builder to start the process of creating workflows for your mobile apps.

Before you begin
Role required: admin

Procedure

Navigate to All > System Mobile > Mobile App Builder.

Results
The Mobile App Builder application scope selection screen opens in a separate browser tab.

Note: The Mobile App Builder tour modal window displays when you first log in to the Mobile App Builder and each time you log in, if you don’t select the Don’t show me this again option. The tour is a visually helpful way to familiarize yourself how to build apps. You can always reopen the tour modal window using the help menu.

On the application scope selection screen the following options are available.

• Get information about Mobile App Builder and how to use the tool, by selecting Go to product documentation.
• Customize the appearance of the mobile cards you use within Mobile App Builder, by selecting Open Mobile Card Builder.
• Choose an existing application scope to access its mobile records. This list displays all application scopes that you are permitted to edit. Use the Search application scopes field or select the Name column header to find the application scope you need. Select an application scope name to open the Mobile App Builder categories home screen.
Mobile App Builder categories home screen

Use the Mobile App Builder categories home screen to select or create the most commonly used configuration options. Select category options like Screens or Cards & Icons to open records for you to create and edit.

The Mobile App Builder home screen contains of the following elements.

• Left menu panel which groups the most important mobile components into categories. These menu items are fixed and you cannot add or remove them from the list. Depending on your category selection, different record types are displayed in the main area of the page. Use the scroll bar or search option ( ) to find your exact record. Alternatively, to create records for your selected category, select the New button.

Note: The All mobile records menu item is a tool to quickly search and access record types familiar to you.

• A help menu ( ) with the following options.
  
  ◦ Record configuration tour - Provides an on-screen step-by-step guide demonstrating how to create records.

  Note: The Record configuration tour menu option is only available in the Mobile App Builder record screen.

  ◦ Mobile App Builder overview - Opens the Mobile App Builder tour modal window.

  ◦ App preview instructions - Opens a modal window with instructions to help users download mobile apps and log into their instances.
Go to product documentation - Opens the ServiceNow mobile documentation in a new browser.

Version - Displays the Mobile App Builder version for use during troubleshooting.

An option to change application scope displays at the top of the screen. Select the change scope option (Change scope) option to return to the application scope selection screen and select an alternative application scope to work in.

Note: To ensure that records are not saved in the wrong application scope, you cannot edit a cross-scope record if there are any unsaved changes.

Menu categories in Mobile App Builder

Mobile app configs

The Mobile app configs category is the configuration option for ServiceNow’s iOS and Android mobile applications. Select an option to personalize the app theme, set user access permissions, and create a navigation bar. You must select one configuration per app type. This category is based on the table sys_sg_native_client.

Note: You can open an existing Mobile Onboarding app configured from a previous release. However, you cannot create a new Mobile Onboarding app configuration.

You can either select the new button (New) to create a new mobile app configuration. Alternatively, select the name of a listed mobile app configuration to open the defined
Screens

Screens are the essence of the mobile app experience, enabling you to deliver unique and customized capabilities to provide different user experiences. Screens have different configurable components according to the screen type you select. This category is based on the table `sys_sg_screens`.

**Note:** Mobile App Builder supports launcher screens, list screens, record screens, input form screens, and mobile web screens.

You can either select the new button (New) to select a screen type to configure a new mobile screen. Alternatively, select the name of a listed screen to open the records required for that configuration, in the Mobile App Builder record screen.

Cards & icons

Cards consist of layouts that show visuals, text, and data. Icons are symbols that act as actions or shortcuts when connected to other screens. With both of these items, you can create rich, interactive mobile experiences. This category is based on the tables `sys_sg_icon`, `sys_sg_view_config`, `sys_sg_view_template`, and `sys_sg_item_view`.

You can either select the new button (New) to select a card type to configure. Alternatively, select the name of a listed card, card template, icon, or legacy card to open the records required for that configuration, in the Mobile App Builder record screen.
**Note:** You are able to open and edit an existing legacy card configured from a previous release. However, you cannot create a new legacy card.

---

### Functions

Use functions to define which actions users can perform within the mobile app. This category is based on the table `sys_sg_button`.

Either select the new button (New) to create a new function. Alternatively, select the name of a listed function type to open the records required for that configuration, in the Mobile App Builder record screen.

### Data

Data items provide the data presented in a screen. Select the table you want data from and define the conditions that must be met for the data to be displayed. This category is based on the table `sys_sg_data_item`.

You can either select the new button (New) to select a data item type. Alternatively, select the name of a listed data item to open the records required for that configuration, in the Mobile App Builder record screen.

### All mobile records

Use the **All mobile records** category to quickly find records within specific tables without the need to drill down into multiple layers of the Now Platform configuration tree. For example, a filter record can be selected here instead of having to select a list screen record and then drill down to the filter record.

Use the **Record type** field to search and select the table you require. The results display a list of records of the selected record type. Use the search option (Search) to filter the records from the list.
Mobile App Builder record screen

Use the Mobile App Builder record screen as the workspace where you perform all your mobile configurations.

The layout of the Mobile App Builder record screen provides instant access to any record connected to the root-level record, selected from the categories home screen. When selecting a record from the Now Platform configuration tree, the fields applicable to that record are displayed in the center configuration panel. As you select different records from the Now Platform configuration tree, the content in the configuration panel changes.

Now Platform configuration tree

The Now Platform configuration tree shows your selected record as the top-level record in a hierarchical display. Underneath your selected record are all the child records belonging to the parent record. You can instantly access and select any of the child records, at any level in the hierarchy. The content and the fields of any selected record displays in the configuration panel, where you view and edit these fields.

For large configuration trees the Mobile App Builder condenses screen records that are child records of launchers or mobile app configurations. This capability ensures that the configuration tree is a more manageable size. These partially loaded records are recognizable because they don’t have any child records displayed beneath them. Open and edit these floored records and any records associated with it by choosing the record and then selecting the Open record to edit button. Once selected, the Mobile App Builder displays the chosen floored record as the root-level record. You can edit this record and all its child records. To return to the parent record,
select the **Back to <parent record name>** button, located next to the record name in the tool bar.

As you work on records, indicators display on the configuration tree. A red indicator (●) shows that there are required fields that must be populated, before saving your changes. An orange indicator (●) shows that records have been edited but not yet saved.

**Note:** The red and orange indicators are displayed on the right side of the hierarchical tree. This means that for a hierarchical tree with many levels of child records, the indicators may not be immediately visible. Either use the horizontal and vertical scroll bars. Alternatively, use the resize panel controller, which exists between the Now Platform configuration tree panel and the configuration panel.

---

### Configuration panel

The configuration panel is the main work area of the Mobile App Builder where you perform the editing of your records. When you select a record, field types that are specific to the selected record open in the configuration panel. For a list of all the different field types and their usage, see Record screen input field types. Not all fields require input or editing and only the fields marked with an asterisk (*) are mandatory.
Mobile interface example panel

The mobile interface example panel, helps you identify the UI element you are working on within your mobile device. Some records may include multiple UI elements. To view these UI elements, select the relevant tab in the example panel.

UI elements of the record screen

The Mobile App Builder record screen contains the following UI elements.

<table>
<thead>
<tr>
<th>UI element</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Home button" /></td>
<td>The home button, which you can use at any time to return to the categories screen.</td>
</tr>
<tr>
<td><img src="image" alt="Asset Look-up" /></td>
<td>A record indicator that displays the top-level record you are working on.</td>
</tr>
<tr>
<td><img src="image" alt="Application scope" /></td>
<td>An application scope indicator that displays the application scope you are working in.</td>
</tr>
<tr>
<td><img src="image" alt="Quick-reference button" /></td>
<td>A quick-reference button that opens the mobile documentation in a separate tab.</td>
</tr>
</tbody>
</table>
The save button is highlighted to show if there are any unsaved record changes anywhere in the record hierarchy. If no record changes exist, the button is inactive.

The more button, which provides the following options:
- The table name of the selected record, for example, Record screen [sys_sg_form_screen].
- **Delete record** - Select this option to delete the displayed record.
- **Open in platform** - Opens the record you are working on in the web-based UI.

**Note:** All screens in the Mobile App Builder are exact reflections of screens in the web-based UI.

### Record screen input field types
Familiarize yourself with different field types and their usage in the Mobile App Builder record screen.

When either creating a new record or editing an existing record, you are presented with field types specific to the record you selected. You cannot add additional field types, as this ensures that you only work with field types relevant to the open record. The following are all the available record screen input type fields.

<table>
<thead>
<tr>
<th>Input field type</th>
<th>Function</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text input</td>
<td>Text field which can contain up to 40 characters.</td>
<td></td>
</tr>
<tr>
<td>Text area input</td>
<td>Text area which can contain an unlimited number of characters.</td>
<td></td>
</tr>
<tr>
<td>Toggle switch</td>
<td>Switch to enable or disable the functionality.</td>
<td></td>
</tr>
<tr>
<td>Option button</td>
<td>Select a single option from the two or more displayed options.</td>
<td></td>
</tr>
<tr>
<td>Menu list</td>
<td>Select the down arrow from the menu to display a list of choices and to select a single item.</td>
<td></td>
</tr>
<tr>
<td>List</td>
<td>Add list entries and define a value for each of them.</td>
<td></td>
</tr>
<tr>
<td>Condition type</td>
<td>Set the condition type and the condition definitions for the selected record</td>
<td></td>
</tr>
<tr>
<td>Record screen table</td>
<td>Tables consist of either a single reference record or multi-reference record. Depending on the predefined setup of the table you can either choose new records, edit existing records, or perform both these actions. For information on working with these tables, see Record screen tables.</td>
<td></td>
</tr>
</tbody>
</table>

### Record screen tables
Use tables in the Mobile App Builder record screen to reference child records to a parent record. These records can consist of either single reference records or multi-reference records.
Tables in a record screen are one of a few input field types. For a complete list, see Record screen input field types. Tables contain reference records where you can select or create a child record for the parent record to reference. The option to select or create a reference record is predefined according to the record you select.

### Launcher sections

<table>
<thead>
<tr>
<th>Name</th>
<th>Order</th>
<th>Application scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recently Viewed</td>
<td>10</td>
<td>Field Service Mobile</td>
</tr>
<tr>
<td>Popular</td>
<td>20</td>
<td>Field Service Mobile</td>
</tr>
<tr>
<td>Browse</td>
<td>30</td>
<td>Field Service Mobile</td>
</tr>
</tbody>
</table>

### Usage of Choose and New buttons in record screen tables

Each table within a record screen table displays either a **Choose** button (Choose), a **New** button (+ New), or both buttons (Choose + New).

- In a situation where you can only reference existing records, the **Choose** button displays.
- In a situation when you can only create a new reference to the record you are working on, the **New** button displays.

If the **Choose** or **New** button is inactive after you make a single selection, it means that only a single record can be referenced. This is known as a single-reference record. Alternatively, when more than one selection can be made, this is a multi-reference record.

To change a single-reference record, first remove the existing reference by selecting the delete row button (Delete), then either choose or create a new reference.

For more information about working with these options, see Creating reference record entries in record screen tables and Editing record reference entries in record screen tables.

### UI elements in a table reference record

The Mobile App Builder table reference records contain the following UI elements.

<table>
<thead>
<tr>
<th>UI element</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actionable items in a table reference record are blue. These items can be record names which open the record, remove buttons, and fields for inline editing.</td>
<td></td>
</tr>
<tr>
<td>The column sort button is available within the header row of all tables in the Mobile App Builder. Select the column sort button ( ▲ ) to list the column contents in either ascending or descending order. The black triangle indicates the sorting order for the entire table.</td>
<td></td>
</tr>
<tr>
<td>You can perform inline editing in certain fields, such as the <strong>Order</strong> field. When performing inline editing, you provide content within the table and you do not need to open the record.</td>
<td></td>
</tr>
<tr>
<td>Use the remove button to remove an entry from the record. It does not delete the entry from the database.</td>
<td></td>
</tr>
</tbody>
</table>
Related information
  Creating reference record entries in record screen tables
  Editing record reference entries in record screen tables

Actions and processes within the Mobile App Builder

Familiarize yourself with the different processes to help you navigate and use the Mobile App Builder.

Creating new and selecting existing records from the categories home screen

Select records from the Mobile App Builder categories home screen for you to either edit or create from new.

Before you begin
Role required: admin

Procedure
1. Navigate to All > System Mobile > Mobile App Builder.
2. Search and select the name of your application scope in the Mobile App Builder application scope selection screen.
3. Select a category from the categories home screen and do one of the following actions:
   • Select the New button to create a new record of the selected category.
   • Select a record name listed in the main area of categories home screen, to open and edit an existing record.

Results
By either selecting the New button or selecting an existing record the Mobile App Builder record screen opens. If you select New, the record screen displays a record without any of the fields completed. Whereas if you select an existing record, predefined data displays for you to edit.

Creating reference record entries in record screen tables

Create new entries in Mobile App Builder’s record screen tables and view where the Now Mobile configuration tree displays the new entry in its hierarchy.

Before you begin
Role required: admin

Procedure
1. Navigate to All > System Mobile > Mobile App Builder.
2. Search and select the name of your application scope in the Mobile App Builder application scope selection screen.
3. Select a category from the categories home screen.
4. Select to either open a new record, by selecting the New button or edit an existing record, by selecting on a record’s name.
5. In the record screen, where available in a table, select the New button to open a empty reference record in the configuration panel.
The Now Platform configuration tree automatically updates with the new record you created.

6. Complete the fields as required and select **Save**.

**Editing record reference entries in record screen tables**

Learn to edit existing entries in Mobile App Builder’s record screen tables.

**Before you begin**
Role required: admin

**Procedure**

1. Navigate to **All > System Mobile > Mobile App Builder**.

2. Search and select the name of your application scope in the **Mobile App Builder** application scope selection screen.

3. Select a category from the categories home screen.

4. Select to either open a new record, by selecting the **New** button (New) or edit an existing record, by selecting on a record’s name.

5. In the record screen, where available in a table, select the **Choose** button (Choose) to open a list of reference records within all the application scopes you have access to.

**Note:** You can use records from other scopes, however you are not able to create a new record in an alternative application scope.
6. Select a record name and then select the **Apply** button (Apply). The record screen opens with the predefined data.

7. Edit the fields as required and select **Save**.

**Saving changes in Mobile App Builder**

Save all the changes you make within the Mobile App Builder with a single action. There is no need to save each change individually before moving onto another screen or record.

**Before you begin**

Role required: admin

**About this task**

**Procedure**

1. Navigate to All > System Mobile > Mobile App Builder.

2. Search and select the name of your application scope in the Mobile App Builder application scope selection screen.

3. Make the necessary changes in the categories and their records.

4. Select the Save button (Save), to save all your changes to the database all at once.

**Note:** Changes that you make are not saved to the database until you select the **Save** button. Mobile App Builder has several warning modals that display if you attempt to navigate away from a record with unsaved changes. For example, if you select the **Home** button or back browser button, a warning message displays about unsaved changes.
Deleting records in Mobile App Builder

For ease-of-use, the delete button in Mobile App Builder exists in the same location throughout the tool. When deleting a record all child records are also removed from the configuration tree, but not deleted from the database.

Before you begin
Role required: admin

Procedure
1. Navigate to All > System Mobile > Mobile App Builder.
2. Search and select the name of your application scope in the Mobile App Builder application scope selection screen.
3. Find the record you want to delete by either:
   • Using the All mobile records category and searching for your record and then selecting the name of the record.
   • Drilling down into the category and record type until you find your record.
4. Select the more button ( )
5. Select Delete record and then confirm that you want to delete the record.

Note: After confirming the delete action, the record is no longer displayed but still remains in the database.

Accessing Mobile Card Builder within Mobile App Builder

Navigate between the Mobile App Builder and the Mobile Card Builder to access and edit personalized card and card templates. The Mobile Card Builder opens in a separate tab in your browser.

Before you begin
Role required: admin

About this task
When using Mobile App Builder you can access the Mobile Card Builder to edit cards and card templates using the graphical interface.

Note: You cannot open a record in Mobile Card Builder if the record you are trying to open has unsaved changes in Mobile App Builder.

Procedure
1. Navigate to All > System Mobile > Mobile App Builder.
2. Access the Mobile Card Builder within the Mobile App Builder in one of the following ways.

<table>
<thead>
<tr>
<th>Location</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>From the Mobile App Builder home page</td>
<td>• Select Open Mobile Card Builder from the home page.</td>
</tr>
<tr>
<td>Location</td>
<td>Action</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>a. From the home page, select a scoped application from the <strong>Application scopes</strong> area.</td>
<td></td>
</tr>
<tr>
<td>b. Select <strong>Cards and icons</strong> from the configuration tree.</td>
<td></td>
</tr>
<tr>
<td>c. Select <strong>New</strong> to create a new card, card template, or icon, then select <strong>Continue</strong>.</td>
<td></td>
</tr>
<tr>
<td>The <strong>Mobile App Builder</strong> configuration panel opens, displaying the card or icon option you selected.</td>
<td></td>
</tr>
<tr>
<td>d. Select the button <strong>Open in Mobile Card Builder</strong>, to customize your card or icon.</td>
<td></td>
</tr>
<tr>
<td>• Select the <strong>Open in Mobile Card Builder</strong> button ( ), which is displayed in the configuration panel for card, card template, icon, or legacy card records.</td>
<td></td>
</tr>
</tbody>
</table>

**Mobile Card Builder**

Use Mobile Card Builder to create or modify card templates using a graphical interface.

An introduction video to the Mobile Card Builder and its capabilities.

**Customize mobile card views**

**Examples of cards in the Mobile Agent app.**

Mobile Card Builder is a visual tool you can use to edit the templates and cards used in the ServiceNow mobile applications for iOS and Android. The ServiceNow mobile interface uses cards to display information about records on your instance. The following places make use of cards to display record information:
• Launcher screen record sections
• Calendar screens
• Form screens
• Genius search results
• List screens
• Map screens

In versions up to and including Paris, mobile cards existed as Item views, using records on the legacy card [sys_sg_item_view] table. In Quebec and later releases, mobile cards created in and and Studio are created as Mobile views using records on the card template [sys_sg_view_templates] and card [sys_sg_view_config] tables. Elements created using legacy card records will continue to work in Quebec and later releases, however, cards created in this release use the new mobile view tables.

 Depending on whether a mobile card is based on a legacy card, card, or a card template, different options are available when using Mobile Card Builder. These differences are stated in the relevant areas of the documentation.

**Mobile Card Builder open screen**

The Mobile Card Builder open screen appears when you first open the card builder. Use this screen to select what you want to work on.

The available cards are divided into sections on the left side of the screen. The right side of the screen displays images representing the cards in the selected section.

**Cards**

Use the **Cards** option to modify an existing card design. Use the **Application** and **Applet** lists to find the cards associated with a specific application and applet.

**Templates**
Use the **Templates** option to build a card using an existing template. Below each template image is the templates name, short description, and author. You can also click **Create blank template** to start with an empty template.

**Default templates**

Use the **Default templates** option to select from templates created by ServiceNow.

**Custom templates**

Use the **Custom templates** option to select from templates you have created or customized from the default templates.

**My templates**

Use the **My templates** option to select from the templates that you have created.

Use the **Open** button in the lower right corner to open the selected card template.

**Mobile Card Builder stage**

After you open a card template or choose to create a new card template, the Mobile Card Builder stage displays. Use this interface to create and update your cards.

**Mobile Card Builder user interface**

Learn about the Mobile Card Builder user interface to get started building and modifying your mobile cards.

© 2022 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.
Top panel

The top panel of the Mobile Card Builder contains the following information and options.

**Menu**

Use the menu to access additional options.

- Use **Open** to close the current card and open another card.
- Use **Edit card properties** to update the name and short description of your card.
- Use **Change template** to replace the current card layout with one in a card template.

**Note:** Card templates can only be applied to cards created in the Quebec and later releases.

- Use **Reset stage** to remove all elements from the current mobile card.

**Height and width**

This section displays the height and width of your card in pixels. This information automatically updates as you add and remove components from your card.

**Undo**
Select to undo the last update you made to your card. You can also enter Ctrl +Z or ⌘+Z on your keyboard to perform this action.

Redo
Select to redo the last update removed by the Undo command. You can also enter Ctrl+Shift+Z or ⌘+Shift+Z on your keyboard to perform this action.

Save
Select save to save your work.

Left Panel

<table>
<thead>
<tr>
<th>Card</th>
<th>The left panel of the card builder contains the following information and options.</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Knowledge Category View" /></td>
<td><strong>Card section</strong>&lt;br&gt;The topmost section of the left panel shows the name of the current card.</td>
</tr>
<tr>
<td><img src="image" alt="Knowledge Category View" /></td>
<td><strong>Component tree</strong>&lt;br&gt;Below the card section you can see the components of your card organized in a tree structure. Select a component to highlight that component in the center panel. When you select a component, that component's configuration options appear in the right panel.</td>
</tr>
</tbody>
</table>

Center Panel
Use the center panel to design the layout of your card. The panel displays the components of your card. To view the card as it appears in a mobile app, Select the Preview button.
Components in the center panel

Select a component of your card to highlight it. You can see a tab with the name of your component, and a trash can icon. Select that icon to remove the component from your card. More configuration options for the selected component appear in the right panel.

Adding a component

Select a plus icon (++) to select a new component to add to your mobile card. For information about adding a button, see Card actions in Mobile Card Builder.
Remove components by selecting them and then select the delete icon.

**Right Panel**

The right panel of the card builder contains the configuration options for the selected component. The available configuration options depend on the type of component selected.

When you change a component's configuration, you can see the changes immediately in the center panel. If you are using the preview option, select **Update preview** to update the preview of your component.

Configuration options include formatting options such as:

- Size
- Margins
- Corner radius
- Text alignment

You can also use this panel to map field values to components in your card. For details see [Customize a screen using Mobile Card Builder](#).

**Component configuration**

<table>
<thead>
<tr>
<th>Field type</th>
<th>Field value</th>
<th>Field label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mapped field</td>
<td>name</td>
<td></td>
</tr>
<tr>
<td>Height</td>
<td>auto</td>
<td></td>
</tr>
<tr>
<td>Width</td>
<td>auto</td>
<td></td>
</tr>
<tr>
<td>Margin top</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Margin right</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Margin bottom</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Margin left</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Text styles</td>
<td>Custom</td>
<td></td>
</tr>
<tr>
<td>Appearance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text alignment</td>
<td>Left</td>
<td></td>
</tr>
<tr>
<td>Max number of lines</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**Mobile Card Builder components**

Learn about the components that make up mobile cards
Adding and removing components

Containers

Containers components contain other components. A container can contain any card component, including another container. Use containers to organize your card components, and determine how the components within the container are aligned and distributed.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>Height of your container in pixels</td>
</tr>
<tr>
<td>Width</td>
<td>Width of your container in pixels</td>
</tr>
<tr>
<td>Margin top</td>
<td>The top margin of your container in pixels</td>
</tr>
<tr>
<td>Margin bottom</td>
<td>The bottom margin of your container in pixels</td>
</tr>
<tr>
<td>Margin left</td>
<td>The left margin of your container in pixels</td>
</tr>
<tr>
<td>Margin right</td>
<td>The right margin of your container in pixels</td>
</tr>
<tr>
<td>Corner radius</td>
<td>The corner radius of your container. Increase this value to create rounded corners for your container.</td>
</tr>
<tr>
<td>Container direction</td>
<td>Determines orientation direction of the components within the container. Select from these values:</td>
</tr>
<tr>
<td></td>
<td>• Horizontal</td>
</tr>
<tr>
<td></td>
<td>• Vertical</td>
</tr>
<tr>
<td>Alignment</td>
<td>Determines the alignment of the components within the container. Select from these values:</td>
</tr>
<tr>
<td></td>
<td>• Stretch</td>
</tr>
<tr>
<td></td>
<td>• Center</td>
</tr>
<tr>
<td></td>
<td>• Left</td>
</tr>
<tr>
<td></td>
<td>• Right</td>
</tr>
<tr>
<td></td>
<td>• Top</td>
</tr>
<tr>
<td></td>
<td>• Bottom</td>
</tr>
<tr>
<td>Distribution</td>
<td>Determines the distribution of the components within the container. Select from these values:</td>
</tr>
<tr>
<td></td>
<td>• Equal</td>
</tr>
<tr>
<td></td>
<td>• Auto</td>
</tr>
</tbody>
</table>

Images

Image components are used to display an image. You define an image to display by selecting a field on your record that contains an image, for example, the Avatar field on a User record.
### Image configuration properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| Mapped field | The field on your record containing the image.  
> **Note:** When creating a card template, you will not be able to select a field. This selection is done when your card is associated with a specific screen. |
| Height | Height of your image in pixels |
| Width | Width of your image in pixels |
| Corner radius | The corner radius of your image. Increase this value to create rounded corners for your image. |
| Top margin | The top margin of your image in pixels |
| Right margin | The right margin of your image in pixels |
| Bottom margin | The bottom margin of your image in pixels |
| Left margin | The left margin of your image in pixels |
| Scaling | The scaling for your image. Select from these values:  
• Fill  
• Fit |

### Text

Text components display text on your card. This can be static text, or text from the label of value of a field. Use text components to the information your users need to see first when viewing lists of records.

### Text configuration properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| Field type | The type of text field. Select from these values:  
• Field value  
• Field label  
• Static text |
| Field type text | The text that appears in the text component. This value is only used when you have selected Static text in the Field type property. |
| Mapped field | The field containing the text to be used. The text component will display either the label of the field or the value of the field, |
## Text configuration properties (continued)

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>Height of your text component in pixels</td>
</tr>
<tr>
<td>Width</td>
<td>Width of your text component in pixels</td>
</tr>
<tr>
<td>Corner radius</td>
<td>The corner radius of your text component. Increase this value to create rounded corners for your text component.</td>
</tr>
<tr>
<td>Top margin</td>
<td>The top margin of your text component in pixels</td>
</tr>
<tr>
<td>Right margin</td>
<td>The right margin of your text component in pixels</td>
</tr>
<tr>
<td>Bottom margin</td>
<td>The bottom margin of your text component in pixels</td>
</tr>
<tr>
<td>Left margin</td>
<td>The left margin of your text component in pixels</td>
</tr>
<tr>
<td>Text alignment</td>
<td>Alignment for the text in your component. Select from these values:</td>
</tr>
<tr>
<td></td>
<td>• Left</td>
</tr>
<tr>
<td></td>
<td>• Center</td>
</tr>
<tr>
<td></td>
<td>• Right</td>
</tr>
<tr>
<td>Max number of lines</td>
<td>The maximum number of lines for the text in your element.</td>
</tr>
<tr>
<td>Font weight</td>
<td>The font weight for your text. Select from these values:</td>
</tr>
<tr>
<td></td>
<td>• Regular</td>
</tr>
<tr>
<td></td>
<td>• Bold</td>
</tr>
<tr>
<td>Font size</td>
<td>The font size for your text. Enter a numerical value.</td>
</tr>
<tr>
<td>Text color</td>
<td>The color of your text. The value for this property is an HTML color code, for example #A0522D for sienna.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You cannot change the background color of a text component, but you can place your text component within a container component, and change the container background color.</td>
</tr>
</tbody>
</table>

Create a card template with Mobile Card Builder

Use Mobile Card Builder to create a card template to use for cards in your mobile screens.

**Before you begin**

Role required: admin

**About this task**

**Note:** Use these steps to create a new card template from scratch, and then associate the new card template to an existing screen. Card templates can only be applied to screens created in Quebec of later releases using mobile card views.

**Procedure**

1. Navigate to **All > System Mobile > Mobile Card Builder**.
2. In the open screen pop-up, select **Create blank template** button in the upper right corner. The **Template properties** pop-up window displays.
3. In the **Name** field, enter a name for your card template.
4. In the **Short description** field, enter a description of your card template. Use your short description to indicate the type of record best suited to your card template, for example, incidents, or user records.
5. Select **Save**. The Mobile Card Builder opens, displaying an empty mobile card.
6. Add elements to your card template using the card builder interface. For details on using the card builder UI, see Mobile Card Builder user interface.

7. Open the screen where you want to apply your card template. You can do this using two methods:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open a screen card from Studio</td>
<td>In Studio, open the screen where you want to apply a card template, then select the Customize in Mobile Card Builder.</td>
</tr>
<tr>
<td>Open Card builder from the web-based UI</td>
<td>In Card builder, select Menu &gt; Open. Use the Application and Applet lists to select your screen (applet), then select the Open button in the bottom-right corner of the window.</td>
</tr>
</tbody>
</table>

8. To apply your card template to the screen, select Menu > Change Template.

9. If you want to assign field values to your card, you can do that at this point. For details on that process, see Assign field values to your mobile card.

10. Select Save.

**Customize a screen using Mobile Card Builder**

Use Mobile Card Builder to modify the cards for your screens.

**Before you begin**

Role required: admin
Procedure

1. Navigate to All > System Mobile > Mobile Card Builder.

2. In the open screen, use the Application list to select the application containing your screen.

3. Use the Applet list to select your screen (applet).

4. Select the card where you want to apply a card template.

   Note: A screen may contain more than one card view. For example, a list screen may have an legacy card for the list and another for the form.

5. Select Open button.
   Card builder opens using its current card template.

   Note: If you have your screen open in Mobile App Builder, you can also open the template by selecting the Mobile Card Builder button.

What to do next
To change the layout of the elements of your mobile card, see Mobile Card Builder user interface.

When you are happy with the appearance of your card, and want to add values from your records, see Assign field values to your mobile card.

Assign field values to your mobile card
Use Mobile Card Builder to show field labels and values from your tables in mobile cards.

Before you begin
Role required: admin

About this task
Data from your records is displayed on your cards using text elements. These elements can display either the label of a field or the value of a field from a record.
Text elements used to display data from an incident record

In this example, the top three text elements display the values of the **State**, **Number**, and **Short Description** fields. Under these elements, you can see pairs of text elements to display both the label and value of the **Opened by** and **Priority** fields. The dotted outlines around these fields are containers used to organize the text.

**Procedure**

1. Open your mobile card in Card Builder using one of the following methods:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Mobile Card Builder from Mobile App Builder</td>
<td>In Mobile App Builder, open the screen where you want to modify your card, and select open in option.</td>
</tr>
<tr>
<td>Open Mobile Card Builder from a Mobile Studio applet.</td>
<td>In Mobile Studio open the screen (applet) where you want to modify your card and select the Customize in Mobile Card Builder option.</td>
</tr>
<tr>
<td>Open Card builder from the web-based UI</td>
<td>Navigate to <strong>System Mobile &gt; Mobile Card Builder</strong>, then select the card or card template you want to modify using the pop-up window that displays when Card Builder first opens.</td>
</tr>
</tbody>
</table>

2. Select a text in your mobile card, or create a new one.
   For details on creating elements, see [Mobile Card Builder user interface](#).
   With your text element selected, you can see the configuration properties in the **Component configuration** panel on the right of the screen. If you do not see this panel, select the **Expand configuration panel** button.

3. Under **Field type**, select either an option.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field Value</td>
<td>The text element displays the value of a field on your table. For example, if you select <strong>Number</strong>, the field displays the record number.</td>
</tr>
<tr>
<td>Field Label</td>
<td>The text element displays the name of a field on your table. For example, if you select <strong>Number</strong>, the field displays the word <strong>Number</strong>.</td>
</tr>
</tbody>
</table>

4. Under **Mapped field**, select the list button ( ) to select a field from your table.

5. Use the **Select a field** window to select a field. When you have chosen your field, select **Select**.

![Select a field](image)

6. Repeat steps 2 through 5 to assign any other values from your table that you want to appear on your card.
   
   If you want to display both the name and value of the field, use two text elements side by side, as shown in this example.
7. Select Save.

Understanding mobile cards and legacy cards

Learn the differences between cards and legacy cards, and how to work with each type in Mobile Card Builder.

View configuration-based cards

Screens created in Quebec and later releases have mobile cards based on view configurations. These cards have greater flexibility, and can take advantage of card templates. More details on this type are given in the next section.

Item view-based cards

Mobile applets created in Paris and earlier releases have mobile cards based on item views. These legacy cards can still be edited using Mobile Card Builder, but have some limitations, as described in the next section.

Editing cards
Cards take advantage of templates to make creating your mobile cards faster and easier.

Editing card templates

Card templates define the layout of the fields on your mobile card, as well as any display properties that affect how your card looks. To change the look or appearance of your card, you must edit its template rather than the card itself. The options to add or remove elements and select containers are not available when editing cards. For detail on templates, see Create a card template with Mobile Card Builder.

Editing cards

After you have assigned a template to your card, you can edit the card in Mobile Card Builder. Here you define what information to display in the fields laid out in your selected template. For detail on this process, see Assign field values to your mobile card.

Editing legacy cards

Legacy cards can be edited as described in Mobile Card Builder user interface, with the following exceptions:

Templates are not available
You cannot assign a card template to legacy cards. Templates that use cards, and are not compatible with legacy cards. This option appears disabled in the menu for this card type.

**Reset stage option is not available**

The Reset stage option, which removes all elements from the current mobile card is not available for legacy cards. This option appears disabled in the menu for this card type.

Legacy cards do not have a separate template to modify. You are able to modify containers, add or remove fields, and make other changes to the card appearance directly in the mobile card rather than in a separate template record.

**Card actions in Mobile Card Builder**

Design and add actions to cards that you create within the Mobile Card Builder. You can use buttons to enable your users to perform actions directly within a card. For example, to accept or reject suggestions, call listed contacts, and to add items to a cart.

You can configure card actions to provide a highly visible interface for actions your users need to take quickly. These actions appear within cards or action lists, and are easier for users find than swipe actions or actions in the menu.

Users interact with card actions by tapping buttons within a card or action list. Users can perform an action right from a card, such as an approval. You can also display a menu with to give the user a choice of multiple actions.

Configure a card action using Mobile Card Builder

Use Mobile Card Builder to add card actions to a mobile card.

**Before you begin**

Role required: admin

**About this task**

When creating mobile cards, you may want to add buttons to enable your users to perform various actions. Usually the process of configuring card actions occurs when creating a card template within the Mobile Card Builder. This topic details the process of configuring card actions as a standalone procedure.
Procedure

1. Navigate to All > System Mobile > Mobile Card Builder.

2. In the Mobile Card Builder menu, select Templates.

3. Select the card template where you want to add a card action and select Open, or create a new card template using the Create blank template button.

4. Select a plus icon to select a new component to add to your mobile card.

5. Select Button from the displayed components. The Component configuration menu displays options related to button configuration.

6. Select the display options to define the appearance of your buttons. Button configuration options include:
   - The height and width of a button
   - The addition of a border and rounded edges
   - The size of margins to provide padding between other components

7. Select the format and label properties of the button. The available options are:
   - Label - A button with text only, for example an OK button. Enter the text required for the button.
   - Icon - A button with an icon only, for example a trash icon (трASH) to denote a delete button. Select an icon from the menu.
   - Label and Icon - A button with text and an icon, for example an edit button (edit). Enter the desired text and select the image from the fields.

8. Select if you want the button to contain a single action or multiple actions. The actions listed in the functions menu come from the Function [sys_sg_button] table.
• Single - Search in the function menu to allocate a single action to a button.
• Multi - Creates a single button which, when tapped, reveals additional card actions in a menu at the bottom of the screen. Select Add Menu Item and from the displayed menu add as many card actions as required. Select Finish after you configure each card action.

Drag the order icon ( ) to change the order of the card actions. You can also edit and delete the card actions at any point.

9. Define the appearance of buttons, in terms of its background color, text color, and border color. You can define the color by either:

• Manually entering the hexadecimal color value.
• Selecting an appearance palette and then select either the Swatches or List tab at the top of the menu.
Note: For icons, black is the only color available for the base record. Select a text color input to change the icon color from black to the selected color.

10. Select Save.

Card template element attributes
Use this reference to determine which attributes to use for your card template elements.

Each card template element has one or more required attributes that you must create. The sections below show the required attributes for each card template element type.

Note: When using the Mobile Card Builder, the listed attributes do not need to be manually created, as the tool automatically creates the attributes in the background.

Attributes for the Text type

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>TextValue</td>
<td>Used to display static text.</td>
<td>Use the Translatable value field to enter the text to display.</td>
</tr>
<tr>
<td>FieldLabel</td>
<td>Used to display the label of a table field.</td>
<td>Use the Value field to enter the name of a field. For example, short_description.</td>
</tr>
<tr>
<td>FieldValue</td>
<td>Used to display the value of a field in a record.</td>
<td>Use the Value field to enter the name of a field. For example, short_description.</td>
</tr>
<tr>
<td>FieldFunction</td>
<td>Used to display either a count of records, or the value by which records are grouped.</td>
<td></td>
</tr>
</tbody>
</table>
### Attributes for the Image type

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageURL</td>
<td>Used to display a static image from the attachment[sys_attachment] table on your instance.</td>
<td>Use the <strong>Value</strong> field to enter a SysID for a record in the attachment[sys_attachment] table.</td>
</tr>
<tr>
<td>FieldValue</td>
<td>Used to display an image from a field in a record.</td>
<td>Use the <strong>Value</strong> field to enter the name of a field. For example, avatar.</td>
</tr>
</tbody>
</table>

### Attributes for the Button and MenuButton types

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Button</td>
<td>Used to select the function that runs when users tap a button.</td>
<td>Use the <strong>Value</strong> field to enter the SysID of a function from the Functions[sys_sg_button] table. For details on creating functions, see <a href="#">Mobile functions</a>.</td>
</tr>
<tr>
<td>TextValue</td>
<td>Used to display static text.</td>
<td>Use the <strong>Value</strong> field to enter the name of a field. For example, short_description.</td>
</tr>
<tr>
<td>FieldValue</td>
<td>Used to display the value of a field in a record.</td>
<td>Use the <strong>Value</strong> field to enter the name of a field. For example, avatar.</td>
</tr>
<tr>
<td>Icon</td>
<td>Used to display an icon.</td>
<td>Use the <strong>Value</strong> field to enter the SysID of an icon record from the Icon[sys_sg_icon] table.</td>
</tr>
</tbody>
</table>

### Attributes for the Menu type

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>TextValue</td>
<td>Used to display static text.</td>
<td>Use the <strong>Value</strong> field to enter the name of a field. For example, short_description.</td>
</tr>
<tr>
<td>Icon</td>
<td>Used to display an icon.</td>
<td>Use the <strong>Value</strong> field to enter the SysID of an icon record from the Icon[sys_sg_icon] table.</td>
</tr>
</tbody>
</table>
Attributes for the HtmlText type

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>SearchExternalSource</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HtmlReplaceMap</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HtmlSanitizationAllowList</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HtmlRemoveList</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StringReplaceMap</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RenderType</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Property Attributes**

Use property attributes to define additional properties for your card template slot. These properties are not associated with a specific attribute type.

Attributes for the Menu type

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>BackgroundColor</td>
<td>Color of the background for the card template element.</td>
<td>Use the <strong>Value</strong> field to define the color. Colors must be in color hex code format, or the name of a color theme. For example Primary or #8a8a8a.</td>
</tr>
<tr>
<td>BorderColor</td>
<td>Color of the border for the card template element.</td>
<td>Use the <strong>Value</strong> field to define the color. Colors must be in color hex code format, or the name of a color theme. For example Primary or #8a8a8a.</td>
</tr>
<tr>
<td>TextColor</td>
<td>Color of the text or the card template element.</td>
<td>Use the <strong>Value</strong> field to define the color. Colors must be in color hex code format, or the name of a color theme. For example Primary or #8a8a8a.</td>
</tr>
</tbody>
</table>

**Mobile Studio**

Use Studio to create and modify mobile components for use in any of the ServiceNow mobile apps.
Use the application explorer to browse mobile components

The Application Explorer appears on the left edge of the screen and displays your mobile components, such as applets, application launchers, and functions. Open your items by clicking them, or select a section, and use the pop-out icon (/button) to display a card view of the components in that section.
Use the card view to filter and search for specific components

The card view displays each of record in the selected section as a card. Each card displays the title of the component, as well as its creator, last update time, and whether the component is active.

Use the Sort by field to sort your components by name or update time. You can also use the search field to filter your results by name.
Create and modify mobile components

Use studio to design layouts for your applets and launchers, set permissions, determine offline-mode availability. See documentation on specific components for instructions on how you can use Studio to make changes and create modify your mobile experience.

Cross scope elements in Studio

Some of the components of mobile apps may use elements from other scopes. You cannot edit these components within Studio, but can view them in a read-only state. These forms display a warning message indicating the current scope as well as the scope of the cross scope element.
Custom fields in Studio

A mobile component may contain custom fields that are not editable within Studio. When you load components containing custom fields, a warning message displays at the top of the form. This warning contains a link you can use to edit the custom fields in ServiceNow’s native UI.

Create a mobile application using Guided Application Creator

Use the Guided Application Creator in Studio to create a base scoped application and mobile app components. These components can be used in any of the ServiceNow mobile apps.

Before you begin
Role required: admin

About this task
Studio is a development environment where your application developers can work on custom applications in one centralized location. Much of the configuration for mobile apps in this section take place in Studio (System Applications > Studio). For more information about Studio, see ServiceNow Studio.
Role required: admin

Procedure
1. Navigate to System Applications > Studio.
2. In Studio, click Create Application or select an existing application from the list.
   The application that you create here is a new scoped application which will contain your app. Scoped applications help restrict data and application files to just this one application. For more information on scoped applications, see Application scope. You can also access the Guided Application Creator outside Studio by navigating to System Applications > My Company Applications, and then click Create new. Studio opens a Guided Application Creator window where you create your application.
3. Optional: If you are launching Guided Application Creator for the first time, click Let’s get started on the welcome screen.
4. Follow the steps on the screen to create a name, description, and logo for your application, and then click Create.
5. Optional: In the Roles field, select the roles to associate with your app.
Users with the selected roles can access your application. If you have selected no roles, users with any role will have access to the application.

6. Optional: Click Create new role to create a new role to associate to the application.

7. Click Continue when you are finished defining roles for your application.

8. Select Mobile as the format for this application, and then click Continue.

9. Select or create tables that you want to use in your mobile app.
   The Guided Application Creator can create list applets for these tables. If you create new applets later, you can will not be limited to the tables you select here.

10. Click Done with Tables when you are finished adding tables to your application.

11. Click the Start button to create applets for your selected tables using the Guided Application Creator.
    If you do not want to create applets at this point, you can click X in the upper right corner of the window to close the Guided Application Creator and return to Studio.

12. Fill in the fields in the Guided Application Creator form as needed:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for the app. The applet launcher created in this process will use this name.</td>
</tr>
<tr>
<td>Description</td>
<td>Description for the app.</td>
</tr>
<tr>
<td>Tables</td>
<td>Tables for which Guided Application Creator will create applets.</td>
</tr>
<tr>
<td>Roles</td>
<td>Roles for your applets. Users with the selected roles can access the created applets.</td>
</tr>
</tbody>
</table>

13. Click Create.
    The Guided Application Creator will create a list and form for each of the selected table, including New, Edit, and Delete functions for each table.

14. Click the Done with apps button, and then the Done button to return to Studio.

Results
You now have a new application that you can access in Studio.

What to do next
Create apps to add functionality to your mobile app using Guided Application Creator, or close Guided Application Creator and create apps using Studio.

Configure a standard data item
Configure a standard data item to query data for your screens that do not require parameter input.

Before you begin
Role required: admin

Procedure
1. Navigate to All > System Mobile > Mobile App Builder.
   The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.
2. Search for the application scope you are working in and then select the name of the application scope. The Mobile App Builder categories home screen displays.

3. Select the Data category, and then select **New**.

4. Select Data item and then select **Continue**.

5. In the Data Items tab, click **Create New**.

6. Complete the following fields as needed.

**Data item fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties section:</strong></td>
<td></td>
</tr>
<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 it easily.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional additional details about the data item.</td>
</tr>
<tr>
<td><strong>Data section:</strong></td>
<td></td>
</tr>
<tr>
<td>Table</td>
<td>The table you want the data item to pull information from.</td>
</tr>
</tbody>
</table>

**Note:** Custom tables are not available by default. You can change this behavior by modifying the `subscription.custom_table.enforce_entitlement` system property. For details on making this change, see Allow or restrict access to custom tables in mobile data items.

| Group by                | Groups query results based on the selected field from the menu. |
| Condition type          | Determines what type of condition your data item uses. Select from: |
|                        | **Declarative** |
|                        | Use a declarative condition to create conditions for the data item using the condition builder. |
|                        | **Scripted** |
|                        | Use a script to determine the conditions of your data item. A text window to enter a script appears below this field when you select this option. |
|                        | **Append Encoded Query** |
|                        | Use this option only when creating data items for your... |
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chart screens. For details on that data item type, see Configure an encoded query data item for chart screens.</td>
<td></td>
</tr>
</tbody>
</table>
| Condition           | Set of conditions for the data item to conform to. You can create conditions using the condition builder.  
  Note: This field is only available when you select Declarative in the Condition type field. |
| Sort by             | Lets you adds fields to the form that allow you to configure how to sort the list. In the condition builder, select the field you want the list to be sorted by. For example, select Caller. Then select ascending or descending to determine the order of the list.  
  Note: This field is only available when you select Declarative in the Condition type field. |

7. **Optional:** If you have selected **Scripted** in the **Condition type** field, you must create a script in the **Query Condition Script** field.

(Optional) Your scripted condition must return a query string, which the instance uses to filter the data item. Use the input variable to access information from the current record.

```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);
```

This example uses the input variable and filters the data item for records matching the current records company and location. It then appends the text \(^EQ^ORDERBYDESCsys_updated_on\) to the query, which sorts the data item records by the **Updated on** field.

8. **Select** **Save**.

**What to do next**
Associate a data item with a screen. For more information, see **Mobile screens**.

**Related information**
- Configure a parametrized data item
- Configure a group by data item
Empty state display

Use an empty state to indicate to your users that the displayed page does not contain data. You can add an image, text, and buttons to direct users to perform an action, view a particular screen, or review specific information.

In some situations, such as the first time a user accesses a page or if no relevant items are listed on a page, data might not be displayed. You can configure a default empty state to be displayed in those situations. You can customize additional empty states for particular lists and screens. If an empty state is not defined for specific lists and screens, the default empty state is used.

Empty state display hierarchy

The default empty state is triggered for all screens and segments without an associated empty state. An empty state defined for a specific screen takes priority over the default empty state. An empty state defined for embedded lists takes priority over an empty state defined for a screen.

**Note:** There is a separate default empty state for search results. You can also define an empty state if no search results are listed, after a navigation tab is selected.

<table>
<thead>
<tr>
<th>Default empty state</th>
<th>Configured empty state</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Default empty state" /></td>
<td><img src="image2.png" alt="Configured empty state" /></td>
</tr>
</tbody>
</table>

© 2022 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.
**Empty state structure**

You can define up to one image element, two text elements, and three button elements within a single empty state display. The elements are displayed in the vertical alignment in the order image, text, and then button. The following graphic shows the element order with its corresponding element name. Unused elements are not displayed in the empty state.

**Empty state with predefined system names used in configuration**

Configure an empty state

Configure an empty state to inform users that a page currently does not contain data. Add an image, text, and buttons to customize the empty state and improve the user experience.

**Before you begin**

Whether creating a default or specific empty state, the following items might be required if you plan to display an image or button:

- An image from the Attachments [sys_attachment] table.
- Defined button actions from the Function [sys_sg_button] table. For more information, see Configure a smart button.

Role required: admin

**About this task**

This task explains how to configure a default empty state used in all screens. For information about how to customize empty states for specific screens, see the following topics:

- Configure an empty state for a list screen.
- Configure an empty state for an embedded list in a form screen.
- Configure an empty state for search results.
Procedure

1. Create a view configuration ("view config"), which is the receptacle where you define empty state elements and their attributes.

   a. In the web-based UI, enter `sys_sg_view_config.list` in the filter navigator.

   b. Select **New** to create a view config.

   c. On the form, fill in the fields.

       **View Config form**

       | Field           | Value                                                                 |
       |-----------------|-----------------------------------------------------------------------|
       | Name            | Title for the view config.                                            |
       | Application     | Scoped application associated with this record. This field defaults to |
                     | the current application.                                             |
       | Description     | Optional description of the view configuration.                       |
       | Active          | Option for enabling the display of the view config setup within the   |
                     | empty state. For this configuration, select this option.             |
       | Table           | No table is necessary for this configuration.                         |
       | View template   | The value should be Default Empty State Template.                      |

   d. Right-click in the header and select **Save**.

2. Select an element to add to the view config.

   a. On the View Config Elements form, select **New**.

   b. On the form, fill in the fields.

       **View Config Elements form**

       | Field       | Value                                                                 |
       |-------------|-----------------------------------------------------------------------|
       | Name        | A title for the view config element.                                   |
       | View config | This field is automatically set with the name of the new view config. |
       | Application | Scoped application associated with this record. This field defaults to |
                     | the current application.                                             |

   c. In the **View template slot** field, select the reference lookup icon (🔍) and select a template slot to add to the view config element.
<table>
<thead>
<tr>
<th>Element name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>es-image-slot-1</td>
<td>The image element of the empty state. You can use only one image.</td>
</tr>
<tr>
<td>es-text-slot-1</td>
<td>The header element of the message, which is displayed in bold.</td>
</tr>
<tr>
<td>es-text-slot-2</td>
<td>The body element of the message which is displayed in a non-bold font.</td>
</tr>
<tr>
<td>es-button-slot-1</td>
<td>The top button element of the empty state.</td>
</tr>
<tr>
<td>es-button-slot-2</td>
<td>The middle button element of the empty state.</td>
</tr>
<tr>
<td>es-button-slot-3</td>
<td>The bottom button element of the empty state.</td>
</tr>
</tbody>
</table>

d. Right-click in the header and select Save.

3. Configure attributes for either the image, text, or button element you added in the view config.

<table>
<thead>
<tr>
<th>Element</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>b. Set the following values:</td>
</tr>
<tr>
<td></td>
<td>• Name: Use the reference lookup icon and select ImageURL.</td>
</tr>
<tr>
<td></td>
<td>• Description: Optional description of the element.</td>
</tr>
<tr>
<td></td>
<td>• Value: Paste the sys_id of the image from the Attachments [sys_attachment] table.</td>
</tr>
<tr>
<td></td>
<td>✪ Note: To locate the sys_id of an element, right-click on the element name and select Copy sys_id.</td>
</tr>
<tr>
<td></td>
<td>c. Select Submit.</td>
</tr>
<tr>
<td>Text</td>
<td>a. On the View Config Element Attributes form, select New.</td>
</tr>
<tr>
<td></td>
<td>b. Set the following values:</td>
</tr>
<tr>
<td>Element</td>
<td>Action</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
</tr>
<tr>
<td></td>
<td>• <strong>Name</strong>: Use the reference lookup icon and select <code>TextValue</code>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Description</strong>: Optional description of the element.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Translatable value</strong>: Enter text for the button. This text is translated if a translation of the UI takes place.</td>
</tr>
<tr>
<td></td>
<td>c. Select <strong>Submit</strong>.</td>
</tr>
</tbody>
</table>

**Note**: Button elements require the configuration of both the `TextValue` and `Button` attributes.

a. On the View Config Element Attributes form, select **New**.
b. Set the following values:
   • **Name**: Use the reference lookup icon and select `TextValue`.
   • **Description**: Optional description of the element.
   • **Translatable value**: Enter text for the button. This text is translated if a translation of the UI takes place.
c. Select **Submit**.
d. On the View Config Element Attributes form, select **New**.
e. Set the following values:
   • **Name**: Use the reference lookup icon and select `Button`.
   • **Description**: Optional description of the element.
   • **Value**: Paste the `sys_id` of the image from the Function `[sys_sg_button]` table.

**Note**: To locate the `sys_id` of an element, right-click on the element name and select **Copy sys_id**.
f. Select **Submit**.

Button elements can also include the following optional attributes: `BackgroundColor`, `BorderColor`, and `TextColor`. You can set colors other than the default settings for a button by providing a theme color name (for example, primary) or an RGB color value (for example, #8a8a8a).
4. **Optional:** Add another view config element and then add additional attributes, repeat Step 2 and Step 3.

5. Attach the view config to the empty state.

   a. In the web-based UI, enter `sys_sg_empty_state.list` in the filter navigator to open a list of empty states.

   b. Select **New** to create an empty state.

   c. On the form, fill in the fields.

   **Empty states form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Title for your empty state record. Either your default empty state or a screen-specific empty state.</td>
</tr>
<tr>
<td>Application</td>
<td>Scoped application associated with this record. This field defaults to the current application.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the empty state.</td>
</tr>
<tr>
<td>View Config</td>
<td>The view config record that you created.</td>
</tr>
<tr>
<td>Active</td>
<td>Option for enabling the display of the empty state.</td>
</tr>
</tbody>
</table>

   d. Select **Submit**.

**Results**

An empty state displays in any screen that does not contain data.

**Navigation bar**

Users can access launcher screens, screens, settings, and notifications with the navigation bar in the mobile app.

**Note:** Beginning with the Rome release, some terminology has changed for certain ServiceNow® mobile products. This topic content references Mobile Studio, which retains the terminology used in previous releases. For more information, see [ServiceNow mobile terminology changes](#).

The navigation bar consists of these components:

- Navigation bar
- Navigation bar tabs
  - Screen tab
    - Launcher tab
  - Saved tab
The navigation bar appears at the bottom of each mobile app. You can create navigation bar tabs in the navigation bar. Users can access launcher screens and regular screens with the navigation bar.

Note: The navigation bar in each mobile app is pre-configured with **Notifications** and **Settings** navigation bar tabs.

When you add more than five tabs to the navigation bar, a **More (⋯)** tab appears. Tapping the **More** tab opens a list view showing more tabs.
Screen tabs and launcher screen tabs

Use a screen tab to enable a single screen. Single screens can be a map, calendar, or list of contacts.

For details on creating screen navigation tabs, see Configure a screen tab.
Configure the navigation bar

Configure the navigation bar that appears at the bottom of the mobile app. Your users use the navigation bar to quickly navigate to screens, screen launchers, settings, and notifications.

**Before you begin**
Role required: admin

**About this task**

*Note:* Beginning with the Rome release, some terminology has changed for certain ServiceNow® mobile products. This topic content references Mobile Studio, which retains the terminology used in previous releases. For more information, see ServiceNow mobile terminology changes.

**Procedure**

1. Navigate to All > System Mobile > Mobile App Builder.
   - The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.

2. Search for the application scope you are working in and then select the name of the application scope.
   - The Mobile App Builder categories home screen displays.

3. Select the **Navigation bar** record in the left menu and modify the fields as needed:
Navigation form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the app that uses this navigation bar. For example, Mobile Agent is the navigation bar name for the Mobile Agent app.</td>
</tr>
<tr>
<td>Color</td>
<td>The hexadecimal value for the color of the navigation bar. If this field is left blank, the default colors of the mobile theme are used.</td>
</tr>
<tr>
<td>Quick Actions Menu Color</td>
<td>The hexadecimal value for the color of the quick actions icon that appears on the screen. If this field is left blank, the default colors of the mobile theme are used.</td>
</tr>
</tbody>
</table>

What to do next
After configuring the navigation bar, you can add specific tabs to it:

1. On the same navigation bar record, you might need to select **Edit in original scope** to switch to the original scope.
2. In the appropriate scope, select **Navigation tabs**, and then select **Choose**.
3. Select the desired navigation bar tab record or add, and then select **Apply**.
   The new navigation bar tab record is added.
4. Select **Order** to adjust the order the navigation bar tab should appear within the navigation bar.

   **Note:** The navigation bar is ordered from least to greatest or from left to right. This means that any tab having the value of zero always appears on the left side of the navigation bar.

To add a new tab to your navigation bar, see **Configure a launcher screen tab** or **Configure a screen tab**.

Configure a launcher screen tab
Configure a tab on your navigation bar to direct users to a launcher screen.

**Before you begin**
Role required: admin

**About this task**

**Note:** Beginning with the Rome release, some terminology has changed for certain ServiceNow® mobile products. This topic content references Mobile Studio, which retains the terminology used in previous releases. For more information, see **ServiceNow mobile terminology changes**

**Procedure**

1. Navigate to **All > System Mobile > Mobile App Builder**.
   The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.
2. Search for the application scope you are working in and then select the name of the application scope. The Mobile App Builder categories home screen displays.

3. Select All mobile records in the left menu.

4. From the Record type list, select Launcher tab [sys_sg_applet_launcher_tab], and then select New. The new launcher tab screen appears where you can create a tab.

5. On the form, fill in the fields as needed.

Launcher tab form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Name that displays for your navigation bar tab.</td>
</tr>
<tr>
<td>Active</td>
<td>Toggle that sets whether the navigation bar tab is active. Inactive navigation bar tabs aren't visible.</td>
</tr>
<tr>
<td>Launcher</td>
<td>Launcher screen that appears when a user taps the tab. This field is only available for launcher screen tab records. For details on creating a launcher screen, see Launcher screens.</td>
</tr>
<tr>
<td>Icon</td>
<td>Icon that displays for your navigation tab.</td>
</tr>
<tr>
<td>Role access</td>
<td>Roles that can access this launcher tab.</td>
</tr>
</tbody>
</table>

6. Select Save.

What to do next

After creating a launcher screen tab, add the new tab to a navigation bar. For more information, see Configure the navigation bar.

Note: Changes to navigation bars are not visible to end users until they refresh their metadata. To refresh metadata, refresh any screen launcher in the app.

Configure a screen tab

Configure a tab on your navigation bar to direct your users to a commonly used screen, such as a map, calendar, or list.

Before you begin

Role required: admin

About this task

Note: Beginning with the Rome release, some terminology has changed for certain ServiceNow® mobile products. This topic content references Mobile Studio, which retains the terminology used in previous releases. For more information, see ServiceNow mobile terminology changes.
Procedure

1. Navigate to All > System Mobile > Mobile App Builder.
   The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.

2. Search for the application scope you are working in and then select the name of the application scope.
   The Mobile App Builder categories home screen displays.

3. Select All mobile records in the left menu.

4. From the Record type list, select Screen tab, and then select New.

5. On the form, fill in the fields.

   **Screen tab form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Name that displays for your screen tab.</td>
</tr>
<tr>
<td>Active</td>
<td>Toggle that sets whether the screen tab is active. Inactive screen tabs are invisible.</td>
</tr>
<tr>
<td>Screen</td>
<td>Screen that appears when a user taps on the tab. For details about creating a screen, see Create a screen.</td>
</tr>
<tr>
<td>Icon</td>
<td>Icon that displays for your screen tab.</td>
</tr>
<tr>
<td>Role access</td>
<td>Roles that are able to access the screen tab.</td>
</tr>
</tbody>
</table>

6. Select Save.

**What to do next**

After creating a launcher screen tab, add the new tab to a navigation bar. For details about configuring a navigation bar, see Configure the navigation bar.

**Note:** Changes to navigation bars are not visible to end-users until the metadata is refreshed. To refresh metadata, refresh any screen launcher in the app.

**Launcher screens**

Launcher screens serve as landing pages or home pages. Using a launcher screen, you can access screens in various formats, as well as search, do quick actions, and find user information.

**Note:** Beginning with the Rome release, some terminology has changed for certain ServiceNow® mobile products. This topic content references Mobile Studio, which retains the terminology used in previous releases. For more information, see ServiceNow mobile terminology changes.

Launcher screens serve as landing pages. A launcher screen serves as a container to give your users access to functions and information.
When creating a new launcher screen, you configure the following few components within the launcher screen itself:

**Headers**

The header of the launcher screen defines how the title of the screen appears and what information is shown in the header.

**Global search**

Global search gives your users the ability to search for information within the defined search sources from a mobile app.

**UI sections**

UI sections contain the screen, charts, and record information on your launcher screen pages.

**Quick actions**

Quick actions give users the ability to access commonly used functions from a launcher screen.

---

### Configure a launcher screen header

Create a launcher screen header to define how the title of the screen appears.

**Before you begin**

Role required: admin

**About this task**

⚠️ **Note:** Beginning with the Rome release, some terminology has changed for certain ServiceNow® mobile products. This topic content references Mobile Studio, which retains the terminology used in previous releases. For more information, see ServiceNow mobile terminology changes

**Procedure**

1. Navigate to **All > System Mobile > Mobile App Builder**.
   The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.

2. Search for the application scope you are working in and then select the name of the application scope.
   The Mobile App Builder categories home screen displays.

3. Select **All mobile records** in the menu.

4. From the Record type list, select **Launcher screen [sys_sg_applet_launcher]**, and then select one of the following options:
• **New**
  • An existing launcher screen record
  Configuring the header title is required. You can also configure the header function instance, which is optional.

5. To configure the header title, locate the Settings pane, and then select **New**. The launcher header screen appears where you can create a header title.

6. Enter a new header title in the **Properties** field and select **Save**. To use a variable in the header title such as combining Hello with the variable (username), use an existing header title record:
   a. Instead of selecting **New** in the Settings pane, select **Choose**.
   b. Select one of the base system header titles.

7. **Optional:** To configure the header function instance, select **Launcher screen** in the menu on the left. In the Launcher screen page, locate the Header function instance section. When you configure a header function instance, an icon appears in the header:

   ![Header function instance icon](image)
Tapping the icon takes a user to the destination screen that is configured with the header function.

8. In the Header function instance section of the page, select **New**.


### Function instance form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the header function instance. This name is internal and isn’t visible to end-users.</td>
</tr>
<tr>
<td>Display label</td>
<td>Label that is visible to end-users.</td>
</tr>
<tr>
<td>Order</td>
<td>Optional field that you can use to set the order that the UI functions appear. If you have multiple functions, set <strong>Order</strong>.</td>
</tr>
<tr>
<td>Active</td>
<td>Toggle that turns the header function on and off. If the toggle is enabled, the header function is enabled.</td>
</tr>
<tr>
<td>Icon</td>
<td>Icon that appears in the header of the launcher screen.</td>
</tr>
<tr>
<td>Function</td>
<td>Function that runs when a user taps on the icon in the header of the launcher screen.</td>
</tr>
</tbody>
</table>

10. Select **Save**.

### Global search for mobile

Give your users the ability to search for information within the defined search sources from a mobile app.
Global search bar

You can include a global search bar on your applet launcher. Use search to give your users the ability to quickly find information within the defined search sources. Examples include people, catalog items, and knowledge base articles in your instance.

There are two search engines available to activate within applet launchers.

- **Zing** - The default search engine used to search Now Platform® record data. For more information, see Zing text indexing and search engine.

- **AI Search** - AI search engine available for Now Mobile, which offers visually intelligent query features to help users quickly find the answers they need. For more information see, AI Search.

Search criteria

Create search criteria for your applet launcher to help your users find related information quickly. You can configure which sources the search uses, as well as options like search result limits. For details on configuring custom search criteria:

- For Zing search engine, see Configure search criteria.
- For AI Search search engine, see the Now Mobile documentation AI Search in mobile.

**Note:** AI Search utilizes the data sources of catalog items, KB articles and users. Use AI Search, wherever possible, when using these data sources. Use Zing in the applet launcher if you utilize other data sources not associated with AI Search, for example user tasks or requests.

Voice search

Voice search enables your users to use global search using the speech recognition feature of their mobile device. Users access voice search by tapping the microphone icon ( 🔊) that
appears on the right side of a search bar. For more information on configuring voice search, see Enable voice search.

Photo search

Photo search enables users to use global search by taking a picture. The Google Vision API identifies the picture and uses that identification as search query. Users access photo search by tapping photo icon ( ) that appears in your applet launcher search bar. For more information on photo search, see Mobile photo search.

Enable global search in your applet launcher

Enable global search using the Zing search engine on your applet launcher, to give your users the ability to quickly find information within the defined search sources.

Before you begin
Role required: admin

Procedure

1. Navigate to All > System Mobile > Applet Launcher
2. Open the record for the applet launcher where you want to add search functionality.
3. In the Header tab, select Homepage Search - Catalog, Knowledge, People in the Search Configuration field.
4. Click Save.

Configure global search options for your applet launcher

Configure global search options, using the Zing search engine, to control how your app presents search results to your users.

Before you begin
Role required: admin

Procedure

1. Navigate to All > System Mobile > Applet Launchers.
2. Open the record for the applet launcher where you want to add search functionality.
3. In the **Search Configuration** field, click the reference icon (①) to open the record preview, and then click **Open Record**.

4. In the **Placeholder** field, enter the text that you want to appear in the search bar before a user enters a value into the field. The placeholder text appears in the search bar before your users enter a search query.

5. Select **Interleave Result** to display interleaved search results. If the option is not selected, search results are separated by the search source.

6. Click **Update**.

7. In the **Search Context Configuration** field, click the reference icon (①) to open the record preview, and then click **Open Record**.

8. In the **Search Results Limit** field, enter the maximum allowed search results.

9. In the **Suggestions Limit** field, enter the maximum allowed search suggestions.

10. Click **Save**.

**What to do next**
Enable voice search so that your users can search using native speech recognition. For more details, see **Turn on voice search**

**Populate search suggestions in the mobile apps**
If you are upgrading from a previous release, run a script to populate search suggestions with data from a platform search table to provide search suggestions to your users. Alternatively, you can wait until users search for keywords instead of running this script.

**Before you begin**
Role required: admin
In new instances, the search suggestions are enabled by default. In upgraded instances, you must enable the search suggestions. For more information, see **Enable search suggestions**.

**About this task**
The Now Platform collects search data and analytics that generate search suggestions. If you are upgrading from a previous release, the search analytics do not contain any data yet. To immediately provide suggestions to your users, you can populate the search suggestions using knowledge, catalog, and user search records from the Text Searches [text_search] table.

Search suggestions improve over time as more people use the app. Search Suggestions is a Now Platform feature. For more information, see **Search Suggestions**.

⚠️ **CAUTION:** Populating search suggestions can be a resource-intensive task that may take a while to complete. Do not run this script during peak hours. Populating search suggestions is not supported on domain-separated instances.
Procedure

1. Navigate to All > System Definition > Scheduled Jobs.

2. Open the Populate Suggestions to avoid Cold Start - NowMobile App scheduled job.

3. Activate the record and select Execute Now.
   Running this scheduled job populates the Search Events [sys_search_event], Search Source Events [sys_search_source_event], and Search Suggestions [sys_search_suggestion] tables with records from the Text Searches [text_search] table.

Results

Users see suggestions when they start typing in a search field in the Now Mobile app.

Configure search criteria

Create a custom search for your applet launcher to help your users find related information quickly. Administrators can configure which search sources the search uses, as well as options like search result limits.

Configuration for search consist of linking the following records:

   Applet launcher
   The applet launcher is where the search bar appears within your mobile application.

   Global search configuration
   The global search configuration [sys_sg_global_search] record is what associates your applet launcher with a search context configuration and an item configuration. This record is also where you can configure placeholder text, and enable photo search.

   Item configuration
   Item configurations provide a pattern for data in your applet, and control how your data appears within a segment. In this case, the appearance of your search results.

   Search context configuration
   The search context configuration [sys_search_context_config] record connects your search source with your search configuration. This record is also where you can set limits on search results and search suggestions.

   Search source
   Search source [sys_search_source] records determine what tables on your instance are included in your searches.
Create a search source

Create a search source to define what records can be found in your search bar.

Before you begin
Role required: admin

About this task
In these steps, you create the search source to define what search results your users can see. Each search source consists of a table, and optional conditions. User will see results from the selected table, limited by the conditions you define in the search source. You could, for example, select the Incident [incident] table, and then create a condition so that only incidents in the open state appear in search results. You can define more than one search source for a search, so your users can find records from multiple tables that match their search input.

Procedure
1. In the filter navigator, enter sys_search_source.list.
2. Click New.
3. On the Search Source field, fill in the fields.

Search Source form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of your search source. Enter a name that includes the name of the table you intend to use to make the search source easy to identify.</td>
</tr>
<tr>
<td>Application</td>
<td>The scoped application for the search source. This field is ready only and automatically contains the name of the current application.</td>
</tr>
<tr>
<td>Table</td>
<td>The table for this search source.</td>
</tr>
<tr>
<td>Conditions</td>
<td>Conditions under which a record from the selected table appears in search results. For example, if you select &lt;Active&gt;&lt;is&gt;&lt;true&gt;, only active records appear in your search results.</td>
</tr>
</tbody>
</table>
**Example**
A search source configured for the User [sys_user]

![Search Source](image)

4. Click **Save**.

5. **Optional**: Repeat steps 1 through 4 to define additional sources for your search. Note that additional search sources can impact the performance of your searches.

Create a global search configuration

Create a global search configuration. This record connects your applet launcher with the other records used to configure search.

**Before you begin**
Role required: admin

**Procedure**
1. In the filter navigator, enter `sys_sg_global_search.list`.
2. Click **New**.
3. In the global search configuration form, fill in the fields:

   **Global search configuration form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for your search configuration.</td>
</tr>
<tr>
<td>Application</td>
<td>The scoped application for the search configuration. This field is ready only and automatically contains the name of the current application.</td>
</tr>
<tr>
<td>Placeholder</td>
<td>Text that appears in the search bar before a user enters any text.</td>
</tr>
<tr>
<td>Search Context Configuration</td>
<td>Search context configuration record used for this search configuration.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Leave this field empty. You create a search context configuration record in the following steps.</td>
</tr>
<tr>
<td>Interleave Result</td>
<td>Whether search results are interleaved. Deselect this option to display search results separated by search source.</td>
</tr>
<tr>
<td>Enable Photo Search</td>
<td>Whether photo search is available for your search bar. If the glide.sg.image_recognition.search.enable system property is false, you cannot select this option. For details on this property see Create the image search system property.</td>
</tr>
</tbody>
</table>

4. In the **Search Context Configuration** field, click the reference icon (🔍).

5. In the search context configuration list, click **New**.

6. In the Search Context Configuration form, fill in the fields:

   **Search context configuration form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for your search context configuration.</td>
</tr>
<tr>
<td>Search Results Limit</td>
<td>Limit for the initial results returned in the search. Scrolling down with load additional results in increments defined by this value. This field has a default value of 10.</td>
</tr>
<tr>
<td>Application ID</td>
<td>The scoped application for the record. Select the same scoped application as the applet launcher where you want to use with this search configuration.</td>
</tr>
<tr>
<td>Suggestions limit</td>
<td>Limit for search suggestions listed for the search. This field has a default value of 10.</td>
</tr>
</tbody>
</table>

7. Right-click the header of the Search Context Configuration form, and click **Save**. The Search Context Configuration record saves. After the save, the Application Search Sources related list appears on the form.

8. In the Application Search Sources related list, click **New**.

9. In the **Source** field of the Application Search Sources form, select a search source you created in the previous steps.

10. In the **Order** field, add a value to determine the order in which the search source facets appear in the search results.

   **Note:** Search sources display from the lowest order to the highest.
Example
A search context configuration with an associated search source.

<table>
<thead>
<tr>
<th>Name</th>
<th>Mobile Employee Search Config for Services &amp; Items</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application ID</th>
<th>ServiceNow Mobile - Mobile Employee Search Config for Services &amp; Items</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

11. **Optional:** Repeat steps 8 through 10 to add additional search sources if you created them in previous steps.

12. Click **Submit**.

13. Right-click the header of the global search configuration form, and click **Save**.

14. In the **Global Search M2M Item Configuration** list, click **New**.

15. In the Global Search M2M Item Configuration form, click the reference icon (🔍) next to the **Item Configuration** field.

16. Select the item configuration you want to use for a table that contains your desired search results. For example, if you want your search to return problem records, select the item configuration that has **problem** in its **Table** field.

17. On the Global Search M2M Item Configuration form, click **Submit**.

18. **Optional:** Repeat steps 9–12 to add item configurations for additional record types if you plan to include multiple tables in your search.

19. Click **Save** when you have finished adding item configurations.

**Add your search configuration to an applet launcher**
Update your applet launcher to use your custom search configuration.
**Before you begin**  
Role required: admin

**Procedure**

1. Navigate to All > System Applications > Studio.
2. Select your application.
3. In Application Explorer, navigate to Mobile Studio > Applet Launchers.
4. Open the applet launcher where you want to add search.
5. Click Include Search option to enable it.
6. In the Search Criteria field, select the global search configuration record to associate with your search option.
7. Click Save.

**Enable voice search**

Enable your users to search for items, articles, and people using native speech recognition from an app on their mobile device.

**Before you begin**  
Role required: admin

**About this task**

⚠️ **CAUTION:** Voice search uses native speech recognition and relies on your operating system's cloud server to transcribe voice into text search. If you have data privacy concerns about search queries moving to the operating system cloud server, do not turn on voice search.

**Procedure**

1. In the application navigator, enter sys_properties.list.
2. Click New to add a new system property.
3. On the form, fill in the fields.

**System property form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter glide.sg.voice_search.enabled.</td>
</tr>
<tr>
<td>Type</td>
<td>Select true</td>
</tr>
<tr>
<td>Value</td>
<td>Enter True.</td>
</tr>
</tbody>
</table>

4. Click Submit.

**Results**

The global search bar in the Now Mobile app includes a microphone icon. Users must select this icon and allow the app to access speech recognition on their mobile device to use voice search.

**Mobile photo search**

Configure photo search to give your users the ability to perform image-based searches using the objects around them.
Photo searches in your mobile apps

When photo search is configured on your instance, a photo icon (📷) appears in your applet launcher search bar. Your users can tap this icon to use the camera on their mobile device to take a picture. The picture is identified using the Google Vision API, which returns one or more results. Users can select a result, which is used as their search query.

Google Firebase account

To take advantage of image recognition in your mobile applications, you need a Firebase account with the Google Vision API enabled. Google Firebase and the Google Vision API are third-party services that must be subscribed to separately. Once you have a Firebase account, you can connect that account to your ServiceNow instance by entering your Firebase App ID and Web API key. For details on this process see Configure photo search.

Third-party data usage

Images you take using the photo search feature are sent to Google for identification using Google Vision API. ServiceNow does not have control of the image once it has been sent. For details on how Google Vision API handles your image, see https://cloud.google.com/vision/docs/data-usage.

Configure photo search

Configure photo search for your mobile applications by enabling photo search in system properties, creating a Firebase account, and connecting your account to your ServiceNow instance.

Create a Firebase account

Create a Google Firebase project and enable the Google Vision API.

Before you begin

Role required: admin
Photo search requires a Google Cloud / Firebase account to analyze images and return search results. If you already have a Google account, you can use your existing account. If not you can create an account at https://firebase.google.com/.

**Note:** Configuration for your Firebase account includes upgrading your account to the Blaze plan. This plan is a billable service that you must configure with Google.

**Procedure**
1. Navigate to Firebase website, and sign in with a new or existing Google account.
2. Click **Create Project**.
3. Give your project a name and continue through the guided steps. The Firebase project page appears once you have completed the project setup.
4. From the project setup page, add an iOS or Android app to your project by selecting the **iOS** or **Android** button.

![Firebase Setup](example-project-screenshot)

5. In the **App setup** page, fill in the form to add an app to your Firebase project.
   a. Enter a **bundle ID** and **App nickname** for your app. There are no specific requirements for the bundle ID and app nickname fields.
   b. You do not need to download the configuration file, add an Firebase SDK, add an initialization code, or run your app to verify installation. You can click **Next** or **Skip this step** to bypass these steps.
      After completing setup you can see the configuration page for your project, with your new application listed.
6. Click the **Spark Plan** button.
7. Select the Blaze plan.

   **Note:** The Blaze plan is a billable plan. For details on pricing see Firebase pricing.
8. When prompted, create a billing account and enter in your payment details.
10. Log in to the website using the same Google account used in the previous steps.
11. In the header, select the project you created in previous steps.
12. Use the **APIs & Services** menu option to locate and enable the Google Vision API.
13. Return to the Firebase website, and open your project settings page.
14. In another browser tab or window, log in to your ServiceNow instance.

15. Navigate to **System Mobile > Mobile Publishing > Native Clients**, and select the record for the mobile app where you want to configure photo search.

16. In the **Vision iOS App ID** field, enter the App ID for your iOS app listed on your Firebase project settings page.

17. In the **Vision Android App ID** field, enter the App ID for your Android app listed on your Firebase project settings page.

18. In the **Vision API Key** field, enter the Web API Key listed on your Firebase project settings page.

19. Click **Update**.

**Connect your instance to your Firebase project**

Enter your project ID and Google Vision API key into your instance to begin using photo search.

**Before you begin**

Role required: admin

**Procedure**

1. Navigate to **All > System Mobile > Mobile Publishing > Native Clients**.

2. Select the record for the app for which you want to enable photo search.

3. In the **Vision iOS App ID** field, enter the iOS App ID from your Firebase project.

4. In the **Vision Android App ID** field, enter the Android App ID from your Firebase project.

5. In the **Vision API Key** field, enter the Google Vision API key from your Firebase project.

6. Click **Update**.

Your app is configured to use the Google Vision API for photo search. Repeat the steps if for your other mobile apps if you want to include the photo search option in those apps. Before using the photo search, you must still enable the system property as detailed in the next steps.

**Create the image search system property**

Create and enable a system property to control access to image search on your instance.

**Before you begin**

Role required: admin

**Procedure**

1. In the filter navigator, type **sys_properties.list** to display the list of system properties for your instance.

2. Click **New**.

3. Fill in the system property form as needed.

**System property form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>glide.sg.image_recognition.search.enable</td>
</tr>
<tr>
<td>Type</td>
<td>true</td>
</tr>
</tbody>
</table>
4. Click Submit.
Your instance is configured to use photo search. You can disable photo search on your instance at any time by deleting the property record or by setting the Value field to false.

Note: Enabling this system property grants access to image search for all users.

What to do next
Enable global search in your applet launcher to begin using photo search on your mobile applications. For details on enabling search on your applet launchers see Enable global search in your applet launcher.

Enable photo search on your applet launcher
Enable photo search on your applet launcher record in Studio to present the photo search option for your users.

Before you begin
Role required: admin

Procedure
1. In the application explorer in Studio, navigate to Mobile Studio > Applet Launchers.
2. Open the applet launcher where you want to enable photo search.
3. Enable the Include Search option if it is not already enabled.
4. Enable the Photo Search option.
5. Click Save.

Configure an empty state for search results
Configure a customized empty state for search results to provide users with information and actions to help improve their search criteria. This information will replace the default empty state that is provided for search results.

Before you begin
You should have an empty state for the following situations:

• When a search for a term does not produce any results.
• When a search using a navigation tab does not produce any results.

For more information, see Configure an empty state.
Role required: admin

Procedure
1. In the web-based UI, enter sys_sg_global_search.list in the filter navigator.
2. Select the global search configuration type, either AI Search or Zing, for which you want to display the empty state search results.
3. In the Default empty State field, select the empty state to be displayed when a searched term does not produce any results.
4. In the **Filtered Empty State** field, select the empty state to be displayed when a selected navigation tab does not produce any results.

5. Right-click in the header and select **Save**.

**Launcher screen UI sections**

UI sections contain the applets, charts, and record information on your launcher screens.

Note: Beginning with the Rome release, some terminology has changed for certain ServiceNow® mobile products. This topic content references Mobile Studio, which retains the terminology used in previous releases. For more information, see [ServiceNow mobile terminology changes](#).

**UI sections**

Use UI sections to display screens and record information on your launcher screens. UI sections serve as containers for different types of data displayed within your launcher screens.

Types of UI sections include:

---

**Defects**

A Defects UI section helps you to visualize your data. Use the two sub-types, **Chart** and **Single Score** UI sections to display a preview of a single score report, time series report, or **Performance Analytics** widget.

**Chart** and **Single Score** previews are displayed as cards. When a user taps a chart, they open the analytics screen which displays more information on the report or widget.

For information on configuring analytics sections see [Configure chart UI sections](#).

---

**Stories in Current Sprint**

7
Pending Completion

14
Completed

---
Use a **Content** section to display a combination of video, image, and text cards in a carousel format, at the top of your screen. Users can interact with the promotional displays by either viewing videos, being redirected to web pages, or navigating to defined areas on their mobile device.

For more information on mobile campaign configuration, see **Configure content UI sections**.

---

**Welcome System Administrator**

**Horizontal Icons**

- ![Map Test](#)
- ![Employee Directory Test](#)
- ![URL Test](#)
- ![List Test](#)

**Vertical Icons**

| ![Map Test](#) | 554 |
| ![Employee Directory Test](#) | 554 |
| ![URL Test](#) | |
| ![List Test](#) | 37 |

**Note:** Legacy Icon sections are not as flexible as Icon sections.

Use **Legacy Icon** sections to display icons that navigate to other screens but not to launcher screens or functions. Users can tap on an icon to navigate to a different list, map, calendar, or any other sys_sg screen type. You can select a **Horizontal** or **Vertical** layout for your icons. The guidance for Legacy Icon layout is the same as the layout used for an Icon section.

For details on creating Legacy Icon sections, see **Configure legacy icon sections**.
Use a **Record** section to display record information on a card. Records from the selected list screen are displayed as cards. The fields on the cards can be configured. Users tap on an item to see additional record information in a separate record screen. Users can also tap an item to take actions.

Use the record section for records which require the user’s attention, such as high priority or overdue items, and unassigned tasks. Record sections can also be used to display employee records. Choose from a **Horizontal** or **Vertical** layout when creating your record section.

**Horizontal layout**

Use horizontal record sections to display a set number of items you want the user to see. Users can scroll through cards horizontally, and can select **See All** to view the complete list in another screen.

**Vertical layout**

Use vertical sections when you have more room and want to display a set number of records within the screen. Vertical record sections are good for records like critical tasks where you want the user to see details as soon as they land on the page.

For details on creating record sections, see **Configure record sections**.

---

Use **Media** sections to display a single image or video on your launcher screen. Users can tap a media section to navigate to another screen or to navigate to a static URL.

Media sections can display a single line header, and up to two lines of additional text. You can define a navigation function and the text to appear on the navigation button.

Media sections are also a way to add a static company logo to the top of your launcher screen.

For information on configuring media sections see **Configure media UI sections**.
Use **Icon** sections to display screens, launcher screens, and functions using the icon defined for each. Users can tap on an icon to list, maps, reports, calendars, or any other screen or launcher screen type. You can select a **Horizontal** or **Vertical** layout for your icons.

**Horizontal layout**

Use this section type to display many screens or launcher screens in one horizontal view. Limit the name character count to 20 or fewer characters so it does not get cut off. Use this section type to group related screen or launcher screens that do not require counts.

**Vertical layout**

When displayed vertically, you can use the **Display feed count** option to display a count of records within each screen. Use this type for actionable items such as approvals or requests. For details on creating navigation sections, see [Configure an icon UI section](#).

---

## Configure legacy icon sections

Use the legacy icon section type to enable users to easily access your screens. Select how you want to visually represent this option.

**Before you begin**

Role required: admin

**About this task**

**Note:** Beginning with the Rome release, some terminology has changed for certain ServiceNow® mobile products. This topic content references Mobile Studio, which retains the terminology used in previous releases. For more information, see [ServiceNow mobile terminology changes](#).

**Procedure**

1. Navigate to **All > System Applications > Studio**.
2. Select your application.
3. In Application Explorer, navigate to **Mobile Studio > Applet Launchers** and select **Applet Launchers**.
4. Open your applet launcher.
5. In the **UI Section** of your applet launcher, click the add button (✚) to create a new icon UI section.
6. On the form, fill in the fields.
### UI section form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section Name</td>
<td>Name for the UI section.</td>
</tr>
<tr>
<td>Section Type</td>
<td>Type of UI section. For an icon section, select <strong>Icon</strong>.</td>
</tr>
<tr>
<td>Orientation</td>
<td>Orientation of the UI section. You can select either <strong>Horizontal</strong> or <strong>Vertical</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>• Use a vertical view when you want all scores to be visible on the screen without scrolling, or when your titles or values are long</td>
</tr>
<tr>
<td></td>
<td>• Use a horizontal view when you want to display several scores in a scrollable section.</td>
</tr>
<tr>
<td>Hide when empty</td>
<td>Option for hiding empty UI sections when there is no content to display.</td>
</tr>
<tr>
<td>Select the applets to display</td>
<td>Applets that you select from the Selected Applets list to appear in the UI section.</td>
</tr>
</tbody>
</table>

7. Select **Save**.

### Configure record sections

Use the record section type to display records from a selected list as cards with important information. These cards enable users to easily access your record screens. Select how you want to represent this option visually.

**Before you begin**
Role required: admin

**About this task**

**Note:** Beginning with the Rome release, some terminology has changed for certain ServiceNow® mobile products. This topic content references Mobile Studio, which retains the terminology used in previous releases. For more information, see ServiceNow mobile terminology changes

**Procedure**

1. Navigate to **All > System Mobile > Mobile App Builder**.
   The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.

2. Search for the application scope you are working in and then select the name of the application scope.
   The Mobile App Builder categories home screen displays.

3. Select **All mobile records** from the menu.

4. From the **Record type** field, select **UI Section [sys_sg_section]**, and then select **New**.

5. From the Create a section dialog box, select **Record section** and then select **Continue**.
6. Complete the following fields as needed.

**UI section form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties section</strong></td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td>Name for the record UI section.</td>
</tr>
<tr>
<td>Active</td>
<td>Toggle that sets whether the record section is available or not.</td>
</tr>
<tr>
<td><strong>Settings section</strong></td>
<td></td>
</tr>
<tr>
<td>Hide header</td>
<td>Header of this UI section is hidden when this field is enabled. When enabled, the title and the See all button are both hidden.</td>
</tr>
<tr>
<td>Display title</td>
<td>Title of the section is visible. This option is available only when the header is visible. The title of the section comes from the title of your record section.</td>
</tr>
<tr>
<td>Hide &quot;See all&quot; button</td>
<td>The See all button within the header is hidden when this field is enabled. This field is available only when the header is visible.</td>
</tr>
<tr>
<td>Orientation</td>
<td>Orientation of the record section. You can select either Horizontal or Vertical based on the amount of space available on your screen.</td>
</tr>
<tr>
<td>Hide section if empty</td>
<td>Option for hiding empty UI sections when there's no content to display.</td>
</tr>
<tr>
<td>Max items display count</td>
<td>Maximum number of cards that are displayed on the launcher screen. Enter a number in this field. If the number of available records in the destination screen exceed the number set in this field, users must tap See all in the header to see the remaining records.</td>
</tr>
<tr>
<td>Destination screen</td>
<td>Destination screen determines which records are displayed as cards in the record section. Clicking the card enables you to navigate to the record screen for a record. Selecting See all displays the full list screen you selected for your destination screen.</td>
</tr>
<tr>
<td>Role access</td>
<td>User roles that can access this UI section. This field is an optional setting.</td>
</tr>
</tbody>
</table>

7. Select Save.

**Configure additional capabilities in a record section**

Define additional record section capabilities like the number of rows, specifying the card size, and the type of card-swiping action. These features are configured in the web-based UI as opposed to the Mobile App Builder.
Before you begin
Role required: admin

Procedure
1. Navigate to All.
2. In the filter navigator, enter sys_sg_item_section.list, to open a list of record sections.
3. Select a displayed record section.
4. On the form, fill in the fields.

Note: This table lists all features that are not yet available for configuration in the Mobile App Builder. For configurations available within Mobile App Builder, see Configure record sections.

Record section form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scrolling type</td>
<td>Method by which users scroll through the mobile cards displayed in a row. The options are:</td>
</tr>
<tr>
<td></td>
<td>• Free scroll - Allows the user to scroll multiple cards with a single swipe.</td>
</tr>
<tr>
<td></td>
<td>• Snap to Start - Causes the user to scroll one card at a time. This field is available when you select Horizontal in the Orientation field.</td>
</tr>
<tr>
<td>Is custom size</td>
<td>Enables you to define the specific dimensions of cards within the record section. If this field is selected the Card size field does not display, instead the Custom height and Custom width fields display.</td>
</tr>
<tr>
<td>Custom height</td>
<td>The height of cards in the record section. The range is from 42 through 296 pixels. This field is available when you select the field Is custom size.</td>
</tr>
<tr>
<td>Custom width</td>
<td>The width of cards in the record section. The range is from 64 through 304 pixels. This field is available when you select the field Is custom size.</td>
</tr>
<tr>
<td>Card size</td>
<td>Size of card displayed in all rows. The following card sizes, in pixels, are available:</td>
</tr>
<tr>
<td></td>
<td>• Small: (width 304 x height 108)</td>
</tr>
<tr>
<td></td>
<td>• XSmall: (width 148 x height 184)</td>
</tr>
<tr>
<td></td>
<td>• Medium: (width 304 x height 148)</td>
</tr>
<tr>
<td></td>
<td>• Large: (width 304 x height 243)</td>
</tr>
<tr>
<td></td>
<td>• XLarge: (width 304 x height 314)</td>
</tr>
<tr>
<td></td>
<td>If the listed card sizes do not meet your requirements, you can customize a card size. For more information, see Customize a card size for a record section.</td>
</tr>
<tr>
<td></td>
<td>This field is not available if you select the field Is custom size.</td>
</tr>
<tr>
<td>Row count</td>
<td>The number of rows displayed in each record section. You can have from 1 through 3 rows. Selecting None displays 1 row. This field is available when you select Horizontal in the Orientation field.</td>
</tr>
<tr>
<td>Column count</td>
<td>The number of columns displayed in each record section. You can display from 1 through 3 columns. Selecting None displays 1 column. This field is available when you select Vertical in the Orientation field.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Wide column count</td>
<td>This number of columns displayed in each record section for either larger devices like tablets or in landscape mode. You can display from 1 through 4 columns. Selecting None displays 1 column. This field is available when you select Vertical in the Orientation field.</td>
</tr>
</tbody>
</table>

Note: The selection made in the Wide column count field must be greater than the number in the Column count field.

5. Right-click in the header and select Save.

Customize a card size for a record section

Customize the size of cards in a record section if the default options do not suit your requirements.

Before you begin
Role required: admin

About this task

Note: You can also customize the card size by selecting the Is custom size field from the Record section form, and then define the card height and width in pixels.

Procedure

1. Navigate to All.
2. In the filter navigator, enter sys_sg_item_section.list, to open a list of record sections.
3. Select a listed record section.
4. From the Card size field, select the reference lookup icon (🔍).
5. In the Card sizes form, select New.
6. On the form, fill in the fields.

Card sizes form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card height</td>
<td>The height of the cards in the record section. The range is from 42 through 296 pixels.</td>
</tr>
<tr>
<td>Card width</td>
<td>The width of cards in the record section. The range is from 64 through 304 pixels.</td>
</tr>
<tr>
<td>Type</td>
<td>Name of the defined card type size.</td>
</tr>
</tbody>
</table>

7. Right-click in the header and select Save.

Configure chart UI sections

Use the chart UI section type to display dashboard previews that enable users to navigate to reports and Performance Analytics chart widgets.

Before you begin
Role required: admin
Procedure

1. Navigate to All > System Applications > Studio.
2. Select your application.
3. In Application Explorer, navigate to Mobile Studio > Applet Launchers.
4. Open your applet launcher.
5. In the UI section of your applet launcher, click the add button (+) to create a new chart UI section.
6. On the form fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section Name</td>
<td>Name for the UI section.</td>
</tr>
<tr>
<td>Section Type</td>
<td>Type of UI section.</td>
</tr>
<tr>
<td>Orientation</td>
<td>Orientation of the UI section. You can select either Horizontal or Vertical.</td>
</tr>
<tr>
<td>Hide when empty</td>
<td>Option for hiding empty UI sections, when there is no content to display.</td>
</tr>
<tr>
<td>Select the dashboard preview</td>
<td>Determines the applets to display in the dashboard preview of the chart UI section. Select the required applets and click the arrow to move them to the selected column.</td>
</tr>
</tbody>
</table>

Note:
- Use a vertical view when you want all scores to be visible on the screen without scrolling or your titles or values are long.
- Use a horizontal view when you want to display several scores in a scrollable section.

7. Click Save.

Results

A dashboard preview is displayed in your applet launcher. Tap on the preview to navigate to a preconfigured chart. See Mobile platform dashboards for examples of the charts you can utilize.
Configure an icon UI section

Use the icon UI section type to navigate to screens, launcher screens, and functions.

Before you begin
Role required: admin

Procedure
1. Navigate to All > System Mobile > Mobile App Builder. The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.
2. Search for the application scope you are working in and then select the name of the application scope. The Mobile App Builder categories home screen displays.
3. Select All mobile records from the menu.
4. From the Record type list, select UI Section [sys_sg_section], and then select New.
5. From the Create a section dialog box, select Icon section and then select Continue.

6. On the form, fill in the fields.

<table>
<thead>
<tr>
<th>UI section form</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties section</strong></td>
</tr>
<tr>
<td>Title</td>
</tr>
<tr>
<td>Active</td>
</tr>
<tr>
<td><strong>Settings section</strong></td>
</tr>
<tr>
<td>Hide header</td>
</tr>
<tr>
<td>Orientation</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Hide section if empty</td>
</tr>
<tr>
<td>Icon section destination</td>
</tr>
<tr>
<td>Role access</td>
</tr>
</tbody>
</table>

7. Select Save.
Configure additional capabilities in an icon section

Define additional icon section capabilities like the type of item to display (icon, image, text-only), the number of columns, and the settings for larger screens. These features are configured in the web-based UI as opposed to the Mobile App Builder.

**Before you begin**
Role required: admin

**Procedure**
1. Navigate to **All**.
2. In the filter navigator, enter `sys_sg_navigation_section.list`, to open a list of icon sections.
3. Select a listed icon section.
4. On the form, fill in the fields.

**Note:** This table lists all features that are not yet available in the Mobile App Builder. For configurations available within Mobile App Builder, see Configure an icon UI section.

### Icon section form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display feed count</td>
<td>Displays a count of records within each screen. Use this option with both vertical and horizontal orientation.</td>
</tr>
<tr>
<td>Display type</td>
<td>Option that defines whether the icon section contains images, icons, or cards with text (basic). Use the basic option if you want to emphasize the count number.</td>
</tr>
<tr>
<td>Column count</td>
<td>Number of columns displayed in each icon section. You can display from 1 to 3 columns. Selecting <strong>None</strong> displays 1 column. This field is available when you select <strong>Vertical</strong> in the Orientation field.</td>
</tr>
<tr>
<td>Wide column count</td>
<td>Number of columns displayed in each icon section for either larger devices like tablets or in landscape mode. You can display from 1 to 4 columns. Selecting <strong>None</strong> displays 1 column. This field is available when you select <strong>Vertical</strong> in the Orientation field. <strong>Note:</strong> The selection made in the Wide column count field must be greater than the number in the Column count field.</td>
</tr>
</tbody>
</table>

5. Right-click in the header and select **Save**.
6. In the **Icon section destination** area, select **Icon section destination** from the dialog box.
7. Select **New**.
The Icon section destinations table displays.
8. On the form, fill in the fields.

### Icon section destination

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Icon Section</td>
<td>This field is automatically set with the name of the icon section.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order</td>
<td>The order in which the icon section destination displays. For example, an entry of 100 would place this icon section destination before an icon section with an entry of 200.</td>
</tr>
</tbody>
</table>

9. Using the reference lookup icon (🔍) in the **Icon Section Destination** field, create a destination for the **Display type** you selected in the Icon section form.

   a. Select **New** in the Icon section destinations form.

   b. From the Select Icon Section Destination Type dialog box, select either function, launcher, or screen as the destination type, and then select **OK**.

   c. Enter a name for the icon section destination function.

   d. Use the reference lookup icon to select either a function, launcher screen, or screen depending on your selection made from the Select Icon Section Destination Type dialog box.

   e. Depending on the **Display type** you selected, complete one of these fields:

      - **Icon** - Use the reference lookup icon to select an existing icon to display in the icon section.
      - **Image** - Use the reference lookup icon to select the image to display in the icon section. For instructions on how to include images that you select from this field, see **Adding images to an icon section**.
      - **Background color** - Enter a hexadecimal color value as the text banner's background color to display in the icon section.

      **Note:** If the relevant field is empty, a default icon, image, or background color is used.

   f. Select **Submit**.

   g. Right-click in the header and select **Save**.

### Adding images to an icon section

Upload images or icon-style images to use within your icon sections. Use personalized images to brand your mobile application. This feature is configured in the web-based UI as opposed to the Mobile App Builder.

**Before you begin**

Role required: admin

**Procedure**

1. Upload an image to the Icon images table and access the sys_id of the image.

   a. Navigate to **All**.

   b. In the filter navigator, enter `sys_sg_icon_image.list`, to open a list of icon images.

   c. Select **New**.
d. Enter a name for your image.

e. Select **Click to add** to browse and select an image to upload.

f. Select **Update**.
   Your image is added to the Icon images table.

g. Right-click on the element name and select **Copy sys_id**.

2. Create an image-type icon from the image you selected.

   a. Navigate to **All**.

   b. In the filter navigator, enter `sys_sg_icon.list`, to open the Icons table.

   c. Select **New**.

   d. Enter a name for your icon of type image.

   e. In the **Icon Name** field, enter the attribute name `ImageRef`.

   f. Right-click in the **Icon Value** field and paste the `sys_id` of the element name you selected.

   g. Select **Image** in the **Type** field.

   h. Select **Submit**.

3. Map the defined image to an icon section.

   a. Navigate to **All**.

   b. In the filter navigator, enter `sys_sg_navigation_section_destination.list`, to open the Icon section destinations table.

   c. Do one of the following.
      - Select a displayed icon section destination from the list.
      - Select **New** and then from the Select Icon Section Destination Type menu, select either launcher, screen, or function option.

   d. In the **Image** field, use the reference lookup icon to select the icon type image you created.

   ️ **Note:** For details to complete the rest of the Icon section destination type table, see Configure additional capabilities in an icon section.

   e. Select **Submit**.

**Configure content UI sections**

Use the content UI section type to display text, images, and videos as campaign cards at the top of your screen. Use these campaigns to deliver messages and important information to your users.

**Before you begin**

Role required: admin
Procedure

1. In the web-based UI, enter `sys_sg_content_section.list` in the filter navigator to open a list of content sections.
2. Click New to create a content section.
3. On the content section form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>A title for the content UI section. This title is not displayed in the mobile UI.</td>
</tr>
<tr>
<td>Active</td>
<td>Select whether to display the campaign in your mobile instance.</td>
</tr>
</tbody>
</table>

4. Click Submit.
5. Add your UI section to an applet launcher:

   - **Note:** Content, media, and navigation UI sections are added to applet launchers in the applet launcher record within the web-based UI. Icon, item, and chart UI sections are added to applet launchers within mobile Studio.

   a. Navigate to System Mobile > Applet Launchers.
   b. Open the applet launcher record for the launcher where you want to add your UI section.
   c. Click the Body tab.
   d. In the Applet Launcher Sections list, click the Insert a new row link.

      - **Note:** If the Insert a new row link does not appear, you might not be in the same application as the applet launcher record you are editing. Switch to the appropriate application and reload the form.

   e. In the Order field, enter a number representing the order in which the UI section is displayed on the applet launcher. UI sections appear on the launcher from the lowest order to the highest.
   f. Press the Tab key to move to the Section field.
   g. In the Section field, select the UI section and click the green check.
   h. After you have added your UI section, click Update to save the applet launcher record.

Results

The content UI section you created can now contain a mobile campaign. For full instructions, see Configure mobile campaign components.

Configure media UI sections

Configure a media UI section type to display images or videos on your applet launcher.
Before you begin
Role required: admin

Procedure
1. In your instance, enter sys_sg_section.list in the filter navigator.
2. In the mobile sections list click New.
3. In the Select section Type pop-up, select Media Section.
4. Click OK.
The instance displays a media section form.
5. In the media section form, fill in the fields as needed.

### Media section form

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Title for your media section</td>
</tr>
<tr>
<td>Active</td>
<td>Whether the record is active. Inactive records do not display in your app.</td>
</tr>
<tr>
<td>Application</td>
<td>The scoped application for this media section. This field is automatically populated with the current scoped application.</td>
</tr>
<tr>
<td>Required Roles</td>
<td>The roles required to see the media section. If this field is left empty, there are no role requirements.</td>
</tr>
<tr>
<td>Headline</td>
<td>Headline for the media section. This headline appears below your image or video.</td>
</tr>
<tr>
<td>Text</td>
<td>Text for the media section. This text appears below the text of your headline.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of media section select one of the following:</td>
</tr>
<tr>
<td></td>
<td>• Image</td>
</tr>
<tr>
<td></td>
<td>• Video</td>
</tr>
<tr>
<td>Image</td>
<td>Image that appears in the media section. This field appears only if the Type field is set to Image.</td>
</tr>
<tr>
<td>Field</td>
<td>Value</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Note:</td>
<td>Images used in your media section must be attached to your media section record. Drag your image onto the media section form to attach, then select the image using the reference lookup icon (🔍). Create images optimized for use in media sections by using a height of 160px and a width of 1366px. If your users primarily use iPhones, use a height of 160px and a width of 375px to 600px.</td>
</tr>
<tr>
<td>Video</td>
<td>Video that appears in the media section. This field appears only if the Type field is set to Video. This video must be a URL for an externally hosted YouTube or Vimeo video.</td>
</tr>
<tr>
<td>Function instance</td>
<td>Select the function instance that is used when a user taps on the media section. The label of your function appears below the text in the Text field.</td>
</tr>
<tr>
<td>Note:</td>
<td>The selected function instance must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>• The Parent table field must contain entries, before selecting the Parent field.</td>
</tr>
<tr>
<td></td>
<td>• The Parent table field must be set to Applet Launcher.</td>
</tr>
<tr>
<td></td>
<td>• The Parent field of the selected function instance must have be the launcher where you intend to add your media section.</td>
</tr>
<tr>
<td></td>
<td>• The Location field of the function instance must be Media Section.</td>
</tr>
<tr>
<td></td>
<td>• The Parent table field must contain entries, before selecting the Parent field.</td>
</tr>
<tr>
<td></td>
<td>◦ The Parent table field must be set to Applet Launcher.</td>
</tr>
<tr>
<td></td>
<td>◦ The Parent field of the selected function instance must have be the launcher where you intend to add your media section.</td>
</tr>
<tr>
<td></td>
<td>◦ The Location field of the function instance must be Media Section.</td>
</tr>
</tbody>
</table>

6. Click Submit.
You have a configured media section that you can add to an applet launcher.

7. Add your UI section to an applet launcher:

**Note:** Content, media, and navigation UI sections are added to applet launchers in the applet launcher record within the web-based UI. Icon, item, and chart UI sections are added to applet launchers within mobile Studio.

a. Navigate to **System Mobile > Applet Launchers**.

b. Open the applet launcher record for the launcher where you want to add your UI section.

c. Click the **Body** tab.

d. In the **Applet Launcher Sections** list, click the **Insert a new row** link.

**Note:** If the Insert a new row link does not appear, you might not be in the same application as the applet launcher record you are editing. Switch to the appropriate application and reload the form.

e. In the **Order** field, enter a number representing the order in which the UI section is displayed on the applet launcher.
UI sections appear on the launcher from the lowest order to the highest.

f. Press the Tab key to move to the **Section** field.

**Quick actions**

Define quick actions for your applet launcher so that your users can access commonly used functions, such as transferring records or using a template in the mobile app.

**Before you begin**

Role required: admin
About this task

Define quick actions for an applet launcher so that your users can access commonly used functions.

Users access quick actions tapping the quick action icon.

For Android users, this icon appears as a plus symbol (+) at the bottom right corner of the applet launcher.

For iOS users this icon appears as an ellipsis symbol (…).

Procedure

1. In Studio, navigate to

2. Navigate to Mobile Studio > Applet Launchers, and select the applet launcher where you want to add a quick action.

3. In the Quick Actions list, click the add button (+).

4. Fill in the fields in the Create a Quick Action window as needed.

Create a Quick Action form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Enter a descriptive label for the quick action</td>
</tr>
<tr>
<td>Icon</td>
<td>Select an icon for the quick action. For more information on mobile icons, see Mobile icons.</td>
</tr>
<tr>
<td>Function</td>
<td>Select a function to use in the quick action, or click the add button (+) to create a new function. For details on creating functions, see Mobile functions.</td>
</tr>
</tbody>
</table>

Note: After you have created your quick action, you can change these values by clicking on the quick action in the Quick Actions list.
5. Optional: Click and drag the selection icon (⿴) to change the order of the quick actions.

6. Optional: Click the delete icon (🗑️) to remove a quick action from your application launcher.

What to do next
Add your applet launcher to the navigation bar to make it accessible to your users. For detail on navigation bar configuration, see Configure the navigation bar.

Data items
Data items provide the data presented in a screen.

Note: Beginning with the Rome release, some terminology has changed for certain ServiceNow® mobile products. This topic content references Mobile Studio, which retains the terminology used in previous releases. For more information, see ServiceNow mobile terminology changes.

Data items are queries on a selected table that bring back data. Screens, which display data from your instance, get their information from data items. For example, if your application contains a list of incidents, the list would use a data item representing records from the incident [incident] table.

Data item types
There are three different types of data items.

Standard data items
A standard data item is used to return data from a single table. A standard data item does not include parameters, but can include a filter to return a set of results from the queried table. For example, a standard data item could query all high priority changes from the change request [change_request] table.

For information on standard data items, see Configure a standard data item.

Parametrized data items
A parametrized data item is used to perform a query containing one or more parameters to return data. Parameters are used to pass in variables to your queries. For example, you could use a parameter to include the value of the logged in user so that you can show a list of all incidents assigned to that user.

For details on parametrized data items, see Configure a parametrized data item.

For a detailed walk-through of a parametrized data item configuration, see Tutorial: Configure a data item with parameters.

Relationship data items
Relationship data items are necessary for embedded lists. Embedded lists are lists that can be embedded within a record screen. Record screens can have one or more embedded segments. An embedded list is a type of segment that can be embedded. When you configure an embedded list, it's important to be familiar with the original record screen table. It's also important to understand the table that contains the records that appear in the embedded list. To create an embedded list, you need a relationship data item. For details about creating a relationship data item, see Configure a relationship data item for an embedded list.
Special use data item configurations

The following data items do not fit into the previous categories, and are used only in the cases detailed in the following sections.

Append encoded query

To navigate from a single score analytics preview or from a chart screen header function to a list of records, you need to configure a special data item using an instance relative URL. This is a standard data item that uses a different type of condition. Find more information about how to configure an append encoded query in Chart screen.

Grouped list data item

A grouped data item is used exclusively for a grouped list screen. Any data item, standard and relationship, parametrized or not, can be grouped by a field. Find more info about how to configure grouped list data items at Grouped list screen.

Configure a standard data item

Configure a standard data item to query data for your screens that do not require parameter input.

Before you begin
Role required: admin

Procedure
1. Navigate to All > System Mobile > Mobile App Builder.
   The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.
2. Search for the application scope you are working in and then select the name of the application scope.
   The Mobile App Builder categories home screen displays.
3. Select the Data category, and then select New.
4. Select Data item and then select Continue.
5. In the Data Items tab, click Create New.
6. Complete the following fields as needed.

Data item fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties section:</strong></td>
<td></td>
</tr>
<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 it easily.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional additional details about the data item.</td>
</tr>
<tr>
<td><strong>Data section:</strong></td>
<td></td>
</tr>
<tr>
<td>Table</td>
<td>The table you want the data item to pull information from.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>Custom tables are not available by default. You can change this behavior by modifying the subscription.custom_table.enforce_entitlement system property. For details on making this change, see Allow or restrict access to custom tables in mobile data items.</td>
</tr>
<tr>
<td>Group by</td>
<td>Groups query results based on the selected field from the menu.</td>
</tr>
<tr>
<td>Condition type</td>
<td>Determines what type of condition your data item uses. Select from: Declarative Use a declarative condition to create conditions for the data item using the condition builder. Scripted Use a script to determine the conditions of your data item. A text window to enter a script appears below this field when you select this option. Append Encoded Query Use this option only when creating data items for your chart screens. For details on that data item type, see Configure an encoded query data item for chart screens.</td>
</tr>
<tr>
<td>Condition</td>
<td>Set of conditions for the data item to conform to. You can create conditions using the condition builder.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>This field is only available when you select Declarative in the Condition type field.</td>
</tr>
<tr>
<td>Sort by</td>
<td>Lets you adds fields to the form that allow you to configure how to sort the list. In the condition builder, select the field you want the list to be sorted by. For example, select Caller. Then select ascending or descending to determine the order of the list.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>This field is only available when you select Declarative in the Condition type field.</td>
</tr>
</tbody>
</table>

**7. Optional:** If you have selected Scripted in the Condition type field, you must create a script in the Query Condition Script field.
Your scripted condition must return a query string, which the instance uses to filter the data item. Use the input variable to access information from the current record.

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

This example uses the input variable and filters the data item for records matching the current records company and location. It then appends the text `ORDERBYDESCsys_updated_on` to the query, which sorts the data item records by the `Updated on` field.

8. Select Save.

What to do next
Associate a data item with a screen. For more information, see Mobile screens.

Related information
- Configure a parametrized data item
- Configure a group by data item

Configure a parametrized data item
Configure a parametrized data item to filter and view just the relevant data according to the selected parameters.

Before you begin
Role required: admin

About this task

ℹ️ Note: Beginning with the Rome release, some terminology has changed for certain ServiceNow® mobile products. This topic content references Mobile Studio, which retains the terminology used in previous releases. For more information, see ServiceNow mobile terminology changes.

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

Procedure
1. Navigate to All > System Mobile > Mobile App Builder.
   The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.
2. Search for the application scope you are working in and then select the name of the application scope.
The Mobile App Builder categories home screen displays.

3. Select the **Data** category from the menu, and then select **New**.

4. From the Create a data item dialog box, select the type of data item you want to create, and then select **Continue**.
   Both the standard and relationship data items can include parameters.

5. Complete the Properties and Data sections as needed.
   For more information about creating a data item, see Configure a standard data item. For example, create a data item for open incidents.

   **Note:** To configure the Condition section you must first complete and save the parameter configuration.

6. In the Parameters section, select **New**.

7. In the Data Parameter screen, in the **Name** field, enter a name for the parameter.
   Parameter names correlate most often with fields on a form. For example, type Priority as the parameter name when the field refers to priority.

8. From the Type field, select the data type of the 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:
   - **Integer**: Opens a numbers-only keypad
   - **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.
   - **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

9. Select **Save**.

10. In the Data item form, in the Condition section, add a query condition for your parameter.
    The condition field should match the parameter for which you are querying the database. For example, if you are creating a data item to query the Priority field, create a condition for Priority is {{priority}}. Make sure that you select the parameter that you created by selecting the search icon (🔍) in the condition builder.

    **Parameterized data item form**
11. Select **Save**.

**What to do next**

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

**Assign a data item with parameters to a list screen**

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

**Before you begin**

Make sure that you have configured a data item that has parameters. For more information, see **Configure a parametrized data item**.

Role required: admin

**About this task**

**Note:** Beginning with the Rome release, some terminology has changed for certain ServiceNow® mobile products. This topic content references Mobile Studio, which retains the terminology used in previous releases. For more information, see **ServiceNow mobile terminology changes**.
Procedure

1. Navigate to All > System Mobile > Mobile App Builder.
   The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.

2. Search for the application scope you are working in and then select the name of the application scope.
   The Mobile App Builder categories home screen displays.

3. Select the Screens category, and then select New.

4. Select the List option in the Create a screen page, and then select Continue.

5. Add a preconfigured parameterized data item to a list screen.
   - In the Screen segments section, select New.
     A new screen section displays within the list screen record.
   - In the Streams section of the New screen segment configuration panel, select New.
     A new list screen section displays.
   - In the Data Item field of your new list stream section, select Choose.
   - Select the parametrized data item that you created, and select Apply.

6. Select the List screen record in the configuration tree.

7. In the UI Parameters section, select New.

8. In the New UI parameter screen, 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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties</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>
<tr>
<td>Display name</td>
<td>Name that displays. This name automatically matches the input name.</td>
</tr>
<tr>
<td><strong>Settings</strong></td>
<td></td>
</tr>
<tr>
<td>Parameter type</td>
<td>Type of parameter. In Mobile App Builder, this field is automatically set to Screen when a new parameter is created from a screen record.</td>
</tr>
<tr>
<td>Screen</td>
<td>Screen record. In Mobile App Builder, this field is automatically set to the parent screen record.</td>
</tr>
<tr>
<td>Input style</td>
<td>How the user interacts with the UI. Choose from Inline or Popup. For this example, choose Inline.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>Whether the user is required to enter information for that field. For this example, leave this check box cleared.</td>
</tr>
<tr>
<td>Field</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Placeholder text</td>
<td>Text that appears below the field type. This option does not appear if you have a default value selected.</td>
</tr>
<tr>
<td>Order</td>
<td>Field used for input ordering.</td>
</tr>
<tr>
<td>Input source</td>
<td>Determines whether the input is populated by the end user or automatically by the system. For this example, choose User input.</td>
</tr>
<tr>
<td>Input type</td>
<td>How the end-user input is added to the system. For example, if you want to add a parameter for the Assigned to field, select Choice list so that users have a list of options to choose from. Select from one of the following options:</td>
</tr>
<tr>
<td></td>
<td>• None: There is no input type.</td>
</tr>
<tr>
<td></td>
<td>• Text: Provides a simple text field. This option works best for fields that require free text input. For example, work notes or resolution details. This is the default input type.</td>
</tr>
<tr>
<td></td>
<td>• Choice list: Opens a list for the end user to select from. This option works well for reference fields that require specific information.</td>
</tr>
<tr>
<td></td>
<td>     Note: The choice list input type returns a maximum of 1,000 results. For references that require more than 1,000 results, use the Search list input type.</td>
</tr>
<tr>
<td></td>
<td>• Search list: Provides a search bar so that end users can search in a list.</td>
</tr>
<tr>
<td></td>
<td>• QR/Barcode: Provides the option to search by QRC or barcode.</td>
</tr>
<tr>
<td>Carried</td>
<td>Use carried parameters to move information between different screens and actions.</td>
</tr>
<tr>
<td>Data section</td>
<td></td>
</tr>
<tr>
<td>Table</td>
<td>The table name for which you want to create a UI parameter. For example, Incident. This field only appears if you select Choice list or Search list as the input type.</td>
</tr>
<tr>
<td>Field</td>
<td>The field name for which you want to create a UI parameter. For example, Priority. This field only appears if you select Choice list or Search list as the input type.</td>
</tr>
<tr>
<td>Default value type</td>
<td>The value that appears by default in the UI field. The Default value type field only appears for Function type parameters. 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: 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>
</tbody>
</table>

9. In the Screen Data Parameter Mapping section, select Choose.

10. From the Choose an item menu, select the data parameter you created with the data item, and then select Apply.
11. Select **Save**.

12. **Optional:** If you want the field on the mobile screen to be automatically populated with a value, configure the auto-fill parameters.

   a. In the **Input source** field for the UI parameter record, select **Auto fill**.

   b. 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.
      - **Constant**: Automatically inputs a constant value.
      - **Source field**: Automatically inputs a value from a table field.
      - **User**: Automatically inputs the currently logged in user.
      - **Append encoded query**: Automatically inputs data from an encoded query.

   c. Select **Save**.

13. Complete any additional screen fields as needed. For more information on creating a screen, see [Create a screen](#).

14. Select **Save**.

**Assign a data item with parameters to a record screen**

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

**Before you begin**

Make sure that you have configured a data item that has parameters. For more information, see [Configure a parametrized data item](#).

Role required: admin

**About this task**

**Note:** Beginning with the Rome release, some terminology has changed for certain ServiceNow® mobile products. This topic content references Mobile Studio, which retains the terminology used in previous releases. For more information, see [ServiceNow mobile terminology changes](#).

**Procedure**

1. Navigate to **All > System Mobile > Mobile App Builder**.
   
   The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.

2. Search for the application scope you are working in and then select the name of the application scope.
   
   The Mobile App Builder categories home screen displays.

3. Select the **Screens** category, and then select **New**.

4. Select the **Record** option in the Create a screen page, and then select **Continue**.

5. In the **Data item** field, select **Choose**.

6. From the Choose an item menu, select the parameterized data item that you created, and select **Apply**.
7. In the UI parameters section of your record screen, select **New**.

8. In the New UI parameter screen, 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.

**UI Parameter form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties section</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>
<tr>
<td>Display name</td>
<td>Name that displays. This name automatically matches the input name.</td>
</tr>
<tr>
<td><strong>Settings section</strong></td>
<td></td>
</tr>
<tr>
<td>Parameter type</td>
<td>Type of parameter. In Mobile App Builder, this field is automatically set to <strong>Screen</strong> when a new parameter is created from a screen record.</td>
</tr>
<tr>
<td>Screen</td>
<td>Screen record. In Mobile App Builder, this field is automatically set to the parent screen record.</td>
</tr>
<tr>
<td>Input style</td>
<td>How the user interacts with the UI. Choose from <strong>Inline</strong> or <strong>Popup</strong>. For this example, choose <strong>Inline</strong>.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>Whether the user is required to enter information for that field. For this example, leave this check box cleared.</td>
</tr>
<tr>
<td>Placeholder text</td>
<td>Text that appears below the field type. This option does not appear if you have a default value selected.</td>
</tr>
<tr>
<td>Order</td>
<td>Field used for input ordering.</td>
</tr>
<tr>
<td>Input source</td>
<td>Determines whether the input is populated by the end user or automatically by the system. For this example, choose <strong>User input</strong>.</td>
</tr>
<tr>
<td>Input type</td>
<td>How the end-user input is added to the system. For example, if you want to add a parameter for the Assigned to field, select <strong>Choice list</strong> so that users have a list of options to choose from. Select from one of the following options:</td>
</tr>
</tbody>
</table>

  - **None**: There is no input type.
  - **Text**: Provides a simple text field. This option works best for fields that require free text input. For example, work notes or resolution details. This is the default input type.
  - **Choice list**: Opens a list for the end user to select from. This option works well for reference fields that require specific information.

    **Note**: The choice list input type returns a maximum of 1,000 results. For references that require more than 1,000 results, use the **Search list** input type.
<table>
<thead>
<tr>
<th>Field</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Search list</strong></td>
<td>Provides a search bar so that end users can search in a list.</td>
</tr>
<tr>
<td><strong>QR/Barcode</strong></td>
<td>Provides the option to search by QRC or barcode.</td>
</tr>
<tr>
<td>Carried</td>
<td>Use carried parameters to move information between different screens and actions.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>The table name for which you want to create a UI parameter. For example, Incident. This field only appears if you select <strong>Choice list</strong> or <strong>Search list</strong> as the input type.</td>
</tr>
<tr>
<td><strong>Field</strong></td>
<td>The field name for which you want to create a UI parameter. For example, Priority. This field only appears if you select <strong>Choice list</strong> or <strong>Search list</strong> as the input type.</td>
</tr>
<tr>
<td><strong>Default value type</strong></td>
<td>The value that appears by default in the UI field. The Default value type field only appears for Function type parameters. Select one of the following options.</td>
</tr>
<tr>
<td></td>
<td>• <strong>None</strong>: There is no default text. This option works well for a list input type.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Manual</strong>: An additional field appears for you to enter a default term. For example, <strong>Search for a field</strong>. The manual default works well for search or text input types.</td>
</tr>
</tbody>
</table>

9. In the Screen data parameter mapping section, select **Choose**.

10. From the Choose an item menu, select the data parameter you created with the data item, and then select **Apply**.

11. Select **Save**.

12. **Optional**: If you want the field on the mobile screen to be automatically populated with a value, configure the auto-fill parameters.

   a. In the **Input source** field for the UI parameter record, select **Auto fill**.

   b. 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.
   • **Constant**: Automatically inputs a constant value.
   • **Source field**: Automatically inputs a value from a table field.
   • **User**: Automatically inputs the currently logged in user.
   • **Append encoded query**: Automatically inputs data from an encoded query.

   c. Select **Save**.

13. Complete any additional screen fields as needed. For more information, see Create a screen.

14. Select **Save**.
Configure a relationship data item for an embedded list

Configure relationship data items so you can control the information that appears in embedded lists within a record screen.

Before you begin
Role required: admin

About this task
A relationship data item is required to create an embedded list. Embedded lists are lists that can be embedded within a record screen. For more information on embedded lists in mobile, see Embedded lists for a form screen.

Procedure
1. Navigate to All > System Mobile > Mobile App Builder. The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.
2. Search for the application scope you are working in and then select the name of the application scope. The Mobile App Builder categories home screen displays.
3. Select the Data category from the menu, and then select New.
4. Select the Relationship data item icon, and then select Continue.
5. Complete the following fields as needed.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties section:</strong></td>
<td></td>
</tr>
<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 it easily.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional additional details about the data item.</td>
</tr>
<tr>
<td><strong>Data section:</strong></td>
<td></td>
</tr>
<tr>
<td>Table</td>
<td>The table you want the data item to pull information from. This table is the table of records that is embedded as list on the parent record screen.</td>
</tr>
</tbody>
</table>

**Note:** Custom tables are not available by default. You can change this behavior by modifying the subscription.custom_table.enforce_entitlement system property. For details on making this change, see Allow or restrict access to custom tables in mobile data items.

<p>| Group by                  | Groups query results based on the selected field from the menu. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table related to</td>
<td>Table that the records in this data item are related to. For example, for a list of Task SLA records embedded into a task record screen, the <strong>Tables related to</strong> value would be the Task table.</td>
</tr>
<tr>
<td>Relationship</td>
<td>Relationship between the table selected in the <strong>Table</strong> field and the table selected in the <strong>Table related to</strong> field.</td>
</tr>
<tr>
<td>Condition type</td>
<td>Determines what type of condition your data item uses. Select from: <strong>Declarative</strong> Use a declarative condition to create conditions for the data item using the condition builder. <strong>Scripted</strong> Use a script to determine the conditions of your data item. A text window to enter a script appears below this field when you select this option. <strong>Append Encoded Query</strong> Use this option only when creating data items for your chart screens. For details on that data item type, see Configure an encoded query data item for chart screens.</td>
</tr>
<tr>
<td>Condition</td>
<td>Set of conditions for the data item to conform to. You can create conditions using the condition builder. <strong>Note:</strong> This field is only available when you select <strong>Declarative</strong> in the <strong>Condition type</strong> field.</td>
</tr>
<tr>
<td>Sort by</td>
<td>Lets you add fields to the form that allow you to configure how to sort the list. In the condition builder, select the field you want the list to be sorted by. For example, select Caller. Then select ascending or descending to determine the order of the list. <strong>Note:</strong> This field is only available when you select <strong>Declarative</strong> in the <strong>Condition type</strong> field.</td>
</tr>
<tr>
<td>Parameters</td>
<td><strong>Note:</strong> The Parameters field must not be used for relationship data items.</td>
</tr>
</tbody>
</table>

**6. Optional:** If you have selected **Scripted** in the **Condition type** field, you must create a script in the **Query Condition Script** field.
Your scripted condition must return a query string, which the instance uses to filter the data item. Use the input variable to access information from the current record.

```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);
```

This example uses the input variable and filters the data item for records matching the current records company and location. It then appends the text

`^EQ^ORDERBYDESCsys_updated_on`

to the query, which sorts the data item records by the Updated on field.

7. Select Save.

**What to do next**

After creating a relationship data item, create a list screen using this newly created data item. For more information, see List applet configuration.

**Configure an encoded query data item for chart screens**

Configure an encoded query data item to query data for chart screens. An encoded query data item is a parametrized data item that is the basis for a chart screen to be passed to a list screen. When tapped it enables users to drill-down to a list of items that the chart is created from.

**Before you begin**

Role required: admin

**About this task**

⚠️ **Note:** Beginning with the Rome release, some terminology has changed for certain ServiceNow® mobile products. This topic content references Mobile Studio, which retains the terminology used in previous releases. For more information, see ServiceNow mobile terminology changes

**Procedure**

1. Navigate to All > System Mobile > Mobile App Builder.
   The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.

2. Search for the application scope you are working in and then select the name of the application scope.
   The Mobile App Builder categories home screen displays.

3. Select the Data category, and then select New.

4. From the Create a data item dialog box, select the type of data item you want to create, and then select Continue.
   You can select either a standard Data item or a Relationship data item.

5. Complete the following fields as needed.
## New data item form

<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 it easily.</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>
</tbody>
</table>

**Note:** Custom tables are not available by default. You can change this behavior by modifying the `subscription.custom_table.enforce_entitlement` system property. For details on making this change, see [Allow or restrict access to custom tables in mobile data items](#).

<table>
<thead>
<tr>
<th>Condition type</th>
<th>Determines what type of condition your data item uses. For an encoded query data item, select <strong>Append Encoded Query</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Append Encoded Query</td>
<td>This field only appears after you select <strong>Append Encoded Query</strong> in the <strong>Condition type</strong> field. Leave this field empty. You use this field in later steps.</td>
</tr>
</tbody>
</table>

6. In the Parameters section, select **New**. The Data Parameter form appears.

7. In the **Name** field, enter a name for your parameter. For example, URL.

8. In the **Type** field, select **String**.

9. Select the data item record in the configuration tree to navigate back to the data item record.

10. In the **Append Encoded Query** field, select the parameter you created in the previous steps.

11. Select **Save**.

### What to do next
For more information on how parameters pass values between screens, see [Parametrized list screens](#).

### Configure a group by data item
Configure a data item for a grouped list screen. You group items in a data item to get a count. For example, you can group a list according to priority and then on a grouped list screen see a count of how many high, medium, and low items exist.

### Before you begin
Role required: admin

### About this task
**Note:** Beginning with the Rome release, some terminology has changed for certain ServiceNow® mobile products. This topic content references Mobile Studio, which retains the terminology used in previous releases. For more information, see [ServiceNow mobile terminology changes](#).
Procedure

1. Navigate to All > System Mobile > Mobile App Builder.
   The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.

2. Search for the application scope you are working in and then select the name of the application scope.
   The Mobile App Builder categories home screen displays.

3. Select the Data category, and then select New.

4. From the Create a data item dialog box, select the type of data item you want to create, and then select Continue.
   You can select either a standard Data item or a Relationship data item.

5. Complete the following fields as needed.

   **Data Item form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the data item.</td>
</tr>
<tr>
<td>Table</td>
<td>Table where the data item gets its data.</td>
</tr>
<tr>
<td>Description</td>
<td>Description used to identify your data item.</td>
</tr>
<tr>
<td>Condition Type</td>
<td>Whether the conditions for the data item are declarative, scripted, or use an encoded query. For a grouped list, select either Declarative or Script.</td>
</tr>
</tbody>
</table>

6. In the Data section, use the Group By field to select the field you want to use to group the records in your list.

7. Select Save.

**What to do next**
To see a walk-through example of steps to create a grouped list, see Tutorial: Configure a list of grouped incidents

**Related information**

Grouped list screen
Configure a group list screen

Allow or restrict access to custom tables in mobile data items

Use system properties to control whether custom tables are available when creating or modifying data items.

**Before you begin**
Role required: admin

**Procedure**

1. Type sys_properties.list in the Application Navigator to open a list of your system properties.

2. Find the subscription.custom_table.enforce_entitlement system property.

3. To permit access to custom tables for data items, set the Value field to false, otherwise, set the value to true.
This system property is in the global scope. If you are not in the global scope, you see a prompt at the top of the page. Click the here link to edit the property.

4. Click Update.

**Mobile dashboard preview**

Dashboard previews display a preview of reports and Performance Analytics widgets in your applet launcher.

Mobile dashboard previews display reports or Performance Analytics previews within an applet launcher in chart UI sections. Adding reports and Performance Analytics widgets helps users more easily identify trends and turning points through indicator scores and visual representation. For more details, see Configure chart UI sections.

**Reporting charts**

Dashboard previews support the follow reporting charts:

- Area
- Bar (vertical bar chart)
- Donut
- Line
- Single score
- Spline
- Step line

**Performance Analytics widgets**

Dashboard previews support the latest score Performance Analytics visualizations. For details about this chart, see Displaying Performance Analytics charts for mobile dashboard.

**Create a mobile dashboard preview**

Use a dashboard preview to display reports and user experience Performance Analytics widgets on your applet launcher pages.

**Before you begin**

Role required: admin

**Procedure**

1. Navigate to All > System Applications > Studio.
2. Select your application.
3. In the Application Explorer, navigate to Mobile Studio > Dashboard Previews, and click the popout icon ( ).
4. Click Create New.
   A new Studio tab opens with a dashboard preview form.
5. On the form, fill in the fields.
## Dashboard preview form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name for your dashboard preview.</td>
</tr>
<tr>
<td>Type</td>
<td>Determine whether your preview should display a Report or Performance Analytics widget.</td>
</tr>
<tr>
<td>Report</td>
<td>Report to be displayed in your preview. This field appears only when the Type field is set to Report.</td>
</tr>
<tr>
<td></td>
<td>Click the arrow to display a list of all reports hosted on the instance. If you select a report not supported for mobile, when you open the applet launcher a message is displayed stating that the content cannot be shown.</td>
</tr>
<tr>
<td></td>
<td>For a list of supported charts, see Mobile dashboard preview.</td>
</tr>
<tr>
<td></td>
<td>The displayed report contains the settings and configurations as defined in the instance.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Although pie charts configured on the instance are also supported in mobile, the data is displayed in a donut chart format.</td>
</tr>
<tr>
<td>Widget</td>
<td>Performance Analytics widget to be displayed in your chart preview. This field appears only when Performance Analytics is selected from the Type field.</td>
</tr>
<tr>
<td></td>
<td>Click the arrow to display a list of all widgets hosted on the instance. If you select a Performance Analytics widget not supported for mobile, when you open the applet launcher a message is displayed stating that the content cannot be shown.</td>
</tr>
<tr>
<td></td>
<td>For a list of supported charts, see Mobile dashboard preview.</td>
</tr>
<tr>
<td></td>
<td>The Performance Analytics widget contains the settings and configurations as defined in the instance.</td>
</tr>
<tr>
<td>Auto-create associated Navigation and Applet</td>
<td>Select this option to enable the automatic creation of chart and list applets. These screens display when users tap the chart preview to access additional information.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Note:</td>
<td>To manually create a chart applet and a navigation function, do not select this check box. For more information, see Manually configure chart applets for dashboard preview.</td>
</tr>
</tbody>
</table>

Navigation

Navigation function to use when a user taps the chart preview. This field is not displayed when the Auto-create associated Navigation and applet field is selected.

6. Click **Save**.

**What to do next**

Add your dashboard preview to a chart UI section in an applet launcher. For more information, see Configure chart UI sections.

**Manually configure chart applets for dashboard preview**

Manually define the chart screen to be displayed and the navigation function to use when your users tap on a dashboard preview. Use this configuration option if you are not using automatic creation of associated navigation and applets.

**Before you begin**

Complete the procedure detailed in Create a mobile dashboard preview. Do not select the field Auto-create associated navigation and applet, which enables automatic creation of associated navigation and applets.

Role required: admin

**Procedure**

1. Navigate to **All > System Applications > Studio**.
2. Select your application.
3. In the Application Explorer, navigate to **Mobile Studio > Applets** and select **Applets**.
4. Click the popout icon (†) that appears to the right of **Applets**.
5. In the applets list, click **Create an Applet**.
6. On the form, fill in the fields.

**Create an applet form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the applet, which appears on a tile in the application.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the applet.</td>
</tr>
<tr>
<td>Icon</td>
<td>The icon image and color for your applet, which is displayed in the main Applets page.</td>
</tr>
<tr>
<td>Available offline</td>
<td>Option for making the applet available to users in offline mode.</td>
</tr>
</tbody>
</table>
7. From **Choose a template**, select the chart icon.
8. Click **Create new**.
9. On the **Date and Field** tab of the New Applet form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Type of chart displayed after tapping on the chart section in the applet launcher, either <strong>Report</strong> or <strong>Performance Analytics</strong>.</td>
</tr>
<tr>
<td>Report</td>
<td>Report to be displayed in your preview. This field appears only when <strong>Report</strong> is selected from the <strong>Type</strong> field. Click the arrow to display a list of all reports hosted on the instance. If you select a report not supported for mobile, when you open the applet launcher a message is displayed stating that the content cannot be shown. For a list of supported charts, see <strong>Mobile dashboard preview</strong>. The displayed report contains the settings and configurations defined in the instance.</td>
</tr>
<tr>
<td>Widget</td>
<td>Performance Analytics widget to be displayed in your chart preview. This field appears only when <strong>Performance Analytics</strong> is selected from the <strong>Type</strong> field. Click the arrow to display a list of all widgets hosted on the instance. If you select a Performance Analytics widget not supported for mobile, when you open the applet launcher a message is displayed stating that the content cannot be shown. For a list of supported charts, see <strong>Mobile dashboard preview</strong>. The Performance Analytics widget contains the settings and configurations defined in the instance.</td>
</tr>
<tr>
<td>Set the default view point to the most recent</td>
<td>Area of the report to highlight when the user opens the chart screen. This option changes depending on the report you select. For time series reports, the option <strong>Highlight most recent data</strong> means that the focus point is at the end of the chart. Clear the</td>
</tr>
</tbody>
</table>
Field | Description
--- | ---
 | check box to set the focus point at the start of the chart.
 | For bar reports, the option Highlight right-hand bar means that the focus is on the last bar in the chart. Clear the check box to set the focus on the first bar of the chart.

10. Click Save.

11. On the Functions tab of the New Applet form, add a navigation button by clicking the add button (+). This navigation function is required to navigate to the specified chart screen.

12. In the Chart Screen Header Function screen, from the two lists, select Navigation as the function type and then select a function.

13. Click Save.

**What to do next**
Add your dashboard preview to a chart UI section in an applet launcher. For more information, see Configure chart UI sections.

**Configure chart views for accessibility**
Give users the option to change chart views from color segments to black and white patterns. This option can be used for accessibility purposes.

**Before you begin**
Role required: admin

**About this task**
To grant users the option to view charts in black and white patterns or colored segments, for all chart types. Make the Enable data visualization patterns for charts button available so users can select the button to view charts as black and white patterns. Alternatively, users who do not select the button can view charts as colored segments.

To display the Enable data visualization patterns for charts button, the system property glide.sg.chart_accessibility must be set to true.

**Procedure**
1. Type sys_properties.list in the Filter 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>glide.sg.chart_accessibility</td>
</tr>
<tr>
<td>Type</td>
<td>true</td>
</tr>
<tr>
<td>Value</td>
<td>true</td>
</tr>
</tbody>
</table>

**Results**
Users can select the display option best suited for their requirements.
**Mobile screens**

Learn how to use screens in ServiceNow mobile. Use screens to make your mobile app functionality available to your end users.

**Screen types**

ℹ️ **Note:** Beginning with the Rome release, some terminology has changed for certain ServiceNow® mobile products. This topic content references Mobile Studio, which retains the terminology used in previous releases. For more information, see ServiceNow mobile terminology changes

Screens are made up of one or more embedded screens, which are screens embedded with a record screen. Each screen has configurable components, conditional formatting, sorting order, and filters.

You can choose one of the available screen types within Mobile App Builder. Each screen type provides a unique experience or work flow, typically one per navigation tab, which your users can access on their mobile app. The screen types may include other optional screens for your users to access for additional information. All screens display information that is based on records in a data item.
Launcher screens serve as landing pages or home pages. Using a launcher screen, you can access screens in various formats, as well as search, do quick actions, and find user information.

For more information, see Launcher screens.
List screen

Use list screens to display a list of records queried from a data item. List screens can be configured together with a record screen. When both screens are configured together, your users can tap a record on the list screen to display the record screen for that record.

For more information, see List screen.
**Grouped list screen**

Use a grouped list screen to display a list of records that are grouped by a specific field. A group list screen uses a special group by data item.

For more information, see [Grouped list screen](#).
Record screen

User record screens to display information from one specific record. On a record screen, you can configure up to 4 different types of embedded screens as segments. These 4 different types of embedded screens include details, activity, related lists, and embedded lists.

For more information, see Form screen.
Employee directory screen

This is a pre-formatted list screen created in Mobile Studio, which displays a list of user records. Your user can tap a record on the list to see more information from the record. You can also create functions to let your users swipe records to perform actions on them.

• The employee directory screen shows additional information, such as how to contact a user and where the user is located.

• (Optional) The related lists screen shows the related lists for a selected record.

• Use the employee directory screen in other situations where it is helpful to have a list with associated images. For example, a list of catalog items.

For more information, see Employee directory screen.
**Map screen**

Use a map screen to display a dynamic map that lets your users see the locations that are associated to the records in a data item. Below the map, your users can see data cards that show information about these records. Users can tap a card to see details for the record.

- The record screen shows additional information that you define when a user taps a record.
- (Optional) The activity stream screen shows the activity stream details for a selected record.
- (Optional) The related lists screen shows the related lists for a selected record.

For more information, see [Map screen](#).
Calendar screen

Use a calendar screen to display records associated with date fields. Users can tap a date on the calendar portion of the screen to see records associated with a specific date, in the event stream displayed in the lower portion of the screen. The record screen shows additional information that you define when a user taps a record.

For more information, see Calendar screen.
Mobile web screen

Use a mobile web screen to open a URL from within a ServiceNow app. You can configure relative URLs to open pages within the ServiceNow platform, or an external link. For example, a user can see a knowledge article on the instance via the Service Portal.

Relative URLs that direct your users to open pages within your instance are opened inside the mobile app. URLs that open external links will open the link in the default browser of the user’s mobile device.

For more information, see Mobile web screen.
Chart screen

Chart screens display an interactive view of a report or performance analytics chart. Users can tap on a chart to display a list of relevant records.

- The chart screen can display time series and single score reports that are used in the web-based UI.
- The chart screen can display score type reporting charts that are configured to use the Latest score visualization.

For more information, see Chart screen.

Input form screen

Input form screens display inputs to allow your users to quickly enter information into mobile apps. Use input form screens to create or edit records, complete surveys, or any other situation where your users need to enter information.

Input form screens are screens that you use to execute an action or a function. They are not the first screen a user interacts with in a workflow.

For more information, see Input form screen.
Create a screen

Create a screen for a specific task, within your mobile app. Each screen comprises of one or more embedded screens displayed to your user to complete a task.

Before you begin
Role required: admin

About this task
Screens are a collection of components used to visualize your data templates. Depending on the screen type you select, different options are available.

⚠️ Warning: You can also configure individual screens in the Now Platform web-based UI. Unlike the Mobile App Builder, the Now Platform doesn’t prevent you from using configurations that might break the app.
Procedure

1. Navigate to **All > System Mobile > Mobile App Builder.**
   The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.

2. Search for the application scope you are working in and then select the name of the application scope.
   The Mobile App Builder categories home screen displays.

3. Select the **Screens** category, and then select **New.**

4. Select one of the displayed screen types, and then select **Continue.**

Create a screen

What to do next
What you configure for the screen depends on the screen type you select. For more information on how to configure a specific screen type, navigate to the topic that is specific to that screen.

Copy a mobile component

Use Studio to create a copy of a mobile component

Before you begin
Role required: admin

Procedure

1. In Studio, open the component you want to copy.
   For example, to copy an applet navigate to **Mobile Studio > Applets** and select **Applets.**

2. Open a supported copyable component.

© 2022 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.
You can make a copy of the following components:

- Applet launchers
- Data items
- Employee directory applets
- Form applets
- List applets
- Map applets

3. Click the **Make a Copy** button.

   **Note:** The **Make a Copy** button is disabled if the component has unsaved changes.

4. In the Make a Copy pop-up, enter a name for the new applet in the **Name** field.

5. Select an application in the **Application** field. The new component is copied to that application.

6. Click **Save**.

   The instance creates a copy of your component, and duplicate versions of all necessary components. New components are named with the name entered in the **Name** field, with an appended suffix for the specific component type.

   **Note:** Copied applets do not automatically create copies of the associated data item. The new applet will use the same data item as the original record.

---

**List screen**

Use a list to display a list of records. Records in lists display in a card format, showing a limited selection of the information in the record.
Use a list applet to see records in a list view. Within a list applet, each record has an item view. This item view is a template which contains a selected few fields that display as a preview for each record. You can configure the item view in studio when creating a list applet.

Use the optional features available with list screens to improve your users experience.

**Form screen**

Configure your list screen applet to include a form screen. This configuration automatically displays a form view of a record when your users tap it in the list screen. Learn more about form screens in Form screen.

**List filters**

Mobile list filters enable your users to find what they need in the records on your instance. For more detail on filters, see Mobile list screen filters.

**Dynamic screen name**

Configure your list screen applet to inherit the name of a previous screen selection or form field and display it as the header on a subsequent screen. For more information, see Configure a dynamic screen name for a list screen.

**Functions**

Functions determine which actions users can perform in your mobile apps. For more detail on functions, see Mobile functions.
Configure lists with forms

Configure your list with a form so your users can tap list items to view details of the records on your list. For details on list configuration, see List applet configuration.

Parametrized list applets

Use parameters to pass information into your list applets. For example, when a user navigates taps on a group in a group list applet, a list containing records in the group is displayed. The list of records in the group is a parametrized list, which accepts the value of the group your user selected as a parameter.

To learn about parametrized list configuration, see Parametrized list screens.

List screen functions

Add functions to your list applets to give your users the ability to take actions directly on a form screen. List applets support both swipe and top menu functions. For more information on functions, see Mobile functions.
**Swipe functions**

Use swipe functions to act on a specific record on the list. You can configure one or more actions to appear when a user swipes on a record. You can also configure a different set of actions depending on whether a user swipes to the left or to the right.
Top menu functions

Use top menu actions for less frequently used actions, or actions that apply to the list rather than individual records, such an Approve all action.

List filters tbd

Dynamic screen name

Use dynamic screen name functionality to let users to easily identify the screen or field which they are currently viewing. You can configure a screen to inherit a name from a previous screen selection. These dynamic names can be used with grouped lists, where the selected group name becomes the header on a subsequent screen. Alternatively, a field in a form can be used as a header in a different screen. For more information, see Configure a dynamic screen name for a list screen.

Display name from a redirection parameter

Use a dynamic screen name to inherit a name from a previous screen selection. In
the example, the screen name is coming from a list which is grouped by category from an applet launcher.

Display information from field previous record

Use dynamic form screen names to display information from a specific record in a table. The form screen name can come from any field in a record. In the example, the screen name is coming from the number field.
List applet configuration

Configure a list applet so that your users can see a list of records from their mobile app. To access a record from the list, a user just has to tap the record name.

Before you begin
Role required: admin

Procedure
1. Navigate to All > System Applications > Studio to open Studio.
2. Select a scoped application where you want to create your applet.
   If you have not created an application, you can create one by using the Create Application button. For information on creating a scoped application for mobile, see Create a mobile application using Guided Application Creator.
3. In Application Explorer, navigate to Mobile Studio > Applets and select Applets.
4. Click the pop-out icon ( ) that appears to the right of Applets.
5. In the applet list, click Create an applet.
   The Create an Applet window appears on the screen.

Create an Applet window

6. In the Create an Applet window. Enter a name and description for your list applet.
7. Click the image next to the Icon field to select a color and image for your list applet icon. If you add your list applet to an applet launcher, this icon displays for the applet.
8. To prevent the list from displaying a list filter, enable Hide filter.
9. In the Choose a template field, select List.
10. Optional: To the right of the Choose a template, enable the checkbox next to Form to include a form screen in your applet. This form displays record information when your users tap a record on the list.
11. Click Create New.
   The Create an Applet window closes, and a new Studio tab opens for your list applet. If you selected the form option in the previous steps, you can also see a tab to configure your form.
12. In the **List Screen** tab, in the **Data Item** field, select a data item from the list. If the Data Item list is disabled, no data items have been created. To create a data item, click the plus icon (++) next to the list to open a **New Data Item** tab. In the **New Data Item** tab, create your data item. For more information on creating a data item, see **Configure a standard data item**.

13. **Optional:** When created, your applet uses the default template visible in the **Template preview** panel for the cards shown in your applet. Click **Customize in Mobile Card Builder** button to use Mobile card builder to update the appearance of the card and which fields appear. For details on customizing an item view, see **Customize a screen using Mobile Card Builder**.

14. Click **Save**.
Results

List applet primary screen

What to do next

List can be configured to include a form screen when they are created. If your list includes a form screen, click the Form Screen tab to configure that screen. For details on how to configure form screens, see Configure a form applet for use with a list applet.

You can also configure your lists to navigate to screens other than forms. For an example of this, see Tutorial: Configure navigation from a list applet to another list applet.

List segment configuration

Create list segments so that your mobile app users can see multiple lists within the same list applet in a tabbed format.

Before you begin

Role required: admin
Procedure

1. Navigate to All > System Mobile > Applets.

2. Open the applet record where you want to create a new segment.
   When you create a list applet in Studio, the instance automatically creates a form applet with the same name. Add the Class field to the list so that you can see which one is the list applet.

3. In the Item Stream Segments list, click Insert a new row.

4. Enter a name for your new segment.

5. Select Ascending or Descending as the sort order in the Interleave Order Direction field.

6. In the Order field, enter a numerical value.
   Segments display from the lowest to the highest value.

7. Clear the Hide filters option to hide the filter options for your lists.
   This option hides list filters for all segments in your list applet.
8. Right-click the form header and select Save.
9. To open the item stream segment record, click the name of your new segment.
10. In the Item Stream M2M Segments list, click Insert a new row.
11. Select the item stream list that you want to add to your segment.
12. Click Update.

Configure a form applet for use with a list applet
Use a form applet as part of a list so that your users can see the details of a record from their mobile app.

Before you begin
Role required: admin

About this task
Your instance creates form applets automatically for calendar, list, and map applets. You can also create your own form applets manually.

Procedure
1. Navigate to All > System Applications > Studio to open Studio.
2. Select a scoped application that contains your list applet.
3. In Application Explorer, navigate to Mobile Studio > Applets and select Applets.
4. Select the applet where you want to configure a form.

   Note: List applets have the option to include a form screen when you create them. Select a list applet where you used this option.

Create an Applet window
Create an Applet

Define the properties of the applet

Choose a template

List Form Employee Directory Map
Group List Calendar URL Chart

5. In your applet configuration form, select the From Screen tab.
6. **Optional:** When created, your applet uses the default template visible in the **Template preview** panel for the cards shown in your applet. Click **Customize in Mobile Card Builder** button to use Mobile card builder to update the appearance of the card and which fields appear. For details on customizing an item view, see **Customize a screen using Mobile Card Builder**.

7. Click **Body**.

8. Move fields from the All fields list to the Selected fields list by double-clicking a field, or selecting a field and clicking the add or remove buttons.

9. **Optional:** Select the **Attachment list** option below the Selected fields list to show a list of attachments at the bottom of the form. For more detail on attachments in mobile, see **Manage attachments on details screens**.

10. Click **Save**.

**Parametrized list screens**

Learn how to use parameters to pass information into a list applet.

Use parameters to pass information into a list applet. For example, a user views a grouped list showing records by grouped by priority. When a user selects a priority, a list containing records matching that priority displays. This second list needs to know which priority the user selected to display the appropriate records. The system accomplishes this task by passing a parameter. The first list stores the selected priority value in a parameter. The second list uses the information in that parameter to display the appropriate records.
Configure a parametrized list applet

Configure a list screen to query a user for a parameter, which the applet uses to filter the records that appear on the screen.

**Before you begin**

Role required: admin

Creating a parametrized list is similar to creating a standard list. In addition to the steps used to create a standard list, you must create:

- A parametrized data item for your list. This data item uses your parameter to filter what data is displayed in your list.
- A parameter on your list applet. This parameter stores the value passed into your list.
- A screen UI parameter mapping to associate the parameter in your list with the parameter in your data item.
**Procedure**

1. Navigate to All > System Applications > Studio to open Studio.

2. Select a scoped application where you want to create your applet. If you have not created an application, you can create one by using the **Create Application** button. For information on creating a scoped application for mobile, see Create a mobile application using Guided Application Creator.

3. In Application Explorer, navigate to **Mobile Studio > Applets** and select **Applets**.

4. Click the pop-out icon ( ) that appears to the right of **Applets**.

5. In the applet list, click **Create an applet**. The **Create an Applet** window appears on the screen.

6. In the **Create an Applet** window. Enter a name and description for your list applet.

7. Click the image next to the **Icon** field to select a color and image for your list applet icon. If you add your list applet to an applet launcher, this icon displays for the applet.

8. To prevent the list from displaying a list filter, enable **Hide filter**.

9. In the **Choose a template** field, select **List**.

10. **Optional**: To the right of the **Choose a template**, enable the check box next to **Form** to include a form screen in your applet. This form displays record information when your users tap a record on the list.

11. Click **Create New**. The **Create an Applet** window closes, and a new Studio tab opens for your list applet. If you selected the form option in the previous steps, you can also see a tab to configure your form.
12. Click the plus icon (+) to the right of the Data Item field. A new Studio tab opens to configure a new data item.

13. On the new data item form, fill in the fields.

<table>
<thead>
<tr>
<th>New Data Item form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Table</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Condition Type</td>
</tr>
</tbody>
</table>

14. Click Save.

15. In the Parameter Definition section, click the plus icon (+) to create a new parameter.

16. In the Parameter Definition pop-up, enter a name for your parameter in the Name field.
17. Click Save.

18. In the Query condition field, create a query that uses your parameter to filter your records.

To access your parameter in the query condition, click the reference value icon ( ).

19. Click Save.

You have a parametrized data item to use in your parametrized list. This example shows a parametrized data item that uses a parameter named Priority to filter the data items for a priority matching that parameter's value.

Note: For more detail on available options when creating parametrized data items, see Configure a parametrized data item.

20. In Studio, navigate back to the tab containing your list applet.

21. In the Data Item field, select the parametrized data item you created in the previous steps.

22. Click Save.

23. In the User Input Parameter Definition section, click the plus icon ( ) to create a new parameter.

24. In the User Input form, fill in the fields.
25. Click **Save**.  
Your parameter is created, and the **User Input** form closes.

26. In the **Screen UI Parameter Mapping** section, click the plus icon (+) to create a new parameter mapping.  
The **Screen UI Parameter Mapping** form displays.

27. In the **Item Parameter** field, select the parameter you created in your data item.

28. In the **UI Parameter** field, select the parameter you created in your parametrized list.

29. Click **Save**.  
The parameter in your list is associated with the parameter in your data item. When a user accesses this list, the applet prompts the user for a value for the parameter. The data item uses that value to filter the records displayed in the list.

**What to do next**  
For an example demonstrating how to use parameters to create a grouped list, see **Tutorial: Configure a list of grouped incidents**.

**Mobile list screen filters**  
Use mobile filters in list screens to enable your users to find what they need in your instance’s records.
Choose one of the following options when configuring your mobile list screens:

**Default filters**

By default, the instance automatically creates a filter and sorting functionality. This filter is based on the fields selected in your list pattern. No further configuration is needed. Users can filter and sort the list based on any of these fields.

**Custom filters**

If the default filters do not fit your requirements, consider creating a customer filter experience for your users. You can manually select the fields available to the list filter, as well as enable keyword filtering. Custom filters can include fields that are not in the list pattern. For more details on creating custom filters, see Configure a custom list screen filter.

**Note:** For those who have upgraded to the Rome release, consider the following:

- If you did not define filters and are using the default setup, you automatically use the default filters and sorting capabilities after the upgrade.
- If you defined filters in earlier versions, then after the upgrade your defined filters are still available. However, you must manually configure the sorting capabilities. See, Configure sorting capabilities within mobile filters.

**Disabling filters**

To disable filters and sorting on list screens, open the list screen record, click the **Filter** tab, and disable the **Show or hide filters for this list screen** option. When filtering is turned off, users are unable to filter or sort records.

---

Configure a custom list screen filter

Manually create a filter experience for your users. Custom filters can include fields that are not in the list pattern.

**Before you begin**

Role required: admin
Procedure
1. Navigate to All > System Applications > Studio.
2. Select your application.
3. In the Application Explorer, navigate to Mobile Studio > Applets, and open the list applet where you want to configure a custom filter.
4. In the list applet form, select the Filters tab.
5. If inactive, enable the Show or hide filters for this list screen field.

Note: Enabling the filter functionality also enables the sorting functionality.
   When Set filters manually is enabled, the default list of filter fields is cleared.
7. Optional: To enable keyword filtering on your list, enable the Enable Keyword filter field. Studio creates a keyword filter and adds it to your filter list.

8. Select the plus icon (⁺) to add a field to your list filter.
9. In the Add Filter pop-up, fill in the fields as needed.

Add filter field
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Field</td>
<td>The field used in this filter item. The data item used by the list determines which fields are available.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Filter Label</td>
<td>The label used for this filter field. This field defaults to the name of the field selected in the Data Field field.</td>
</tr>
<tr>
<td>Operator Selection</td>
<td>The operator type used by the filter. Some fields support multiple operator types. For example, for date fields, select On to filter for a specific date, or Between to filter for records within a date range.</td>
</tr>
<tr>
<td>Filter Type</td>
<td>Filter type used by this filter item. This value is automatically determined by the value in the Operator Selection field.</td>
</tr>
</tbody>
</table>

**Note:** Boolean filter types require an alternative configuration, see Add Boolean fields within a mobile filter.

10. Select Save.
11. Add additional filters by repeating steps 8 through 10 as needed.
12. When you have completed adding fields to your filter, select Save in the list applet form.

**Add Boolean fields within a mobile filter**

Add Boolean fields, such as 'Active: Yes / No', within mobile filters, so users can more easily search for specific data and streamline their results.

**Before you begin**
Role required: admin

**Procedure**
1. Create a filter condition field.
   a. In the web-based UI, enter `sys_sg_filter_condition_field.list` in the filter navigator.
   b. Select New.
   c. On the Filter Condition Field form, fill in the fields.
      i. **Name**: Enter a title for the filter condition field.
      ii. **Table**: Select a table which contains the data to filter.
      iii. **Field**: Select a Boolean type field from the list.
   d. Select Submit.
2. Create a filter condition to group the filter condition fields. For example, choose an incident table and a Boolean-type field to filter based on whether incidents are active or inactive.
   
   **Note:** You need this element even if only one filter condition field exists.
a. In the filter navigator, enter sys_sg_filter_condition.list.

b. Select New.

c. On the Filter Condition Form, fill in the fields.
   i. Name: Enter a title for the filter condition field.
   ii. Type: Select the reference lookup icon (🔍) and select Boolean from the list.
   iii. Label: Enter a name for the Boolean filter displayed to the user.

d. Right-click in the header and select Save.

3. Create a many-to-many relationship to connect the filter condition and the filter condition field.
   a. Select the Filter Conditions Fields tab within the Filter Condition form.
   b. Select New.
   c. In the Filter Condition Field field within the Filter Condition M2M Filter Condition Field form, select the reference lookup icon and select the filter condition field you created.
   d. Select Submit.

4. Add a filter category to display at the top of your filter screen.
   a. In the filter navigator, enter sys_sg_filter_category.list.
   b. Select New.
   c. On the Filter Category Form, fill in the fields.
      i. Name: Enter a title for the filter category field.
      ii. Label: Enter a name for the filter category to display to the user.
      iii. Table: Select the same table you defined in the Filter Condition Field Form.
   d. Right-click in the header and select Save.

5. Create a many-to-many relationship to connect the category and the filter condition.
   a. Select the Filter Conditions tab within the Filter Category form.
   b. Select New.
   c. In the Filter Condition field within the Filter Category M2M Filter Condition form, select the reference lookup icon and select the filter category you created.
   d. Select Submit.

6. Define a filter that contains the filter category.
   a. In the filter navigator, enter sys_sg_filter.list.
   b. Select New in the Filters form.
   c. In the Name field of the Filter form, enter a name for the filter.
   d. Right-click in the header and select Save.

7. Create a many-to-many relationship to connect the category and the filter.
a. Select New in the Filter M2M Filter Categories area of the Filter form.

b. In the Filter Category field within the Filter M2M Filter Category form, select the reference lookup icon and select the filter category you created.

c. Select Submit.

8. Assign the created filter with a Boolean field, to either a list screen or a map screen.

a. In the filter navigator, enter either sys_sg_list_screen.list or sys_sg_map_screen.list.

b. In the Name field, search for the screen where you want to add the filter.

c. Select the required screen and in the form perform the following:
   i. In the Filter field, select the reference lookup icon and select the filter you created.
   ii. Make sure that the Hide filter and sorting field are cleared.

d. Select Update.

Configure sorting capabilities within mobile filters

Use the sorting option in lists and maps to help users organize their filtered results and enable them to view the most relevant information. Examples of configurable sorting options are: recently added, assigned to (ascending), assigned to (descending), and highest priority first.

Before you begin

A sorting option entitled Default always displays at the top of the sorting list. The values of the default sorting are defined using the Add sort and Sorted by fields within a data item. See, Configure a standard data item. If values are not defined within a data item, the default sorting definitions of the instance are taken from the web-based platform definition.

Role required: admin

About this task

By default, when creating item sorting, the instance creates an ascending and descending variant of the item sorting. Each of these entries contains the text "(Ascending)" and "(Descending)" next to the sorting entry. To change the default display option, see Configure sorting display options for mobile filters.

Note: Filters and sorting options use the same defined categories, as defined by item views and view configs.

Procedure

1. Create an item sorting to add fields to a list.

   a. In the web-based UI, enter sys_sg_item_sorting.list in the filter navigator.

   b. Select New.

   c. On the Item Sorting form, fill in the fields.
i. **Name**: Enter a title for the item sorting.

ii. **Type**: Select the reference lookup icon (🔍) and select a filter type from the list.

iii. **Label**: Enter a name for the sorting option displayed to the user in the sorting list.

d. Right-click in the header and select **Save**.

2. Create an item sorting field, to create a relationship between the item sorting field and the item sorting.

   a. Click the **Item Sorting Fields** tab in the **Item Sorting** form.

   b. Select **New**.

c. On the Item Sorting Field form, fill in the fields.
   
   i. **Name**: Enter a title for the item sorting field.

   ii. **Table**: Select a table which contains data to sort.

   iii. **Field**: Select the element from the tree which relates to the selected table.

   iv. **Item Sorting**: This field defaults to the item sorting name you created earlier.

d. Click **Submit**.

3. Repeat step 2 to display additional sorting options.

   ✪ **Note**: To change the sorting display attributes, see **Configure sorting display options for mobile filters**.

4. Add a filter category that contains your item sorting.

   a. In the filter navigator, enter `sys_sg_filter_category.list`.

   b. Select **New**.

c. On the Filter Category form, fill in the fields.
   
   i. **Name**: Enter a title for the filter category field.

   ii. **Label**: Enter a name for the filter category displayed to the user.

   iii. **Table**: Select the same table you defined in the Item Sorting Fields form.

d. Right-click in the header and select **Save**.

5. Create a many-to-many relationship to connect the category and the item sorting.

   a. Select the **Item Sorting** tab within the **Filter Category** form.

   b. Select **New**.

c. In the **Item Sorting** field within the Filter Category M2M Item Sorting form, select the reference lookup icon and select the item sorting you created.

d. Select **Submit**.

6. Define a filter that contains the item sorting.

   a. In the filter navigator, enter `sys_sg_filter.list`.

   b. Select **New** in the Filters form.
c. In the **Name** field in the Filter form, enter a name for the filter.

**d.** Right-click in the header and select **Save.**

**7.** Create a many-to-many relationship to connect the filter and the filter category.
   
   a. Select **New** in the **Filter M2M Filter Categories** area of the Filter form.
   
   b. In the **Filter Category** field within the Filter M2M Filter Category form, select the reference lookup icon and select the filter category you created.
   
   c. Select **Submit.**

**8.** Assign the created filter with the item sorting, to either a list screen or a map screen.

   a. In the filter navigator, enter:
      
      - For a list screen - `sys_sg_list_screen.list`
      - For a map screen - `sys_sg_map_screen.list`

   b. In the **Name** field, search for the screen where you want to add the filter.

   c. Select the required screen and in the form, perform the following:
      
      i. In the **Filter** field, select the reference lookup icon and select the filter you created.
      
      ii. Make sure that the **Hide filter and sorting** field is cleared.

   d. Select **Update.**

**Results**

Your filter sorting configuration may look like the one in the image. Each item sorting has an ascending and descending sorting option associated with it. To change the default display options, see **Configure sorting display options for mobile filters.**
Configure sorting display options for mobile filters

Customize the way sorting options are displayed in your mobile filter and override the default behavior.

**Before you begin**
Role required: admin

**About this task**

**Procedure**

1. In the web-based UI, enter `sys_sg_item_sorting_attribute.list` in the filter navigator.
2. Select **New**.
3. From the **Item Sorting** field, select the reference lookup icon (🔍) and select the item sorting entry to configure.
4. Define attributes to configure for the selected item sorting.
<table>
<thead>
<tr>
<th>Attribute option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define the ascending label</td>
<td>a. From the Name field, select the reference lookup and select Ascending Label.</td>
</tr>
<tr>
<td></td>
<td>b. Enter a custom label in the Translatable Value field. The entered value takes priority over the value in the item sorting Label field in the Item sorting form.</td>
</tr>
<tr>
<td></td>
<td>c. Right-click in the header and select Save.</td>
</tr>
<tr>
<td>Define the descending label</td>
<td>a. From the Name field, select the reference lookup and select Descending Label.</td>
</tr>
<tr>
<td></td>
<td>b. Enter a custom label in the Translatable Value field. The entered value takes priority over the value in the item sorting Label field in the Item sorting form.</td>
</tr>
<tr>
<td></td>
<td>c. Right-click in the header and select Save.</td>
</tr>
<tr>
<td>Hide the ascending/descending suffix</td>
<td>Note: Use this option when the ascending or descending suffix is not needed, for example, “Recently added”.</td>
</tr>
<tr>
<td></td>
<td>a. From the Name field, select the reference lookup and select Hide Direction Label.</td>
</tr>
<tr>
<td></td>
<td>b. Enter the value true in the Value field. The instance does not display the (ascending) and (descending) suffix name next to the item sorting name.</td>
</tr>
<tr>
<td></td>
<td>c. Right-click in the header and select Save.</td>
</tr>
<tr>
<td>Display either ascending/descending option</td>
<td>a. From the Name field, select the reference lookup and select Order Options.</td>
</tr>
<tr>
<td></td>
<td>b. Perform one of the following:</td>
</tr>
<tr>
<td></td>
<td>• Enter the value ascending in the Value field to display only the ascending sorting option.</td>
</tr>
<tr>
<td></td>
<td>• Enter the value descending in the Value field to display only the descending sorting option.</td>
</tr>
<tr>
<td></td>
<td>c. Right-click in the header and select Save.</td>
</tr>
</tbody>
</table>

5. Click Submit.

Results
After configuring various sorting display options, your filter sorting configuration may look like the one in the image.
Configure a dynamic screen name for a list screen

Configure a screen to dynamically inherit a name from a parametrized list applet. This setup enables users to view a passed parameter as the screen name instead of the applet record name.

**Before you begin**
Role required: admin

**Procedure**
1. In the web-based UI, enter `sys_sg_screen.list` in the filter navigator.
2. On the **Applets** form, select a list screen to inherit the dynamic screen name.
3. Apply an existing UI parameter in the UI parameter related list.
a. If the UI parameters, Screen UI element mappings, and Source and UI element related lists are not displayed, add them by clicking the Additional actions icon ( ). Then, select Configure > Related Lists, and then select the required related lists.

b. Click the UI parameters tab.

c. Select an existing UI parameter.

d. On the form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter type</td>
<td>Screen</td>
</tr>
<tr>
<td>Input source</td>
<td>User input</td>
</tr>
<tr>
<td>Input type</td>
<td>Text</td>
</tr>
</tbody>
</table>

e. Click Submit.

4. Define a UI element to serve as the location point of the dynamic screen name.

Note: This step is a one-time configuration. Once you create the UI element, a new record for the screen title location is not required.

a. In the web-based UI, enter sys_sg_ui_element.list in the filter navigator.

b. Click New to create a new UI element.

c. On the form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Location where the dynamic screen name displays. Select Title.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the UI element.</td>
</tr>
</tbody>
</table>

d. Click Submit.

5. Create a screen UI element and map it to the screen type.

a. Click the Screen UI element mappings tab.

b. Click New.

c. On the form, fill in the fields.
**Screen UI element mapping form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screen</td>
<td>Auto-populated with the selected screen, which inherits the dynamic screen parameter.</td>
</tr>
<tr>
<td>UI Element</td>
<td>The UI element to be configured.</td>
</tr>
</tbody>
</table>

d. Click **Submit**.

6. Map the screen UI element with the UI parameter.

   a. Click the **Source and UI element** tab.

   b. Click **New**.

   c. On the form, fill in the fields.

**Source UI element mapping form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Element</td>
<td>The UI element to display the dynamic screen name.</td>
</tr>
<tr>
<td>Source Table</td>
<td>The UI Parameter table.</td>
</tr>
</tbody>
</table>

d. Click **Submit**.

**Results**
The screen name dynamically inherits a value from the redirection parameter.
Configure an empty state for a list screen

Configure an empty state to display on empty list screens, to provide information to further direct users.

Before you begin
You should already have an empty state configured for list screens that do not contain any data. For more information, see Configure an empty state.

Role required: admin

Procedure
1. In the web-based UI, enter `sys_sg_list_screen.list` in the filter navigator.
2. Select a list screen to display the specific empty state.
3. In the Empty State field, select an empty state to associate with the list screen.
4. Right-click in the header and select Save.

Form screen
Form screens display content for a specific single record.

Use form applets to display the content of a single record. You can configure functions on form applets to allow users to make edits and take actions. Form screen supports top menu functions, footer functions, and smart buttons.
When a form screen is created, a details screen is configured by default. Add any of the four additional segments types listed below as needed.

Within a form screen, you can control the embedded segments. There are four types of embedded screen segments available for use when configuring a form screen through Studio.

Form configuration screen

Use the form configuration screen in Studio to control the content and appearance of your form applets.

The form applet configuration screen in Studio

At the top of the form configuration page are the **Overall Settings** and **Body** tabs.

**Overall Settings**

Use the **Overall Settings** tab to define the following elements of your form applet.
• Use the **Data Item** field to define which data item you applet uses. You may select an existing data item or create a new one using the plus icon (+).

**Note:** When navigating to a recently created record, the context of the new record is passed in a form screen. If you do not want this record context passed, select the **Override context in data item** field. To access this field, go to the web-based UI, in the filter navigator enter `sys_sg_list_screen.list` and select the **Override context in data item** checkbox in your specified form screen.

• Use the **Customize in Mobile Card Builder** button to change the appearance of your mobile card or the fields displayed in the card. For more detail on using Mobile Card Builder, see [Customize a screen using Mobile Card Builder](#).

• Use the **UI Style Configurations** section to define the font and background color of fields in your applet.

• The top menu functions available on your form (shown under the **Functions** section.)

**Body**

Use the **Body** tab to define the elements in the form segments on your form applet such as:

• Use the **Modify Segment** button to define which screen segments appear on your form. See the section below for details on each screen segment.

• The **Field Functions** available on your form.

• Define which footer
Form screen segments

Activity stream screen segment

Use an activity stream screen to display work notes, comments, and/or attachments for the record. Your users can configure whether all three of these are visible, or only a select few are visible.

To configure an activity stream screen segment, see Configure an activity stream screen for a form applet.
Details screen segment

Use a details screen to display specific screen fields from the record. On a form, you can see the same item view/pattern that was configured for the list item. Embed a details screen in your form when you want to display more info about the record.

For more information on detail screens, see Configure a details screen for a form applet.

Embedded browser screen segment

Use an embedded browser screen to display content from a URL within a browser section. An embedded browser can display information such as related knowledge articles within your forms.

For more information on embedded browser screens see Configure an embedded browser screen for a form applet.

For a tutorial on embedding using browser screens to display knowledge articles, see Tutorial: Configure an applet to view knowledge articles within a ServiceNow mobile app.
Embedded list screen segment

Use an embedded list screen to display a related list as an embedded tab within a form screen. An embedded list can provide a user immediate access to a single related list without the extra tap when using the related list screen.

To configure an embedded list screen segment, see Embedded lists for a form screen.

Related list screen segment

Use a related list screen to display related lists for the record. Each related list you configure appears in this related list screen. Your users can see the name of the list and tap to see the records within that list.

To configure a related list screen segment, see Configure a related list screen for a form applet.
Configure a parametrized form applet
Configure a form screen to query a user for a parameter, which the applet uses to determine the record that appears on the screen.

Before you begin
Role required: admin

Procedure
1. Navigate to All > System Applications > Studio to open Studio.
2. Select a scoped application where you want to create your applet.
   If you have not created an application, you can create one by using the Create Application button. For information on creating a scoped application for mobile, see Create a mobile application using Guided Application Creator.
3. In Application Explorer, navigate to Mobile Studio > Applets and select Applets.
4. Click the pop-out icon ( ) that appears to the right of Applets.
5. In the applet list, click Create an applet.
   The Create an Applet window appears on the screen.

Create an Applet window

Define the properties of the applet

- Name
- Description
- Icon
- Hide filter

Choose a template

- List
- Form
- Employee Directory
- Map
- Group List
- Calendar
- URL
- Chart

6. In the Create an Applet window. Enter a name and description for your form applet.
7. Click the image next to the Icon field to select a color and image for your form applet icon. If you add your form applet to an applet launcher, this icon displays for the applet.
8. From the Choose a template section of the form, select the Form template.
   (Optional) You can also modify an existing form that is part of a calendar, list, or map applet. In Studio open a calendar, list, or map applet for which you want to configure a form. Then, click the Form Screen tab to view your form configuration.
9. Click **Create New**.
   The **Create an Applet** window closes, and a new Studio tab opens for your form applet.

10. To see the field configuration for your form screen, click **Data and Fields**.

11. Click the plus icon (\(+\)) to the right of the **Data Item** field.
    A new Studio tab opens to configure a new data item.

12. On the new data item form, fill in the fields.

    **New Data Item form**
    
    | Field          | Value                                                                 |
    |----------------|-----------------------------------------------------------------------|
    | Name           | Name of the data item.                                                |
    | Table          | Table where the data item gets its data.                             |
    | Description    | Description used to identify your data item.                         |
    | Condition Type | Whether the conditions for the data item are declarative, scripted, or use an encoded query. Leave this field at its default setting. |

13. Click **Save**.

14. In the **Parameter Definition** section, click the plus icon (\(+\)) to create a new parameter.

15. In the **Parameter Definition** pop-up, enter a name for your parameter in the **Name** field.

16. Click **Save**.

17. In the **Query condition field**, create a query that uses your parameter to filter your records.

    To access your parameter in the query condition, click the reference value icon (\(\))

18. Click **Save**.

    You have a parametrized data item to use in your parametrized form. This example shows a parametrized data item that uses a parameter named *Number* to filter the data item a record with a specific number.

    **Note:** For more detail on available options when creating parametrized data items, see **Configure a parametrized data item**.

19. In Studio, navigate back to the tab containing your form applet.
20. In the **Data Item** field, select the parametrized data item you created in the previous steps.

21. Click **Save**.

22. In the **User Input Parameter Definition** section, click the plus icon (⊕) to create a new parameter.

23. In the **User Input** form, fill in the fields.

### User Input form

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for your UI parameter.</td>
</tr>
</tbody>
</table>
| Input type           | How your users enter a value for this parameter. Select from the following options:  
|                      | • Text                                                               |
|                      | • Choice List                                                        |
|                      | • Search List                                                        |
|                      | • QR/Barcode                                                         |
| Table name           | Table used for the choice list where users select a parameter value. This field is visible only when the **Input type** field is set to **Choice List** or **Search list**. |
| Field name           | The field used for the choice list where users select a parameter value. This field is visible only when the **Input type** field is set to **Choice List** or **Search list**. |
| Default value        | Default value for your parameter. This field is visible only when the **Default value type** field is set to **Manual**. |
| Input style          | Input style for your parameter. Select from **Inline** or **Popup**. |
| Default value type   | Whether the parameter has a default value. Select **None** to have no default value, or **Manual** to enter a manual value in the **Default value** field. |
| Mandatory            | Whether user input for the parameter is mandatory.                   |
| Placeholder text     | Text that appears in the parameter entry field before the users enters a value. |
| Multi-select         | Whether the user can select multiple values from the choice list. This field is visible only when the **Input type** field is set to **Choice List**. |
| Search type          | The type of search used when finding a parameter value. This field is visible only when the **Input type** field is set to **Search List**. |
24. Click **Save**.
Your parameter is created, and the **User Input** form closes.

25. In the **Screen UI Parameter Mapping** section, click the plus icon (➕) to create a new parameter mapping.
The **Screen UI Parameter Mapping** form displays.

26. In the **Item Parameter** field, select the parameter you created in your data item.

27. In the **UI Parameter** field, select the parameter you created in your parametrized form.

28. Click **Save**.
The parameter in your form is associated with the parameter in your data item. When a user accesses this form, the applet prompts the user for a value for the parameter. The data item uses that value to filter the record displayed in the form.

29. Optional: To select the header pattern that you want to use for the list view and the details screen view, click **Change Pattern**.

⚠️ **Note:** If you change the pattern or the selected table, any existing header configuration is deleted.

If the pattern that you choose includes an image field, which is indicated by a circle, select an image type field. For example, select the **Caller > Avatar** field. Make sure that you select the image field, or any other dot-walked fields, before you select the parent field.

#### Header options

![Header options diagram]

⚠️ **Warning:** Although pattern modification is not supported, you can change the JSON that defines the pattern layout. If you modify a pattern record, Studio may not recognize the pattern. In this case, Studio displays a warning and a link to the item view [sys_sg_item_view] record, which contains the JSON for the selected pattern. A pattern that is not recognizable may still function as expected in the mobile app, but it is not editable in Studio.
30. Click Body.

31. Move fields from the **All fields list** to the **Selected fields** list by double-clicking a field. You can also select a field and click the add or remove buttons.

32. **Optional:** Select the **Attachment list** option below the Selected fields list to show a list of attachments at the bottom of the form. For more detail on attachments in mobile, see Manage attachments on details screens.

33. Click **Save**.

You have a completed parametrized form. When a user accesses the form screen, the form prompts the user for a value, and stores that value in the parameter. The value passes from the form parameter to the data item parameter, which defines which record appears on the form.

**Configure an activity stream screen for a form applet**

Configure an activity stream on your form to give your users access to comments, work notes, and attachments relating to the record they are viewing.

**Before you begin**

Role required: admin

**Procedure**

1. In Studio, navigate to **Mobile Studio > Applets**, and select the form applet you want to configure with an activity stream.
   Calendar, map, and list applets also include a form screen. You can add a related list to these screens by open of these applets and selecting the **Form Screen** tab.

   **My List Test**
   
   Applet | List
   
   [List Screen] [Form Screen]

2. Click **Body** to open the body configuration for your applet.

3. Click the **Modify Segments** button to open the **Modify Segments** window.

4. Click the **Activity Stream** slider to enable the activity stream, then click **Save**. Clicking save closes the **Modify Segments** window.

5. In the **Body** tab, click the **Activity Stream** button.

6. Click the sliders next to **Add Comments**, **Add Work Notes**, or **Add Attachments** to add those elements to your form.

7. In the upper right corner of your screen, click **Save**.

**Results**

Your form applet includes an **Activity** tab. Your users can tap this tab to view comments, work notes, and attachments relating to the record.
Configure an embedded browser screen for a form applet

Add an embedded browser screen to display content from a URL within a browser section.

Before you begin
Role required: admin

Before embedding a browser screen into your form, you must have a configured browser screen. For details on creating these screens, see Configure a mobile web screen.

Note: This configuration is performed outside of the Studio interface, in your instances web-based UI.
Procedure
1. Navigate to All > System Mobile > Applets.
2. Open the record for the form applet where you want to add an embedded browser.
   
   ✨ Note: Find your form applets by filtering the Class field for Form screens.
3. In the Form Screen Segments list, click New.
4. In the Select form segment type pop-up, select Form Screen Segment, and click OK.
   A new form screen segment record opens.
5. In the Embedded Screen field, select the browser screen you want to embed into your form.
   
   ✨ Note: Find your form applets by filtering the Class field for Browser Screen.
6. Click Submit.
7. In the form screen, click Update.

Results
Your form is configured with an embedded browser. Your users can access this screen on the form by tapping the tab with the name of this screen.

Configure a details screen for a form applet
Create a details screen to display record fields on your form.

Before you begin
Role required: admin

Procedure
1. In Studio, navigate to Mobile Studio > Applets, and select the form applet you want to configure with an activity stream.
   Calendar, map, and list applets also include a form screen. You can add a related list to these screens by opening these applets and selecting the Form Screen tab.

   ![My List Test Applet | List](image)
   
   List Screen  Form Screen

2. Click Body.
3. If you have more than one screen segment configured for your form app, click the **Details** button to configure fields for the details screen.

4. Move fields from the **All fields** list to the **Selected fields** list by double-clicking a field, or selecting a field and clicking the add or remove buttons.

5. **Optional:** Select the **Attachment list** option below the **Selected fields** list to show a list of attachments at the bottom of the form. For more detail on attachments in mobile see [Manage attachments on details screens](#).

6. Click **Save**.

**Manage attachments on details screens**

Add, remove, rename, and view attachments in your mobile form applets.
Enable attachments on your mobile forms

You can enable attachments on a mobile form. In Studio, open a form applet, click the Body tab, and then click the Details button. Select the Show attachment list check box on the Form Screen tab of the Incidents page.

For details on the form configuration, see Configure a form applet for use with a list applet.

View the attachment list on your form screens

When enabled by an administrator, the attachment list appears at the bottom of the Details tab of your mobile form screens. The attachment list shows a preview of each attachment that is associated with the current record. If there are no attachments, you see "No attachment" instead of the attachment list.
Tap an attachment to show a preview of the attachment.
Manage attachments

To delete an attachment, tap the **Delete** (🗑️) icon, and then click Yes in the confirmation pop-up window.

To rename an attachment, tap the **Rename** (✍️) icon. Then, in the **Edit name** field, enter a name.

**Note:** The ability to perform these actions are controlled by the Access Control List rules on your instance.

**Manage functions on details screens**

Add functions to your details screens to enable your users to take actions or view additional information.
Add functions to your details screens

You can add field and footer functions to the details screen in your form applets in Studio. To add functions, open a form applet, click the Body tab, and then click the Details button.

Add a field function

To add a field function, click the plus icon (+) in the Field Functions section. In the pop-up window, use the Field field to select a field from your form to associate with a function. Then, select a function in the Function field.

Add a footer function

To add a footer function, click the plus icon (+) in the Footer Functions section. In the pop-up window, use the Label field to enter a name for your footer button. Then, select a function in the Function field. Use the Button Emphasis field to select an appearance for your footer function button.

For information on creating functions to add to your applets, see Mobile functions. For more information on placing functions in your applets, see Associate a function with a location in the app.
Screen fields
Screen fields improve the usability of your form applets. With screen fields, your users can view and edit attachments, view and complete checklist items, or display a field without a label.

When configuring a details screen, you select fields to display on your form. When you save the details screen, the instance creates a Screen Field [sys_sg_screen_field] each selected field. You can modify these records to change the screen field type. Change the screen field type to change how your information is presented, and improve the appearance and usability of the applet for your users.

Screen field types

**Text**
Use a text screen field to display text on your form. This text can come from any field on the current record. The text field type has a *Value only* option, which displays the field without a label. In this example, the *Description* field is shown without a label.

```
Incident

● In Progress

Water damage to server
just now

Details
Activity

Seeing puddles of water on the floor of the server room. We've put buckets out to collect the water until this is fixed.

Opened by
System Administrator

Opened
2019-01-30

Caller
Bow Ruggeri
```

**Percentage**
Use a percentage screen field to display a percentage value on your form. This value comes from a field on the current record. Percentages are determined by using a decimal value. For example, a value of 0.35 displays in a percentage field as 35%. This field type has a *Value only* option, which displays the field without a label.

```
Incident

● In Progress

Water damage to server
just now

Details
Activity

Seeing puddles of water on the floor of the server room. We've put buckets out to collect the water until this is fixed.

Opened by
System Administrator

Opened
2019-01-30

Caller
Bow Ruggeri

Project Completion
35%
```
**Image**

Use an image screen field to display an image on your form. The value for this field comes from an image field or field that contains the sys_ID of an image in the Attachment [sys_attachment] table. Users can tap an image to open a preview screen to display the full image.

**Attachment**

Use an attachment screen field to display all the attachments of a record. Buttons to rename or delete attachments display automatically for your users. Users can tap a button to modify attachments.

**Video**

Use a video screen field so that your users can watch a video in your form. The video screen field gets its value from a field on your current record and contains a link to a video. The video must be externally hosted. Users can see the video in a new window when they tap Play Video.
File

Use a file screen field to display a PDF file within your form. The PDF displays as a preview on your form. Users can tap the preview icon to open it in a preview screen where they can scroll and zoom on the PDF document. The file screen field gets its value from a field on the current record. The value must be the sys_ID of an attachment [sys_attachment] record on your instance that contains a PDF file.

HTML

Use an HTML screen field to display HTML content within your form. The HTML screen field can get its value from an HTML field in your current record or a string field that contains an HTML code.
**Checklist**

Use checklist screen fields to display all the checklist items that are associated with a record. Users can tap individual checklist items to mark them as complete or incomplete.

**Date**

Use a date screen field to display a date value on your form. This value comes from a date or date/time field on the current record. Dates are displayed in the format that are defined by your instance. This field type has a **Value only** option, which displays the value of the field without the field label. If the field is configured with an action function, users can change the values of these fields. For more information, see Configure an action function.
Stage
Use stage fields to display a read-only approval or completion status of requested items and services. Stage fields are displayed under the following conditions:

• A stage field is added on a form for Requested items [sc_req_item] records.

• A stage field is added on a form for Catalog item [sc_cat_item] records, as long the stage names and statuses are configured using one of the following flow types:
  ◦ Flow Designer
  ◦ Workflow [wf_workflow]
  ◦ Execution Plan [sc_cat_item_delivery_plan]

• A stage field is added on a form for any other tables if the stage names and statuses are configured using Workflow [wf_workflow].

Add screen fields to your form screen
Add screen fields to improve the usability of your form applets. When you add a screen field, you can change how information is shown in your form or you can provide access to additional elements, such as attachments, videos, or links.

Before you begin
Role required: admin

Procedure
1. Navigate to All > System Mobile > Applets.
2. Open the form applet that you want to add an enhancement to.
   You can see form applets by filtering where Class is Form Screen.
3. In the Form Screen Segments list, click Details.
4. In the Screen fields tab, click New.
5. On the form, fill in the fields.

Form screen form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Type of screen field. For a list of types and a description of how they are used, see Screen fields.</td>
</tr>
<tr>
<td>Application</td>
<td>Application where this field is used. This field is automatically filled.</td>
</tr>
</tbody>
</table>

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>Field Name</td>
<td>Text that appears as a field label. This option is only available for Attachments list fields. All other field types use the same label as the selected Form Field.</td>
</tr>
<tr>
<td>Form Field</td>
<td>Table field that this screen field uses as a data source. Some field types require a specific type of value. These requirements are described in Screen fields.</td>
</tr>
<tr>
<td>Hidden</td>
<td>Option that you can select to prevent this field from showing on the form applet.</td>
</tr>
<tr>
<td>Order</td>
<td>Order in which this field appears. Fields appear on the form applet from the lowest to the highest value.</td>
</tr>
<tr>
<td>Value Only</td>
<td>Value of the field without the field label.</td>
</tr>
<tr>
<td>Note:</td>
<td>Not all field types have this option.</td>
</tr>
<tr>
<td>Screen</td>
<td>Screen where this field is used. This field is automatically filled.</td>
</tr>
</tbody>
</table>

6. Click Submit.

Add a video screen field to a form screen
Learn how to use a video screen field to view embedded videos in your mobile applets.

Before you begin
Role required: admin

Procedure
1. Navigate to All > System Mobile > Applets.
2. Open the form applet that you want to add an enhancement to.
   You can see form applets by filtering where Class is Form Screen.
3. In the Form Screen Segments list, click Details.
4. In the Screen fields tab, click New.
5. In the Type field, select Video.
6. In the Form field, select a field from your table. The value of the field you select is used by the instance as a URL to an embedded video. For example, you can use the text https://www.youtube.com/watch?v=7YNGMhp7yGE&t=95s to embed a video from YouTube.
   Only externally hosted videos are supported. There are no supported methods for displaying videos on your instance.
7. Click Submit.

Configure a dynamic screen name for a form screen
Configure a form screen to dynamically inherit a name from a field in a previous record. This setup enables users to view a single specified field as the screen name instead of the applet record name.
Before you begin
Role required: admin

Procedure
1. In the web-based UI, enter `sys_sg_screen.list` in the filter navigator.
2. Select a form screen to inherit the dynamic screen name.
3. Create a UI parameter in the UI parameter related list.

   a. If the UI parameters, Screen UI element mappings, and Source and UI element related lists are not displayed, add them by clicking the Additional actions icon (≡), selecting Configure > Related Lists, and then selecting the required related lists.

   b. Click the UI parameters tab.

   c. Either select an existing UI parameter or click New to configure a new UI parameter with specific values.

   d. On the form, fill in the fields.

   **UI parameter element mapping form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value / Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter type</td>
<td>Screen</td>
</tr>
<tr>
<td>Input source</td>
<td>Auto fill</td>
</tr>
<tr>
<td>Input type</td>
<td>Source field</td>
</tr>
<tr>
<td>Button parent table</td>
<td>The same table as listed in the form screen.</td>
</tr>
<tr>
<td>Source field</td>
<td>The field to display for the screen name.</td>
</tr>
</tbody>
</table>

   e. Click Submit.

4. Define a UI element to serve as the location point of the dynamic screen name.

   ✋ Note: This step is a one-time configuration. Once you create the UI element a new record for the screen title location is not required.

   a. In the web-based UI, enter `sys_sg_ui_element.list` in the filter navigator.

   b. Click New.

   c. On the form, fill in the fields.

   **UI elements form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Location where the dynamic screen name displays. Select Title.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the UI element.</td>
</tr>
</tbody>
</table>

d. Click Submit.

5. Create a screen UI element and map it to the screen type.

a. Click the **Screen UI element mappings** tab.

b. Click **New**.

c. On the form, fill in the fields.

Screen UI element mapping form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screen</td>
<td>Auto-populated with the selected screen, which inherits the dynamic screen</td>
</tr>
<tr>
<td></td>
<td>parameter.</td>
</tr>
<tr>
<td>UI Element</td>
<td>The UI element to be configured.</td>
</tr>
</tbody>
</table>

d. Click Submit.

6. Map the screen UI element with the UI parameter.

a. Click the **Source and UI element** tab.

b. Click **New**.

c. On the form, fill in the fields.

Source UI element mapping form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Element</td>
<td>The UI element to display the dynamic screen name.</td>
</tr>
<tr>
<td>Source Table</td>
<td>The UI Parameter table.</td>
</tr>
</tbody>
</table>

d. Click Submit.

**Results**

The screen name dynamically inherits the value from a defined field in an existing record. In the graphic, the screen name comes from the number field.
Configure an item UI section for a form screen

Enrich your form screen with the addition of item UI sections displayed in a card format. This capability presents content and users can trigger actions without the need to drill to an additional list screen.

Before you begin
At least one pre-configured item UI section must be created. For more information, see Configure record sections.

Role required: admin

Procedure
1. Create a section screen.
   a. In the web-based UI, enter `sys_sg_sections_screen.list` in the filter navigator.
   b. Select New.
   c. On the Item Sorting form, fill in the fields.
      i. **Name**: Enter a title for the section screen.
      ii. **Icon**: Select the reference lookup icon (🔍) and select an icon from the list.
Note: The icon selected is not be displayed in this configuration.

iii. Fetch Type: Select either Prefetch, On-demand, or Background as required according to your setup. For more information about these options, see Mobile fetch types.

d. Right-click in the header and select Save.

2. Add item UI sections to the selected section screen.

   a. Select the Sections screen sections tab and click New.

   b. On the Sections screen sections form, fill in the fields.

      i. Order: Enter a number to define where to display this item UI section.

      ii. UI section: Select the reference lookup icon and select a pre-configured item UI section.

      iii. Section screen: This field defaults to the section screen name you created earlier.

   c. Select Submit.

3. Optional: Map configured header and footer functions to the selected screen.

   a. Select the Functions instances belonging to the screen tab and click New.

   b. On the Function instance form, fill in the fields.

      i. Name: Enter a title for the function.

      ii. Function: Select the reference lookup icon and from the menu and select a function to add to the screen.

      iii. Label: Enter a title for the function.

      iv. Location: Select either Header or Footer.

      v. Button emphasis: Select an appearance for your footer function button, either Primary, Secondary or Destructive.

   c. Select Submit.

4. Map the item UI sections to a form screen and display it as a tab element.

   a. Enter sys_sg_form_screen.list in the filter navigator.

   b. Select the form screen to contain the item UI section.

   c. Select the Record screen segments tab.

   d. Select New.

   e. From the Select form segment type menu, select Record screen segment and then select OK.

   f. On the Record screen segment form, fill in the fields.
Configure a mobile alert for a form screen

Configure a mobile alert overlay for a form screen to inform users of an important message and to redirect them to a specific screen. Only one mobile alert is available per instance.

**Before you begin**

When configuring an alert for a form screen, you must have a form screen to display the alert. See, *Form screen*.

Role required: admin

**Procedure**

1. Configure a navigation function button for use within a mobile alert.
   a. In the web-based UI, enter `sys_sg_button.list` in the filter navigator.
   b. Select the button to use with the mobile alert.
   c. On the Function, make sure to fill the following fields:
      i. **Name**: Title for the button
      ii. **Type**: Select the Navigation option.
      iii. **Context**: Select the Record option.
      iv. **Destination type**: Select the Screen option.
   d. Select the **Condition** tab and define the conditions that apply for the display and the hiding of the mobile alert. For more information, see *Configure a navigation to an applet*.
   e. Select the **Action Completion** tab.
   f. In the **Destination screen** field, select the reference lookup icon (🔍) and select a screen where the instance redirects the user to.
   g. Select **Submit**.

2. Configure the mobile alert content displayed to the user.
   a. Enter `sys_sg_alert.list` in the filter navigator.
   b. On the Mobile Alerts form, click **New**.
   c. On the form, fill in the fields.
Mobile Alert form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the alert.</td>
</tr>
<tr>
<td>Title</td>
<td>Title of the alert message displayed to the user.</td>
</tr>
<tr>
<td>Message</td>
<td>Body of the message displayed to the user.</td>
</tr>
</tbody>
</table>

d. Select the **Dismiss Button** tab and enter a text that displays to the user in the **Dismiss button label** field. Examples are “Close” and “Dismiss”.

These texts are translated field types associated with the Translated Name / Fields [sys_translated] table, see Translated Name / Field table.

e. Select the **Primary Button** tab.

i. Enter a text in the **Primary button label** field that redirects the user to a screen. Examples are “View” and “Display”.

These texts are translated field types associated with the Translated Name / Fields [sys_translated] table, see Translated Name / Field table.

ii. In the Primary button field, select the reference lookup icon and select the navigation function button you defined earlier.

f. Select **Submit**.

3. Enable the mobile alert in a form screen by adding a reference to the mobile alert.

a. Enter `sys_sg_form_screen.list` in the filter navigator.

b. Select a record screen that displays the mobile alert.

c. In the **Alert** field of the Record screen form, select the reference lookup icon and select the form screen where the alert displays.

d. Select **Submit**.

**Results**
The mobile alert overlay displays over a list screen. The alert contains a message, the navigation function button, and the option to close the alert. There is also an alert icon ( '![](https://example.com/alert_icon.png)') in the screen header. The user can tap on the icon at any time, to review the mobile alert. The red spot indicates to the user that they have not performed the call to action.
Using dynamic segments to display data in a form screen

Dynamic segments in a form screen enable users to see a subset of relevant records rather than a whole list of records.

Use dynamic segments to create a link between selected elements near the top of the screen and the content displayed below. The data displayed can either be the whole list of records or specific records that relate to the individual segments. For example, a timesheet can contain a dynamic segment consisting of the days of the working week. When a day is tapped, the relevant data displays according to the selected segment.

A form screen that contains a dynamic segment is comprised of the following components.
For configuration details, see Configure a dynamic segment to display selected data in a form screen.

- Form screen header - An item view that displays a summary of records within a form screen. For more information, see Form screen.
- Dynamic segment - List of records where each segment is a section of the list, for example, a day of the week. The data displayed in the embedded list screen area varies according to the selected segment.
• Header segment - Optional component that displays the whole list of records. For more information, see Embedded lists for a form screen.

• Segmented control - Area that includes both the dynamic segment and the header segment.

• Embedded list screen - Display area as defined by the user's selection in the segmented control. For more information, see Embedded lists for a form screen.

• Footer function - A function instance or function location that appears at the bottom of a form screen. For more information, see Associate a function with a location in the app.

Form screen containing a dynamic segment

Sizing of dynamic segment buttons
When configuring a dynamic segment, you must define the height and width of the buttons. The options are small, medium, or large. Use the illustration to get a perspective of the sizes available.
Configure a dynamic segment to display selected data in a form screen

Configure a form screen with a dynamic segment so users can view records that correspond to selected segments, rather than view a complete list of records.

**Before you begin**
When configuring a dynamic segment within a form screen, ensure that you have created the following items:

- A form screen, used to include a dynamic segment within the segmented control. For more information, see Configure a parametrized form applet.
- An embedded list, used to display a list of records within the selected form screen. This list is selected from the Dynamic segment item stream field.
- A corresponding list to be displayed when each segment is selected. This process should resemble the embedded list procedure, with the addition of using a target table to which you are referencing.

For more information about embedded lists and the list to be displayed when a segment is selected, see Embedded lists for a form screen

Role required: admin

**Procedure**

1. In the web-based UI, enter sys_sg_form_screen.list in the filter navigator.
2. Select the form screen to which you want to add a dynamic segment.
3. In the Fetch Type field, select On-demand.
4. In the Segmented Control area, select the height and width of the segment buttons. For a guide to the button sizes, see Using dynamic segments to display data in a form screen.
5. Right-click in the header and select Save.
6. Optional: Configure the header segment to display within the segmented control.

⚠️ Note: The header segment is usually the initial button in the segmented control. If this option is not selected, the segmented control only displays the dynamic segment.
a. Select the **Form Screen Segments** tab and select **New**.

b. From the Select form segment type menu, select **Form Screen Segment** and then select **OK**.

c. On the form, fill in the fields.

**Form Screen Segment form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form Screen</td>
<td>This field is automatically set to the name of the selected form screen.</td>
</tr>
<tr>
<td>Application</td>
<td>Scoped application associated with this record. This field defaults to the current application.</td>
</tr>
<tr>
<td>Embedded Screen</td>
<td>List screen displayed when the user selects the header segment.</td>
</tr>
<tr>
<td>Header View Config</td>
<td>The header segment, which usually appears to the left of the segmented control. Example names are All tasks or All week.</td>
</tr>
<tr>
<td>Order</td>
<td>Position where the header segment displays in the segmented control. This number must be lower than the dynamic screen segment, to ensure it is the first segment on the left.</td>
</tr>
<tr>
<td>Sticky</td>
<td>Keeps the header segment in the same location when the user horizontally scrolls through the displayed dynamic segment records.</td>
</tr>
</tbody>
</table>

d. Select **Submit**.

7. Configure the dynamic segment and the records to display when users tap a segment record.

a. Select the **Form Screen Segments** tab and select **New**.

b. From the Select form segment type menu, select **Dynamic Form Screen Segment** and then select **OK**.

c. On the form, fill in the fields.

**Dynamic Form Screen Segment form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form Screen</td>
<td>This field is automatically set to the name of the selected form screen.</td>
</tr>
<tr>
<td>Application</td>
<td>Scoped application associated with this record. This field defaults to the current application.</td>
</tr>
<tr>
<td>Field</td>
<td>Value</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Dynamic segment item stream</td>
<td>An embedded list screen contained within the dynamic segment of the form screen.</td>
</tr>
<tr>
<td>Active</td>
<td>Option for determining whether the dynamic segment is active. For this configuration, select this option.</td>
</tr>
<tr>
<td>Embedded Screen</td>
<td>List screen that contains a list of records displayed to the user when they tap one of the segments from the dynamic segment.</td>
</tr>
<tr>
<td>Order</td>
<td>The location where the dynamic segment displays in the segmented control. If you have defined a header segment, this number must be higher to ensure it is to the right of the header segment.</td>
</tr>
</tbody>
</table>

d. Right-click in the header and select Save.

8. While still in the Dynamic Form Segment Screen form, define the appropriate lists to display when selecting segments in the dynamic segment.

a. From the **Dynamic segment item stream** field, select the information icon ( ), and select Open Record.

b. Select the Item Stream M2M Item Configuration to use for the dynamic section.

c. In the Item Configuration form, ensure that the user is not redirected to a different embedded screen by selecting **Use View config** and verifying that the **Embedded Screen** field is empty.

d. Select Update.

**What to do next**
Define the appearance of selected segment buttons. See, Customize segment button colors in the segmented control area.

**Customize segment button colors in the segmented control area**
Customize the color of segment buttons to help users identify a tapped segment button. For example, use a darker color to indicate that it is a selected button.

**Before you begin**
Role required: admin

**Procedure**
1. Access the View Template form.

a. In the web-based UI, enter `sys_sg_form_screen.list` in the filter navigator.

b. Select the form screen that contains the dynamic segment you want to configure.

c. Select the **Form Screen Segments** tab and select the information icon ( ) next to the embedded screen that contains a dynamic form screen segment.

d. Select **Open Record** from the menu.

e. From the **Dynamic segment item stream** field, select the information icon ( ) and select **Open Record**.
f. Select the Item Stream M2M Item Configuration to use for the dynamic section.

g. In the View config field of the Item Configuration form, select the information icon ( ) and select Open Record.

2. In the View Template form, select the menu icon ( ) and select Configure > Form Layout.

3. In the Configuring View Template form, select Root-view attribute JSON and move it to the selected area.

The Root-view attribute JSON displays in the View Template form.

4. Select Save.

5. Define the appearance of a tapped segment in the segmented control by pasting the following JSON code in the Root-view attribute JSON field of the View Template form.

```json
{
  "OnSelect":
  {
    "BackgroundColor": "Primary",
    "TextColor": "#FFFFFF"
  }
}
```

6. Optional: If the standard primary, secondary, or destructive values do not fit your color scheme, modify the default values for the background color and text color by providing hexadecimal color values.

7. Right-click in the header and select Save.

Embedded lists for a form screen

Use an embedded list to place a list of information that is related to your current record within a segment on your form applet so that your users don’t have to navigate to a related list.

Use embedded lists to display lists of related information in an easily accessible form segment rather than having your users navigate away to a related list. For example, you could add a list of records representing computer parts or software to your work order form.

To create an embedded list on a form, you will first need to perform the following tasks:

Create a data item to contain the data for your embedded list

You need a data item to store the data that appears in your embedded list. To create a data item, follow the process in Configure a standard data item.

Create a list applet using that data item

You need to configure a list to embed into your form. You can create this list using the
Configure an embedded list for a form screen

Create an embedded list that you can include as an embedded list in a form.

**Before you begin**
Role required: admin

**Procedure**

1. In Studio, open navigate to **Mobile Studio > Applets** and select the form applet where you want to add your embedded list.
2. In the applet configuration form, click **Body**.
3. Click **Modify Segments**.

4. In the modify segments pop-up, click to enable **Embedded Screen**.
5. Click **Save**.
   Your form applet has an **Embedded Screen** tab.

6. Click the **Embedded Screen** button to open the embedded screen configuration options.

7. Click the plus icon (+) to add your embedded list.

8. In the **Embedded List** popup, select a relationship for your embedded list in the **Relationship** field.

9. In the **Destination Screen** field, select the list applet you created to use as an embedded list.

10. Click **Save**.
    After clicking **Save**, you will see a warning stating that the data item is being converted to a relationship data item, which can only be used for embedded lists.
11. Click Ok to close the warning popup.

Results
Your form is configured with an embedded list. Your users can access this list on the form by tapping the tab with the name of this list.

What to do next
To learn more about relationship data items used for embedded lists, see Configure a relationship data item for an embedded list.

Configure an empty state for an embedded list in a form screen
Configure an empty state to display on empty form screens to provide information to further direct users. The display of an empty state for an embedded list in a form screen takes priority over an empty state defined for a list screen.

Before you begin
You should already have an empty state configured for form screens that do not contain any data. For more information, see Configure an empty state.

Procedure
1. In the web-based UI, enter sys_sg_item_stream_segment.list in the filter navigator.
2. Select an item stream segment to display the empty state.
3. In the Empty State field, select an empty state to associate with the item stream segment.
4. Right-click in the header and select Save.

Configure a related list screen for a form applet
Create a related list screen to give your users easy access to records related to the record they are already viewing

A related list appears in a tab on your mobile form, and displays a list records relating to the record on the form. For example. If you are viewing a problem record, you may want your related list to display all incident records associated with that problem record.
Creating a related list involves three tasks:

1. Create a parametrized data item. This data item uses a parameter include only items related to your current record.

2. Create a list that displays the related records from your data item. This is the list that appears in the related list tab on your form.

3. Update your form applet to display related lists, and define the relationship between the current record and the related list.

Create a parametrized data item for your related list
Create a parametrized data item to contain the records that display in your related list.

Before you begin
Role required: admin

Procedure

1. In Studio, Navigate to All > Mobile Studio > Data Items.

2. Click the pop-out icon (⋯) that appears to the right of Data Items.

3. In the Data items screen, click Create New.

4. In the Name field, enter a name for your data item.

5. In the Table field, select the table for the records you want to appear in your related list. Remember that this table is for the record in the related list rather than the record on your form. For example, if you want to display a list of incidents related to a problem record, you should select Incident [incident] rather than Problem [problem].

6. Click Save.

7. In the Parameter Definition section, click the add button (➕).

8. In the Parameter Definition pop-up, enter a name for your parameter. Enter a name that matches your field name to make identifying your parameter easier.

9. In the Type field, select a data type for your parameter.
   - String: Use the String type for list parameters, such as priority or state, or for reference fields, such as Assigned to or Caller.
   - Integer: Use the integer type for a parameter containing an integer value.
   - Decimal: Use the decimal type for a parameter containing a decimal value.
   - Boolean: Use the boolean type for a parameter containing a boolean (true or false) value.
   - DateTime: Use the datetime type for a parameter containing a date and time.
   - Date: Use the date type for a parameter containing a date only.

10. Click Save.
    The parameter is a variable that will contain a value from the record on your form. You will use this parameter to define what records are contained in your data item by using the parameter in your data item conditions.
11. In the condition section of your data item, create a condition using the parameter you created in steps 7 through 10. You can access your parameters in the condition builder using the reference value button ( ).

12. Click Save.

Example:
In this example, you have a form that displays a problem record. On the form, you want to display a related list showing incidents associated with that problem. Since the records in the related list are incidents, the data item uses the Incident [incident] table.

For the data item to show only incidents related to the problem on the form, we need to create a parameter for that information. To make it easy to identify, the parameter name is Problem.

In the condition field, create a condition for the Problem field on incident records, and use the parameter as the value. You select the parameter using the reference value button ( ).

Create a list screen to use as a related list
Create a related list using your parametrized data item. This list will appear for your users when they select the related list tab on their form screen.

Before you begin
Role required: admin

Procedure
1. In Studio, navigate to Mobile Studio > Applets.

2. Click the pop-out icon ( ) that appears to the right of Applets.
3. Click **Create an Applet**.

4. In the **Create an Applet** pop-up, select a name and description for your related list.

5. Select the **List** template.

6. Click **Create New**.

7. In the **Data Item** field, select the parametrized data item you created in the previous steps.

8. In the Field Configurations section, select the fields you want to display on your related list for each record.

9. Click **Save**.

10. In the UI Parameter section, click the add button (+) to add a UI parameter for your related list.

11. In the **User Input** pop-up, enter a name for UI Parameter in the **Name** field.

12. Click **Save**.

**Example:**

Continuing the preceding example, you create a new applet to display incidents for your problem record. This applet uses the data item created in the previous steps. This applet needs a parameter to contain the problem record from your problem form. To make it easy to identify, the parameter name is also named Problem. In the IU Parameter Mapping section, you create a mapping between the data item parameter and the applet parameter, so the value can pass between them.
Update your form to display a related list

Update your mobile form to display a list of related records.

Before you begin
Role required: admin

Procedure

1. In Studio, Navigate to All > Mobile Studio > Applets, and open the form applet where you want to display a related list.
   In some cases, the form applet where you want to display your related list is part of a list applet. In that case, open your list applet, and open the Form Screen tab to configure the form.

2. In the Body section of your form screen, click Modify Segments in the upper right corner.

3. In the Modify Segments pop-up, select Related List.

   Modify Segments

<table>
<thead>
<tr>
<th>Displayed Segments</th>
<th>Segments Order</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Details</td>
<td>[Details]</td>
</tr>
<tr>
<td>Activity Stream</td>
<td>[Activity Stream]</td>
</tr>
<tr>
<td>Related List</td>
<td>[Related List]</td>
</tr>
<tr>
<td>Embedded List</td>
<td>[Embedded List]</td>
</tr>
</tbody>
</table>

4. Click Save.

5. In the form screen, click Related List.
   The Related List appears next to the Details button after you enabled related lists in the previous steps.

6. Optional: Select Show related list count to display a count of records in your related list.
7. Click the add button (+) to create a relationship. This relationship is what connects your form to the related list.

8. In the Related List pop-up, fill in the fields.

**Related list fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationship</td>
<td>Select a relationship between the current form and the related list. For example, if you are in a form for problem records, and want to display a related list of incidents, you would choose <strong>incident-&gt;problem</strong>.</td>
</tr>
<tr>
<td>Destination screen</td>
<td>Select the list applet you created in previous steps. This list displays when the user selects the related list tab on their form screen.</td>
</tr>
</tbody>
</table>

9. In the Related List pop-up, click **Save**.

10. On the form screen, click **Save**.

**Results**

Your form displays a tab for your related list. Users tap this tab to see records on that list.

**Note:** Select the **Show related list count** field to display a counter number in each of the related lists. This number relates to the number of records based on your selection in the **Relationship** field.

The number of related records shown in a selected destination screen can be less or equal to the counter number displayed in the related list. The lower number of records is due to additional conditions applied to the list.
Example:
Continuing the preceding example, the problem form must have a related list. In the Related List pop-up, you select the Incident->Problem relationship. In the Destination Screen, you select the related incident list created in the previous steps. After logging out and back in again, you now have a related list on your problem form, which displays a list of incidents associated with that problem.
Disable attachments in mobile apps

Disable attachments for mobile apps by using access control rules.

**Before you begin**
Role required: admin

⚠️ **Note:** You need to elevate to the security_admin role to perform these steps. For details on this role, see **security_admin role**.

**Procedure**
1. Navigate to All > System Security > Access Control (ACL).
2. Filter the list for `<Name> <contains> <sys_attachment>` and `<Operation> <is> <read>`.
3. Find and open the record with the description: *Allow read for records in sys_attachment, if the ACL script returns true.*
4. Clear the **Admin overrides** check box.
5. In the **Script** field, add the following code to the bottom of the script:

   ```javascript
   if( gs.isMobile() ){
     answer = false;
   }
   ```
6. Click **Update**.
   The added code prevents attachments from appearing when the instance is accessed from a mobile device. If you want to prevent your users from uploading attachments, continue on to the next steps.
8. Filter the list for `<Name> <contains> <sys_attachment>` and `<Operation> <is> <create>`.
9. Find and open the record with the description: **Allow create for records in sys_attachment, if the ACL script returns true.**

10. Clear the **Admin overrides** check box.

11. In the **Script** field, add the following code to the bottom of the script:

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

12. Click **Update**.
   The added code prevents attachments uploading when the instance is accessed from a mobile device.

   **Note:** The ability to rename or delete existing attachments on records in mobile is determined by the access control list (ACL) rules on your instance. For more information on these rule and how the effect user permissions on your instance, see **Access control list rules**.

### Grouped list screen

Grouped list applets display a list of records that are grouped by a specific field.

Use a grouped list to provide a view of a list of items grouped by a field. Groups lists can help present a large number of records in an easily navigable format. For example, you can display a list of incidents grouped by category or priority. You can configure grouped lists to navigate to a second list showing the records in the selected group.

![Incidents by Priority](image)

© 2022 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.
Configure a group list screen

Create a group list to provide a view of a list of records grouped by a field you specify.

Before you begin

Before you can create a group list applet, you must create a data item with a group configuration. For more information on configuring a data item, see Configure a standard data item, and note the optional steps for setting the Group by field. 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

Procedure

1. Navigate to All > System Applications > Studio to open Studio.
2. Select a scoped application where you want to create your applet.
   If you have not created an application, you can create one by using the Create Application button. For information on creating a scoped application for mobile, see Create a mobile application using Guided Application Creator.
3. In Application Explorer, navigate to Mobile Studio > Applets and select Applets.
4. Click the pop-out icon ( ) that appears to the right of Applets.
5. In the applet list, click Create an applet.
   The Create an Applet window appears on the screen.

6. In the Create an Applet window. Enter an name and description for your group list applet.
7. Click the image next to the Icon field to select a color and image for your list applet icon. If you add your group list applet to an applet launcher, this icon displays for the applet.
8. In the Choose and template section, select Group List.
9. Click Create New.
   The Create an Applet window closes, and a new Studio tab opens for your group applet.
10. In the Data Item field, select a data item.
Note: If the Data item field list is empty, you may have no data items have been created with a group configuration.

To create a data item, click the plus icon (+) next to the list.

11. Click Save.

Results
You now have an applet displaying a list of records, grouped by the field you defined in your data item.

Group List applet

```
<table>
<thead>
<tr>
<th>Priority</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - Critical</td>
<td>27</td>
</tr>
<tr>
<td>2 - High</td>
<td>4</td>
</tr>
<tr>
<td>3 - Moderate</td>
<td>12</td>
</tr>
<tr>
<td>4 - Low</td>
<td>5</td>
</tr>
<tr>
<td>5 - Planning</td>
<td>17</td>
</tr>
</tbody>
</table>
```

Tutorial: Configure a list of grouped incidents

Use this example to create a list of incident records, grouped by priority.

This example creates a grouped list as shown in the example image. Tapping on a priority displays a list of incidents matching that priority, which in turn can be tapped to display the incident record.
Create a data item for a grouped list

Create a data item that has a group configuration to use with your grouped list applet.

**Before you begin**

Role required: admin

Before you create an applet for your grouped list, you need a data item that has been created with a group configuration.

**Procedure**

1. In Studio, navigate to **Mobile Studio > Data Items**, and click the pop-out icon (⋮) that appears to the right of **Data Items**.

2. Click **Create New**.

3. In the New Data Item form, fill in the fields.
New Data Item form

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the data item. For this example, enter Incidents by Priority</td>
</tr>
<tr>
<td>Table</td>
<td>Table where the data item gets its data. Select Incident [incident].</td>
</tr>
<tr>
<td>Description</td>
<td>Description used to identify your data item.</td>
</tr>
<tr>
<td>Condition Type</td>
<td>Whether the conditions for the data item are declarative, scripted, or use an encoded query. Leave this field at its default setting.</td>
</tr>
</tbody>
</table>

4. In the Group Configuration section, set the Group By field to Priority

5. Click Save.

Completed data item for a grouped list

Results
You now have a data item for the incident table, grouped by priority. In the next steps, you use this data item to create the grouped list applet.
Create a grouped list applet
Create an applet using the grouped list template. This list displays your incident priorities, and the number of records matching each priority.

Before you begin
Role required: admin

Procedure
1. In Studio, navigate to Mobile Studio > Applets, and click the pop-out icon that appears to the right of Applets.
2. Click Create an Applet.
3. In the Create an Applet form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of your applet. For this example, enter Incidents by Priority.</td>
</tr>
<tr>
<td>Icon</td>
<td>Icon used by your applet. Select any icon.</td>
</tr>
<tr>
<td>Description</td>
<td>Description used to identify your applet.</td>
</tr>
<tr>
<td>Choose a template</td>
<td>Template used for your applet. Select Grouped List.</td>
</tr>
</tbody>
</table>

4. Click Create new.
The create an applet pop-up closes, and you are directed to your new applet's form in a studio tab.

5. In the Data Item field, select the Incidents by Priority data item you created in the previous steps.
6. Click Save.

Results
You now have an applet that displays your incident priorities, and a count of records matching each priority. In the next steps you create a navigation function to display a list of records matching the priority when a user taps on one of these priorities.

Create a list applet and data item
Create an applet using a list template. This list displays all the incident records in a matching priority when your user taps a priority in the previous list. You also create a new data item containing all the records matching a specific priority.

Before you begin
Role required: admin

Procedure
1. In Studio, navigate to Mobile Studio > Applets, and click the pop-out icon () that appears to the right of Applets.
2. Click Create an Applet.
3. In the Create an Applet form, fill in the fields.
Create an Applet form

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of your applet. For this example, enter Incidents matching Priority.</td>
</tr>
<tr>
<td>Icon</td>
<td>Icon used by your applet. Select any icon.</td>
</tr>
<tr>
<td>Description</td>
<td>Description used to identify your applet.</td>
</tr>
<tr>
<td>Choose a template</td>
<td>Template used for your applet. Select <strong>List</strong>.</td>
</tr>
</tbody>
</table>

4. Click **Create new**
   The create an applet pop-up closes, and you are directed to your new applet's form in a studio tab.

5. Click the plus icon (+) to the right of the **Data Item** field.

6. On the new data item form, fill in the fields.

**New Data Item form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the data item. For this example, enter Incidents matching Priority</td>
</tr>
<tr>
<td>Table</td>
<td>Table where the data item gets its data. Select <strong>Incident [incident]</strong>.</td>
</tr>
<tr>
<td>Description</td>
<td>Description used to identify your data item.</td>
</tr>
<tr>
<td>Condition Type</td>
<td>Whether the conditions for the data item are declarative, scripted, or use an encoded query. Leave this field at its default setting.</td>
</tr>
</tbody>
</table>

7. Click **Save**.

8. In the **Parameter Definition** section, click the plus icon (+) to create a new parameter.

9. In the **Parameter Definition** pop-up, enter **Group priority** in the **Name** field.

10. Click **Save**.

11. In the **Query condition field** select `<Priority> <is>`, and then click the reference value icon ( 조) to select the **Group priority** parameter created in the previous steps.
12. Return to the Studio tab for your **Incidents matching priority** applet.
13. In the **Data Item** field, select the **Incidents matching priority** data item you created in the previous steps.
14. In the **Field Configurations** section, select the fields you want to include in your incident list.

**Example field configuration for the list applet**

15. Click the **Form Screen** tab at the top of the form.
16. Click the **Replicate from primary** button ( ()) to automatically add fields from your list to the form.
17. Click **Save**.
You now have a list applet to display all incident records with the priority that your users select from the group list you created earlier. In the final set of steps, you will create a navigation function that handles the transition from the grouped list applet to the list applet.

Create a navigation function for your grouped list

Create a navigation function that handles the transition from the grouped list applet to the list applet containing records of a specific priority.

Before you begin
Role required: admin

Procedure
1. In Studio, open the tab for your Incidents by Priority applet. This is the first applet you created to display your records grouped by priority.
2. Click Functions to open the functions section of the form.
3. Click the plus (+) icon to create a new list item function.

4. In the list item function pop-up, click the plus icon (+) to create a new function.

5. In the new navigation form, fill in the fields.

**New Navigation form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of your function. For this example, enter Grouped list navigation.</td>
</tr>
<tr>
<td>Destination Type</td>
<td>Whether the navigation will direct to an applet or applet launcher. Select <strong>Applet</strong></td>
</tr>
<tr>
<td>Description</td>
<td>Description of your function.</td>
</tr>
<tr>
<td>Destination</td>
<td>The applet or applet launcher your function will direct to. Select the <strong>Incidents matching priority</strong> applet you created in previous steps.</td>
</tr>
<tr>
<td>Context</td>
<td>The context for the function. Select <strong>Global</strong>.</td>
</tr>
<tr>
<td>Table</td>
<td>The table used for display conditions. Select <strong>Incident [incident]</strong></td>
</tr>
</tbody>
</table>

6. Click **Save**.

7. Return to the **Incidents by Priority** tab, and select your new navigation function.
8. Click **Save**.

**Employee directory screen**

Employee directory screen display user record information in ServiceNow mobile apps.

Use an employee directory screen to present users with information on user records on your instance. You can design your employee directory screen with functions to allow users to communicate via phone or email, and navigate to their address using their mobile device's navigation functionality.

**Configure an employee directory applet**

Use an employee directory applet to provide a list of employees.
Before you begin
Role required: admin

Procedure

1. Navigate to All > System Applications > Studio to open Studio.
2. Select a scoped application where you want to create your applet.
   If you have not created an application, you can create one by using the Create Application button. For information on creating a scoped application for mobile, see Create a mobile application using Guided Application Creator.
3. In Application Explorer, navigate to Mobile Studio > Applets and select Applets.
4. Click the pop-out icon (⋮) that appears to the right of Applets.
5. In the applet list, click Create an applet.
   The Create an Applet window appears on the screen.

6. In the Create an Applet window, enter a name and description for your employee directory applet.
7. Click the image next to the Icon field to select a color and image for your employee directory applet icon. If you add your combined list applet to an applet launcher, this icon displays for the applet.
8. In the Choose and template section, select Employee Directory.
9. Click Create New.
   The Create an Applet window closes, and a new Studio tab opens for your employee directory applet.
10. 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 Configure a standard data item. 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.
11. Optional: When created, your applet uses the default template visible in the Template preview panel for the cards shown in your applet. Click Customize in Mobile Card Builder.
button to use Mobile card builder to update the appearance of the card and which fields appear. For details on customizing an item view, see Customize a screen using Mobile Card Builder.

12. Click the Profile Screen tab.

13. 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.

   You can also use the Replicate from primary button to copy the same fields you used in the Employee Directory Screen.

14. Click Save.

What to do next
You may want to include functions in your screen to allow users to communicate with or navigate to people in your user records. Learn more about mobile functions in Mobile functions.

Map screen
Map screens display a map with locations that are associated to the records in a data item.
Use a map screen when you need your users be aware of the locations related to the records in your data item. For example, map screens can show your users where their assets are located, or which job locations they need to travel to.

Configure a map screen
Use Studio to configure a map applet so that your users can see location-based information on a map from their mobile app.

Before you begin
Role required: admin Demonstrates how employees can use ServiceNow mobile apps to perform work tasks remotely.

About this task
When you create a data item for a map screen, use information that can be plotted on a map. The map screen includes a card list to display records that are shown on the map.

Procedure
1. Navigate to All > System Applications > Studio to open Studio.
2. Select a scoped application where you want to create your applet.
   If you have not created an application, you can create one by using the Create Application button. For information on creating a scoped application for mobile, see Create a mobile application using Guided Application Creator.
3. In Application Explorer, navigate to Mobile Studio > Applets and select Applets.
4. Click the pop-out icon ( ) that appears to the right of Applets.
5. In the applet list, click **Create an applet**. The **Create an Applet** window appears on the screen.

   ![Create an Applet window](image)

6. In the **Create an Applet** window, enter a name and description for your map applet.

7. Click the image next to the **Icon** field to select a color and image for your map applet icon. If you add your map applet to an applet launcher, this icon displays for the applet.

8. In the **Choose and template** section, select **Map**.

9. Click **Create New**. The **Create an Applet** window closes, and a new Studio tab opens for your map applet.

10. In the **Data Item** field of the map template, 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 **Configure a standard data item**.

11. In the **Location** field, select a field to use as the record's location. This field must be a reference to a Location [cmn_location] record.

12. **Optional:** When created, your applet uses the default template visible in the **Template preview** panel for the cards shown in your applet. Click **Customize in Mobile Card Builder** button to use Mobile card builder to update the appearance of the card and which fields appear. For details on customizing an item view, see **Customize a screen using Mobile Card Builder**.
13. Click **Save**.

**Results**
The configured map screen displays information from your data item plotted on a map, along with a card list to display details on each item.
What to do next
Configure the form screen so that you can give your users a more detailed view of the records from your data item. The map applet automatically includes a form screen, which displays when a user taps on a card from the map screen. For details on how to configure form screens, see Configure a form applet for use with a list applet.

Change the pin colors for locations in your map applets
Learn how to change pin colors to indicate locations on your map applets.

Before you begin
Role required: admin

Procedure
1. In the web-based UI, navigate to System Mobile > Applets.
2. In the Item Stream Segments tab, open the item stream segment record.
3. In the Item Stream M2M Segments list on the item stream segment form, open the item stream record.
4. To the right of the Item Stream field on the item stream M2M segment form, click the reference icon (ASHBOARD), and then click Open Record in the pop-up window.
5. In the Pin color field, enter a color value in hexadecimal format for your map pin.
6. Click Update.

Mobile map screen filters
Use mobile filters in map screens to enable your users to find what they need in the records on your instance.

Choose one of the following options when configuring your mobile map screens:

Default filters
By default, the instance automatically creates a filter and sorting functionality. This filter is based on the fields selected in your map pattern. No further configuration is needed. Users can filter and sort the list based on any of these fields.

Custom filters
If the default filters do not fit your requirements, consider creating a customer filter experience for your users. You can manually select the fields available to the map filter, as well as enable keyword filtering. Custom filters can include fields that are not in the map pattern. For more details on creating custom filters, see Configure a custom map screen filter.

Note: For those who have upgraded to the Rome release, consider the following:

- If you did not define filters and are using the default setup, you automatically use the default filters and sorting capabilities after the upgrade.
- If you defined filters in earlier versions, then after the upgrade your defined filters are still available. However, you must manually configure the sorting capabilities. See Configure sorting capabilities within mobile filters.

Disable filters
To disable filters and sorting on map screen, open the map screen record, click the Filter tab, and disable the Show or hide filters for this list screen option. When filtering is turned off, users are unable to filter or sort records.

Configure a custom map screen filter
Manually create a filter experience for your users. Custom map filters can include fields that are not in the map pattern.

Before you begin
Role required: admin

Procedure
1. Navigate to All > System Applications > Studio.
2. Select your application.
3. In the Application Explorer, navigate to Mobile Studio > Applets, and open the map applet where you want to configure a custom filter.
4. In the map screen form, select the Filters tab.
5. If inactive, enable the Show or hide filters for this list screen option.

Note: Enabling the filter functionality also enables the sorting functionality.
6. Select Set filters manually. When Set filters manually is enabled, the default list of filter fields is cleared.
7. **Optional:** To enable keyword filtering on your list, enable the *Enable Keyword filter* field. Studio creates a keyword filter and adds it to your filter list.

8. Select the plus icon (➕) to add a field to your filter.

9. In the Add Filter pop-up, fill in the fields as needed.

**Add filter field**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Field</td>
<td>The field used in this filter item. The data item used by the list determines which fields are available.</td>
</tr>
<tr>
<td>Filter Label</td>
<td>The label used for this filter field. This field defaults to the name of the field selected in the <em>Data Field</em> field.</td>
</tr>
<tr>
<td>Operator Selection</td>
<td>The operator type used by the filter. Some fields support multiple operator types. For example, for date fields, select <em>On</em> to filter for a specific date, or <em>Between</em> to filter for records within a date range.</td>
</tr>
<tr>
<td>Filter Type</td>
<td>Filter type used by this filter item. This value is automatically determined by the value in the <em>Operator Selection</em> field. <strong>Note:</strong> Boolean filter types require an alternative configuration, see <em>Add Boolean fields within a mobile filter</em>.</td>
</tr>
</tbody>
</table>

10. Select *Save*.

11. Add additional filters by repeating steps 6 through 10 as needed.

12. When you have completed adding fields to your filter, select *Save* in the map screen form.

**What to do next**

After configuring a map screen filter, you can add Boolean type filters or customize the filter sorting capability. Refer to the following documentation topics:

- Add Boolean fields within a mobile filter
- Configure sorting capabilities within mobile filters

**Calendar screen**

Calendar screens display a calendar interface, and records associated with the selected date.
Use a calendar to display records when the dates relating to those records are relevant. For example, displaying when your tasks are due, or when important events take place.

The calendar screen displays a calendar interface. Each date on the screen displays an indicator below any date that has records associated to it.

Below the calendar, your users can see a list of records associated to the date selected in the calendar. You can configure the calendar app to display these records in a form screen when the users taps them.

### Configure a calendar applet
Configure the calendar applet so that your users can see their scheduled tasks in a calendar view in your mobile apps.

### Create the data item and calendar applet
Create the data item and calendar applet in Studio.

### Before you begin
Role required: admin

### Procedure
1. Create a data item for your calendar applet. To create this applet, follow the steps detailed in Configure a standard data item.
   You now have a data item to use with your calendar applet. Leave the tab for this data item open in Studio, as you will need to return to this tab in later steps.
2. In Application Explorer, navigate to Mobile Studio > Applets and select Applets.
3. Click the pop-out icon ($\square$) that appears to the right of Applets.
4. In the applet list, click Create an applet.
   The Create an Applet window appears on the screen.
5. In the Create an Applet window, enter a name and description for your calendar applet.

6. Click the image next to the Icon field to select a color and image for your calendar applet icon. If you add your calendar applet to an applet launcher, this icon displays for the applet.

7. In the Choose and template section, select Calendar.

8. Click Create New. The Create an Applet window closes, and a new Studio tab opens for your calendar applet.

9. After creating the applet, you can see the calendar applet form. Click Submit to save the record. Saving the record displays the record’s related lists, which you need to complete the calendar applet setup. Leave the tab for this applet open in Studio, as you will need to return to this form in later steps.

**Configure a list and form for your calendar**

Configure list and form applets to display the records associated with your calendar dates.

**Before you begin**
Role required: admin

**Procedure**

1. In the applet list, click Create an applet.
   The Create an Applet window appears on the screen.
Create an Applet window

Add some details to your applet

- **Name**: Name your applet
- **Description**: Describe your applet
- **Available offline**: 

Choose a type of applet you want to create

1. In the **Create an Applet** window, enter a name and description for your list applet.
2. In the **Choose and template** section, select **List**.
3. To the right of the **Choose and template** section, enable the **Form** check box.
4. Click **Create New**.
   - The **Create an Applet** window closes, and a new Studio tab opens for your list applet.
5. In the **Data Item** field, use the data item you created in previous steps.
6. In the **Field Configuration** section, add the fields you want in your list to the **Selected Fields** list.
7. Click **Save**.
8. Click the **Form Screen** tab to open the form screen configuration options.
9. Click the **Replicate from primary** button to copy the fields from your list to your form.
10. Click **Save**.

Create an item stream segments

Create an item stream segment and many-to-many segment for your calendar applet.

**Before you begin**
Role required: admin

**Procedure**
1. Return to the tab containing your calendar applet. If this tab is no longer open, you can open **Mobile Studio > Applets** and open your calendar applet.
2. In the calendar applet tab, open the **Item Stream Segments** related list, and click **New**.
3. In the **Name** field, enter a name for your item stream segment.
4. In the **Screen** field, select your calendar applet.
5. Click **Save**.
6. In your instance, outside Studio, enter `sys_sg_time_span_item_stream.list` in the filter navigator.

7. In the time span item streams list, click **New**.

8. In the **Time span item stream** form, fill in the fields as needed.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of your item stream</td>
</tr>
<tr>
<td>Application</td>
<td>Current scoped application. This field is automatically populated.</td>
</tr>
<tr>
<td>Data item</td>
<td>Data item used by the item stream. Use the data item you created in the previous steps.</td>
</tr>
<tr>
<td>Table</td>
<td>Table used by the item stream. Select the same table used by your data item.</td>
</tr>
<tr>
<td>Interleaved field</td>
<td>Select the field you want to use to interleave your data item.</td>
</tr>
<tr>
<td>Start Date Field</td>
<td>The date field on the table the calendar applet uses for start dates. For example the <strong>Opened</strong> field on an incident record.</td>
</tr>
<tr>
<td>End Date Field</td>
<td>The date field on the table the calendar applet uses for end dates. For example the <strong>Closed</strong> field on an incident record.</td>
</tr>
</tbody>
</table>

9. Click **Submit**.

10. Return to Studio, and open the tab for your item stream segment.

11. In the **Item Stream M2M Segments** list, click the plus (+) button.

12. In the **Item Stream** field, enter the name of the time span item stream record you created in steps 6 through 9.

13. Right-click the header of the form and select **Save**.

14. In the **Item Stream M2M Segments**, double click the name of the time span item stream record to open that record.

15. In the **Item Stream M2M Item Configurations** list, click **New**.

16. On the **Item Stream M2M Item Configuration** form, enter the item configuration for the list applet you created in the **Item Configuration** field. You can find the item configuration using the name of the list you created.

17. Click **Submit**.

**Configure an item view for your calendar applet**

Configure the item view for your calendar applet to define the appearance of the list items under your calendar.

**Before you begin**

Role required: admin
Procedure

1. In your instance, outside Studio navigate to **System Mobile > Applets**

2. Open the record for the list applet you created in the previous steps.

3. In the applet form, click the name of the record in the **Item Stream Segments** list to open the item stream segment record.

4. In the item stream segment record, click the record name in the **Item Stream M2M Segments** form to open the item configuration record.

5. In the item configuration form, click the reference icon (🔍) next to the **Item view** field to open the item view record.

6. In the item view record, replace the contents of the **Item view JSON** field with the following JSON:

```json
{
    "Id":28090,
    "Type":"ViewGroup",
    "Orientation":"Horizontal",
    "Alignment":"Center",
    "Distribution":"Auto",
    "Margin":{
        "Top":16,
        "Bottom":16
    },
    "Children":[
        {
            "Id":12259,
            "Type":"ViewGroup",
            "Orientation":"Vertical",
            "Alignment":"Left",
            "Distribution":"Auto",
            "Children":[
                {
                    "Type":"Text",
                    "Id":41927,
                    "Margin":{
                        "Top":3
                    },
                    "CellId":"opened_at",
                    "TextColor":"#293e40",
                    "TextAlignment":"Left",
                    "MaxLines":1,
                    "Font":{
                        "Weight":"regular",
                        "Size":12
                    },
                    "Width":60,
                    "Text":""
                },
                {
                    "Type":"Text",
                    "Id":41928,
                    "Width":60,
                    "Margin":{
                        "Top":9
                    }
                }
            ]
        }
    ]
}
```

© 2022 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.
Note: The formatted JSON here uses the `opened_at`, `closed_at`, and `short_description` fields. If you intend to use different fields in your calendar list, replace these values in the above JSON with the fields you want to display.

7. Right-click the form header and select click **Save** in the context menu.

Results
You applet is configured and ready to use in a mobile app.

Mobile web screen
Mobile web screens open an external URL or a relative URL within your instance.
Use a mobile web screen to Open a URL from within the ServiceNow app. You can configure relative URLs to open pages within the ServiceNow platform, or an external link. For example, a user can see a knowledge article on the instance via Service Portal.

Relative URLs that direct your users to open pages within your instance display within mobile app. URLs that open external links open the link in the default browser of the user’s mobile device.

Configure a mobile web screen

Use a mobile web screen to open a URL from within a ServiceNow mobile application. You can configure relative URLs to open pages within the ServiceNow platform.

**Before you begin**

Role required: admin

**Procedure**

1. Navigate to All > System Mobile > Mobile App Builder.
   The Mobile App Builder opens in a new browser tab and displays the application scope selection screen.

2. Search for the application scope you are working in and then select the name of the application scope.
   The Mobile App Builder categories home screen displays.

3. Select the Screens category from the menu, and then select New.

4. Select the Mobile web option in the Create a screen page and then select Continue.

5. Complete the following fields as needed.
New mobile web form

<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>Description</td>
<td>Additional information about the screen.</td>
</tr>
<tr>
<td>Icon</td>
<td>Icon used to represent your mobile web screen when added to a launcher screen.</td>
</tr>
<tr>
<td>Fetch type</td>
<td>Fetch type settings determine when data is loaded in screens. Change your fetch type to optimize load time performance for your screens.</td>
</tr>
<tr>
<td></td>
<td>Select either Prefetch, On-demand, Background or Dynamic prefetch as required according to your setup. For more information about these options, see Configure a mobile web screen.</td>
</tr>
<tr>
<td>URL</td>
<td>The URL you want the send to user to.</td>
</tr>
<tr>
<td>Hide screen name</td>
<td>Determines if the screen name is not shown as the screen header. This is useful if, for example, the text already appears in the header name or if it does not provide any value to the user.</td>
</tr>
<tr>
<td>Open in external browser</td>
<td>Determines 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>Role access</td>
<td>Determine which user roles can access this screen. If you have selected no roles, users with any role will have access to the screen.</td>
</tr>
</tbody>
</table>

6. Select **Save**.

**Chart screen**

Chart screens display an interactive view of a report or performance analytics widget.
Use chart sections to display time series or single score reporting charts, and performance analytics scores. Charts give your managers or owners indications on trends or items which require their attention.

The chart screen supports the following report types:

- Time Series
- Bar
- Pie

To use a chart screen, you will first need to create a dashboard preview that will link to your chart screen. For details on dashboard previews, see Mobile dashboard preview.

Configure a chart applet for a report

Configure a chart applet to give your users access to pre-configured mobile dashboard views for single score reports

Before you begin
Role required: admin
You must have an existing report to display in your applet. If you have not yet configured a report, see Getting started with reports.

Procedure

1. Navigate to All > System Applications > Studio to open Studio.
2. Select a scoped application where you want to create your applet.
   If you have not created an application, you can create one by using the Create Application button. For information on creating a scoped application for mobile, see Create a mobile application using Guided Application Creator.
3. In Application Explorer, navigate to Mobile Studio > Applets and select Applets.
4. Click the pop-out icon that appears to the right of Applets.
5. In the applet list, click Create an applet.
   The Create an Applet window appears on the screen.
6. In the Create an Applet window. Enter a name and description for your chart applet.

7. Click the image next to the Icon field to select a color and image for your chart applet icon. If you add your chart applet to an applet launcher, this icon displays for the applet.

8. In the Choose and template section, select Chart.

9. Click Create New. The Create an Applet window closes, and a new Studio tab opens for your chart applet.

10. In the Chart Screen tab, fill in the fields as needed.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Whether the chart displays a report or performance analytics chart. Choose Report to configure a single score report chart.</td>
</tr>
<tr>
<td>Report</td>
<td>The report to display in the chart applet. Select report from the list. This list is automatically filtered for single score reports on your instance.</td>
</tr>
<tr>
<td>Set the default view point to most recent</td>
<td>Enable to display the most recent data as the default view point.</td>
</tr>
</tbody>
</table>

11. Optional: In the Functions tab, click the Add button (+) to add global functions for your chart applet. These functions appear in the top menu of your chart applet.

12. Click Save.

What to do next
Configure a navigation function to direct your users from a dashboard preview to your chart applet. For details on this process, see Configure a navigation to a chart applet.
Configure a chart applet for a Performance Analytics widget

Configure a chart applet to give your users access to pre-configured mobile dashboard views of a Performance Analytics widget.

Before you begin
Role required: admin

Procedure
1. Navigate to All > System Applications > Studio to open Studio.
2. Select a scoped application where you want to create your applet.
   If you have not created an application, you can create one by using the Create Application button. For information on creating a scoped application for mobile, see Create a mobile application using Guided Application Creator.
3. In Application Explorer, navigate to Mobile Studio > Applets and select Applets.
4. Click the pop-out icon that appears to the right of Applets.
5. In the applet list, click Create an applet. The Create an Applet window appears on the screen.

Create an Applet window

6. In the Create an Applet window. Enter a name and description for your chart applet.
7. Click the image next to the Icon field to select a color and image for your chart applet icon. If you add your chart applet to an applet launcher, this icon displays for the applet.
8. In the Choose and template section, select Chart.
9. Click Create New. The Create an Applet window closes, and a new Studio tab opens for your chart applet.
10. In the Chart Screen tab, fill in the fields as needed.

Chart screen fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Whether the chart displays a report or Performance Analytics chart. Choose</td>
</tr>
</tbody>
</table>
Field | Description
--- | ---
**Performance Analytics** to configure a performance analytics chart. | **Performance Analytics** to configure a performance analytics chart.

**Widget** | The Performance Analytics widget to display in your chart applet.

Set the default view point to most recent | Enable to display the most recent data as the default view point.

11. **Optional:** In the **Functions** tab, click the **Add** button ( ++) to add global functions for your chart applet. These functions appear in the top menu of your chart applet.

12. Click **Save**.

**Configure a navigation to a chart applet**

Configure a navigation function to direct your users to your chart applet.

**Before you begin**

Role required: **admin**

These steps detail instruction for creating a navigation from a dashboard preview to a chart applet. To make use of the navigation function, you will need to have configured an applet launcher with a dashboard preview. For details on that process, see [Create a mobile dashboard preview](#).

**Procedure**

1. In Studio, navigate to **Mobile Studio > Functions > Navigations**.

2. Click the pop-out icon ( ++) that appears to the right of **Navigations**.

3. In the **Navigations** list, click **Create New**.

4. On the **New Navigation** form, fill in the fields as needed.

**Navigation form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Name of your navigation function</td>
</tr>
<tr>
<td><strong>Destination Type</strong></td>
<td>The type of record your navigation destination. Since you are creating a navigation to a chart applet, select <strong>Applet</strong>.</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>A unique description for your navigation function.</td>
</tr>
<tr>
<td><strong>Destination</strong></td>
<td>The applet or applet launcher to use for your navigation destination. Select the chart applet you created in previous steps.</td>
</tr>
<tr>
<td><strong>Context</strong></td>
<td>Whether the function uses the global or record context. For this example, select global.</td>
</tr>
</tbody>
</table>

5. Click **Save**.
Add a navigation function to a chart preview
Configure your chart preview to use a navigation function. This function directs your users to your chart applet from the chart preview on your applet launcher.

Before you begin
Role required: admin
These steps detail instruction for adding your navigation function to an existing dashboard preview. To make use of the navigation function, you will need to have configured an applet launcher with a dashboard preview. For details on that process, see Create a mobile dashboard preview.

Procedure
1. In Studio, navigate to Mobile Studio > Dashboard Previews.
2. Open the dashboard preview where you want to add your navigation.
3. In the Navigation field, select the navigation function you created in the previous steps.
4. Click Save.

Results
You can now tap the chart preview in your applet launcher to navigate to your chart applet.

Configure a navigation from a chart to a list applet
Configure a navigation to allow your users to navigate to a list of records from your chart applet.

To create your navigation, you need to create the following components:
• Configure a parametrized list and form applet.
• Configure a navigation function to navigate from your chart to your parametrized list and form.
• Associate your navigation function to your chart applet.

These steps assume you have an existing chart applet you want to configure. If you have not yet created a chart applet, you can find details on creating charts at Configure a chart applet for a report.

Create a parametrized list for your chart
Create the list and form you users see when they tap on your chart applet.

Before you begin
Role required: admin
Creating a parametrized list is similar to creating a standard list. In addition to the steps used to create a standard list, you must create
• A parametrized data item for your list. This data item uses your parameter to filter what data is displayed in your list. In this case, the records from your chart applet
• A parameter on your list applet. This parameter stores the value passed into your list.
• A screen UI parameter mapping to associate the parameter in your list with the parameter in your data item.

Procedure
1. Navigate to All > System Applications > Studio to open Studio.
2. Select a scoped application where you want to create your applet.
3. In Application Explorer, navigate to **Mobile Studio > Applets** and select **Applets**.

4. Click the pop-out icon (≡) that appears to the right of **Applets**.

5. In the applet list, click **Create an applet**. The **Create an Applet** window appears on the screen.

6. In the **Create an Applet** window. Enter a name and description for your list applet.

7. Click the image next to the **Icon** field to select a color and image for your list applet icon. If you add your list applet to an applet launcher, this icon displays for the applet.

8. To prevent the list from displaying a list filter, enable **Hide filter**.

9. In the **Choose a template** field, select **List**.

10. **Optional**: To the right of the **Choose a template**, enable the check box next to **Form** to include a form screen in your applet. This form displays record information when your users tap a record on the list.

11. Click **Create New**. The **Create an Applet** window closes, and a new **Studio** tab opens for your list applet. If you selected the form option in the previous steps, you can also see a tab to configure your form.
12. Click the plus icon (✚) to the right of the **Data Item** field. A new Studio tab opens to configure a new data item.

13. On the new data item form, fill in the fields.

**New Data Item form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the data item.</td>
</tr>
<tr>
<td>Table</td>
<td>Table where the data item gets its data. This table should be the same one as the records on your chart.</td>
</tr>
<tr>
<td>Description</td>
<td>Description used to identify your data item.</td>
</tr>
<tr>
<td>Condition Type</td>
<td>Whether the conditions for the data item are declarative, scripted, or use an encoded query. In this case, select <strong>URL</strong>.</td>
</tr>
</tbody>
</table>

14. Click **Save**.

15. In the **Parameter Definition** section, click the plus icon (✚) to create a new parameter.
16. In the **Parameter Definition** pop-up, enter a name for your parameter in the **Name** field.

17. Click **Save**.

18. Next to the **Append Encoded Query** field, click the reference value icon (🔍).

19. Select the

20. Click **Save**.

You have a parametrized data item to use in your parametrized list. The condition type is set to **Encoded Query**, which in this case, is a URL that the data item receives from the chart applet. You configure this in the next steps.

21. In Studio, navigate back to the tab containing your list applet.

22. In the **Data Item** field, select the parametrized data item you created in the previous steps.

23. Click **Save**.

The list record is saved. Because the record is saved using a parametrized data item, the field you need to configure parameters are added to the form.

24. In the **Autofill Parameter Definition** section, click the plus icon (➕) to create a new parameter.

25. In the **Name** field, enter a name for your parameter. For example **URL**.

26. In the **Input Type** field, select **Instance relative URL**.

27. Click **Save**.

28. In the **Screen UI Parameter Mapping** section, click the plus icon (➕).

29. In the **Item Parameter** field, select the parameter from the data item. Since this list is drawn from that data item, there should be only one option.

30. In the **UI Parameter field** select the UI Parameter you created in the preceding steps. Since this list is drawn from that record, there should be only one option.

31. Click **Save**.

**What to do next**

Use the **Field Configurations** sections in the **List Screen** and **Form Screen** tabs to configure which fields you want to appear in your list and form screens. For more detail on these configurations, see **List applet configuration**.
Create a navigation function for your chart applet

Create a navigation function to direct your users from the chart applet to the parametrized list.

Before you begin
Role required: admin

Procedure

1. In Studio, navigate to Mobile Studio > Functions > Navigations.
2. Click the pop-out icon ( ).
3. In the Navigations list, click Create New.
4. In the New Navigation form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for your navigation function</td>
</tr>
<tr>
<td>Destination Type</td>
<td>Whether the function will navigate to an applet or applet launcher. In this case, select <strong>Applet</strong></td>
</tr>
<tr>
<td>Description</td>
<td>Description for your navigation function</td>
</tr>
<tr>
<td>Destination</td>
<td>Select the list you created in the previous steps</td>
</tr>
<tr>
<td>Context</td>
<td>Whether the function uses the global or record context. In this case, select <strong>Global</strong></td>
</tr>
</tbody>
</table>

5. Click Save.

Assign the navigation function to the chart applet

Assign your navigation function to your chart applet so your users can tap the chart to access the list of records.

Before you begin
Role required: admin

Procedure

1. In Studio, navigate to Mobile Studio > Applets.
2. Open your chart applet.
3. In the chart applet tab, click Functions.
4. Click the add button ( ) to add a function to your applet.
5. In the function field, select the navigation function you created in previous steps.
6. Click Save.
7. In the chart applet tab, click Save.
Input form screen

Use input form screens to provide an interface for your users to enter information in mobile apps.

**Note:** Beginning with the Rome release, some terminology has changed for certain ServiceNow® mobile products. This topic content references Mobile Studio, which retains the terminology used in previous releases. For more information, see ServiceNow mobile terminology changes.

Input form screens display inputs to allow your users to quickly enter information into mobile apps. Use input form screens to create or edit records, complete surveys, or any other situation where your users need to enter information. Input form screens work in offline mode, so users don't need an internet connection to input information into the app.

Once a input form screen is created and its inputs and variables defined, the input form screen must be associated with an action item and these inputs and variable must be mapped.

You can control how users input data, as well as the appearance of input types. You can further customize your input form screens using Mobile UI Rules.

Input form screen components

Input form screens consist of a header, and input section, and an option footer section for screens with multiple pages.

**Header**

The header of your input form screen displays the title of your input form screen, as well as **Cancel** and **Submit** buttons. You can change the labels of these buttons when you create your input form screen.

**Input section**

Below the header are your inputs. When building your input form screen, you can choose from several input types to give your users an easy intuitive experience for inputting information.

- Attachment
- Boolean
- Choice
- Date/time
- Number
• Ranking
• Reference
• String

For more detail on these parameter input types, see Input form screen attributes for inputs

Pagination buttons

Pagination buttons display at the bottom of the screen if you configure your input form screen to use multiple pages. Your users can tap the Previous and Next buttons at the bottom of the screen to navigate between pages. As with the header buttons, you can change the label of these buttons when you create your input form screen.

Input form screen limitations

Unsupported features

The following are unsupported input form screen input types.

• E-Signature
• Carried parameters
• Barcode/QR scanner

Configure an input form screen

Configure an input parameter so that your users can see a list of records from their mobile app. To access a record from the list, a user just has to tap the record name.

Before you begin
Role required: admin

Procedure

1. Navigate All > sys_sg_parameter_screen.list.
2. In the Input form Screens list, click New.
3. In the Input form Screen form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Title for the input form screen</td>
</tr>
<tr>
<td>Resume on Cancel</td>
<td>Whether the resume on cancel option is enabled. When enabled, data entered on the screen is saved in memory when a user clicks Cancel. The data is loaded when the user returns to the screen later.</td>
</tr>
<tr>
<td>Available offline</td>
<td>Whether the input form screen is available in offline mode. For details on offline mode, see Offline mode.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the input form screen</td>
</tr>
<tr>
<td>Field</td>
<td>Value</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Active</td>
<td>Whether the input form screen is active. Inactive screens do not appear in your mobile apps.</td>
</tr>
<tr>
<td>Fetch Type</td>
<td>The fetch type for the input form screen. For details on mobile fetch types, see <a href="#">Mobile fetch types</a>.</td>
</tr>
<tr>
<td>Header Card</td>
<td>An optional mobile view to display at the top of your input form screen. For more detail on view configuration, see <a href="#">Understanding mobile cards and legacy cards</a>.</td>
</tr>
<tr>
<td>Header Only on First Page</td>
<td>Whether the header is visible only on the first page of the input form screen.</td>
</tr>
</tbody>
</table>
| Advanced                 | Option to use pagination and section capabilities within an input form screen. Select the check box to display the Sections and Pages tabs.  
  • If checked, all Inputs that are part of input form screen are placed on a single page, if no pages are defined.  
  • If clear, all Inputs that are part of input form screen are paginated so that each input displays in its own page. |

4. Select **Submit**

**Configure input form screen buttons**

Configure labels for the buttons used to submit, cancel, and control pagination on your input form screen.

**Before you begin**

Role required: admin

**Procedure**

1. Navigate **All > sys_sg_parameter_screen.list**.
2. Select an input form screen.
3. Select the **Buttons** tab to display the button label fields.
4. Update the values in the buttons fields as needed.

**Input form screen button fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next Label</td>
<td>Label for the button to advance to the next page of the input form screen.</td>
</tr>
<tr>
<td>Cancel Label</td>
<td>Label for the button to cancel the input form screen.</td>
</tr>
</tbody>
</table>
Configure input form screen inputs

Configure the input fields your users will use to enter information. You can use screen inputs to determine the input type, question text, placeholder values, and other attributes.

Before you begin
Role required: admin

Procedure
1. Click the Inputs tab to display the button input fields.
2. Click Insert a new row... on the Inputs list to begin creating an input.
3. Enter a name for your input, then click the green check icon (✔) to save.
4. Right-click the form header and click Save from the context menu to save the input form screen record.
5. Click the name of your input to open the input record.
6. Fill in the fields as needed.

Input fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the input</td>
</tr>
<tr>
<td>Label</td>
<td>Label displays for the input</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the input. Your users will see this value below the input label.</td>
</tr>
</tbody>
</table>
| Input type | The type of input  
• Attachment  
• Boolean  
• Choice  
• Date/time  
• Number  
• Ranking  
• Reference  
• String |
| Placeholder| Text that appears in an input field before the user enters a value.     |
| Order      | Order in which the input appears in the input form screen.             |
| Mandatory  | Whether the input is mandatory                                        |
### Field Values

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Readonly</td>
<td>Whether the input is read-only</td>
</tr>
<tr>
<td>Autofill Variable</td>
<td>The autofill variable used for this input. For details on variables, Configure input form screen variables and attributes.</td>
</tr>
<tr>
<td>Active</td>
<td>Whether the input is active. Inactive inputs do not appear on the input form screen.</td>
</tr>
</tbody>
</table>

7. When finished with configuring your fields, click **Update**.

### Input Form Screen Attributes for Inputs

Learn about the inputs available for use in input form screens and the attributes used to configure them.

Use input attributes to define the appearance and set limitations on your inputs. Examples include setting minimum and maximum values, or displaying an input value as a percentage.

⚠️ **Note:** All attributes are case-sensitive.

### General Attributes

You can use these attributes with any input form screen, regardless of the input type.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DescriptionRenderType</td>
<td>Define a render type for the input description.</td>
</tr>
<tr>
<td></td>
<td><strong>html</strong></td>
</tr>
<tr>
<td></td>
<td>Allows HTML tags to format the text in the <strong>Description</strong> field of your input.</td>
</tr>
</tbody>
</table>

### Attachment Inputs

Use attachment inputs to allow users to attach files in an input form screen. Attachment inputs require a scripted action item and an actionResult object to upload a file.

⚠️ **Note:** Attachment inputs are only supported for scripted actions.

Below is an example code containing attachment inputs.

```javascript
(function WriteBackAction(parm_input, parm_variable, actionResult) {
  //other write-back inputs and variables here
}
```
**Boolean inputs**

Use boolean inputs for inputs with true or false values. You can configure boolean inputs to appear either as a check box or as a toggle switch.

![Boolean Input Examples](image)

You can use these attributes with inputs with the boolean input type.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RenderType</td>
<td>Specify a render type for the boolean input</td>
</tr>
<tr>
<td><strong>Checkbox</strong></td>
<td>Displays the input as a checkbox.</td>
</tr>
<tr>
<td><strong>toggle</strong></td>
<td>Displays the input as a toggle.</td>
</tr>
</tbody>
</table>

**Choice inputs**

Use choice inputs to give your users options to select from. You can create these choices manually or use an existing table and field to provide the choices.

Choice inputs can appear as text or as images.

**Note:** Choice field dependencies are not supported.

You can configure a choice input to allow users to select a single choice, or multiple choices.

You can use these attributes with inputs with the choice input type.
## Attribute Description

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MultiSelect</strong></td>
<td>Whether users can select multiple choices for the input. The value must be <strong>true</strong> or <strong>false</strong></td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>The table where the choice field is located. <strong>Note:</strong> Use this option to use the choices in an existing table and field rather than manually creating choices. You must select a field on this table using the <strong>Field</strong> attribute.</td>
</tr>
<tr>
<td><strong>Field</strong></td>
<td>The field that contains your choices. <strong>Note:</strong> Use this option to use the choices in an existing table and field rather than manually creating choices. The field defined in this attribute must be on the table defined in the <strong>Table</strong> attribute.</td>
</tr>
</tbody>
</table>

### Date/time inputs

Use date/time inputs to allow your users to input date and time values. You can configure the input to allow date and time, or date only.

You can use these attributes with inputs with the date/time input type.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RenderType</strong></td>
<td>Specify a render type for the date/time input</td>
</tr>
</tbody>
</table>

- **date-time**
  - Displays date and time. If no RenderType is defined, this option is the default.

- **date**
  - Displays only the date.
Number inputs

Use number inputs to allow your users to enter numerical values. Define optional minimum and maximum values to define a number range for your input.

You can optionally configure a number input to appear as a percentage.

By default, the Number input supports decimals.

You can use these attributes with inputs with the number input type.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
</table>
| RenderType | Specify a render type for the number input  
| percentage | Displays a percent symbol(%) after the number in the input.  
| line | Displays the input as a line field.  
| box | Displays the input as a box field.  
| integer | Displays whole integers, such as 3, 1003, or 57. When integer is used, end users can only enter integers and can't enter decimals.  
| Max | Specify a maximum value for the input  
| Min | Specify a minimum value for the input |
Ranking inputs

Use ranking inputs to allow your users to sort the available choices in order of preference or priority. Users tap and drag the available choices.

Choices for the ranking input must be manually defined.

There are no attributes for ranking input type other than the general attributes list later in this document.

Reference inputs

Use reference inputs for inputs that reference a field on a table. These inputs work like reference fields in the forms on your instance. You can configure your reference input with conditions, reference qualifiers, and a search option to help your users find what they need quickly.

The optional search option uses "For text" keyword searches, and can search for elements within the mobile view.

You can use these attributes with inputs with the reference input type.

Required attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SourceTable</td>
<td>The source table for your reference qualifier.</td>
</tr>
<tr>
<td>SourceFieldName</td>
<td>Sys_id of input form screen input of dependent value.</td>
</tr>
</tbody>
</table>
Optional attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MultiSelect</td>
<td>Whether users can select multiple choices for the input. The value must be true or false</td>
</tr>
<tr>
<td>MobileViewld</td>
<td>Mobile view ID (View config) to use for reference list items</td>
</tr>
<tr>
<td>EnableSearch</td>
<td>Whether the search bar displays. The value must be true or false.</td>
</tr>
<tr>
<td>SearchPlaceholder</td>
<td>Text that appears in a search bar as a placeholder.</td>
</tr>
<tr>
<td>Conditions</td>
<td>Enter an encoded query to apply conditions to your reference list. For details on creating these queries, see Encoded query strings.</td>
</tr>
</tbody>
</table>

This example creates an input form screen input for the Assigned to field on the Incident [incident] table. To create this input you use the three required attributes, **TargetTable**, **SourceTable**, and **SourceFieldName**.

1. Your users select from a list of users to assign the incident to. So, you set the **TargetTable** attribute to sys_user.

2. Because we are adding our user to an incident record, the **SourceTable** attribute is set to incident.

3. Finally, we set the **SourceFieldName** to assigned_to to that the user selected will be assigned to the Assigned to field.
String inputs

Use string inputs for text input. You can define a maximum character length for this input.

You can use these attributes with inputs with the string input type.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inline</td>
<td>Whether the input is shown inline, or opens in a separate modal. The value must be <code>true</code> or <code>false</code>.</td>
</tr>
<tr>
<td>MaxLength</td>
<td>Specify a maximum number of characters for the input value.</td>
</tr>
</tbody>
</table>
Configure input form screen variables and attributes

Use screen variables to collect information from the user automatically or define default information. Variables can include information like user IDs and GPS coordinates.

Before you begin
Role required: admin

Procedure
1. Navigate to All > sys_sg_parameter_screen.list.
2. Select an input form screen.
3. Select the Variables tab to display the input fields.
4. Select Insert a new row... on the Name column to begin creating a variable.
5. Enter a name for your variable, then click the green check icon (✓) to save.
6. To the right of your variable name, double-click the space under Variable Type.
7. Select the type for your variable, then click the green check icon (✓) to save.

Important: If the variable type you select uses an attribute to set its value, you must save the input form in step 8. Then you can set the attribute value in step 9.

<table>
<thead>
<tr>
<th>Variable type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database field</td>
<td>Data from a field. Uses the FieldName attribute.</td>
</tr>
<tr>
<td>GPS Coordinates</td>
<td>Longitude and latitude of the user.</td>
</tr>
<tr>
<td>User</td>
<td>Sys_id of the user.</td>
</tr>
<tr>
<td>Date</td>
<td>Current date</td>
</tr>
<tr>
<td>Constant</td>
<td>A value set by the administrator using the ConstantValue attribute.</td>
</tr>
<tr>
<td>Offline Mode</td>
<td>Value that indicates whether an action is done in offline mode.</td>
</tr>
<tr>
<td>ParentContext</td>
<td>Context information that is carried from a parent record into an action. Uses the ContextField attribute.</td>
</tr>
<tr>
<td>Scripted</td>
<td>Script that auto-fills inputs. For example, you can pre-fill a building number based on an end user's profile when making a reservation. Uses the Script attribute.</td>
</tr>
</tbody>
</table>

8. Right-click the form header and select Save from the context menu.
9. Optional: Click the name of your variable to access the variable record.
10. In the Variable Attributes list, click Insert a new row... to begin creating a variable attribute.

**Variable attributes**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FieldName</td>
<td>Name of a field on the table. Use this attribute if you have selected the Database field variable type.</td>
</tr>
<tr>
<td>ConstantValue</td>
<td>Static data defined by the administrator. Use this attribute if you have selected the Constant variable type.</td>
</tr>
<tr>
<td>Script</td>
<td>Code that is used to execute a script that pre-fills an input in the form. Use this attribute if you have selected the Scripted variable type.</td>
</tr>
<tr>
<td>ContextField</td>
<td>Field value of the parent record that you want to carry into an action. Use this attribute if you have selected the ParentContext variable type.</td>
</tr>
</tbody>
</table>

11. Enter a name for your variable attribute, then click the green check icon (✓) to save.

12. To the right of your variable attribute name, double-click the space under Value.

13. Enter a value for your variable attribute, then click the green check icon (✓) to save.

14. When you are finished creating variable attributes, click Update to save the variable record.

15. Return to the record for your input form screen.

16. In the Inputs list, update an input to use your variable by selecting it in the Autofill Variable field.

**What to do next**

After you have created your input form screen variables, you can map these variables to input form screen inputs or action items. For details on this process, see Configure an action item.

**Configure input form pages**

Create input form pages to enable users to view data over multiple organized pages, rather than scrolling through a long list of entries. Input form pages host data from inputs and sections. You control the layout of the pages, and which inputs and sections belong to each page of the input form screen.
Before you begin
Role required: admin

Procedure
1. Navigate All > sys_sg_parameter_screen.list.
2. Select an input form screen.
3. In the input form screen, select Advanced. The Pages and Sections tabs display.
4. Select the Pages tab to display the input form pages associated with a input form screen.
5. Select Insert a new row... on the Input Form Pages list to begin creating a page.
6. Select Name and define a name for your input form page, then click the green check icon (✔) to save.
7. Select Order and enter a number to define the placement order of the page, as part of the input form page. Select the green check icon to save.

   Note: An input form page displays variables and sections based according to the number defined in their respective Order fields. Sections and variables do not have priority over each other.
8. Select Active and select true to display the page or select false to not display the page. Select the green check icon to save.
9. Right-click the form header and click Save.
10. Map inputs or sections to designated pages.

<table>
<thead>
<tr>
<th>Option</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Map an input to a designated page</td>
<td>a. Select the Inputs tab.</td>
</tr>
<tr>
<td></td>
<td>b. Select the input name to map to the page.</td>
</tr>
<tr>
<td></td>
<td>c. In the Page field in the Input screen, select the reference lookup icon (QRST).</td>
</tr>
<tr>
<td></td>
<td>d. From the Input form Pages form, select the page to contain the input.</td>
</tr>
<tr>
<td></td>
<td>e. Right-click the form header and click Save.</td>
</tr>
<tr>
<td>Map a section to a designated page</td>
<td>a. Select the Sections tab.</td>
</tr>
<tr>
<td></td>
<td>b. Select the section to map to the page.</td>
</tr>
<tr>
<td></td>
<td>c. In the Page field in the Input screen, select the reference lookup icon (QRST).</td>
</tr>
<tr>
<td></td>
<td>d. From the Input form Pages form, select the page to contain the input.</td>
</tr>
<tr>
<td></td>
<td>e. Right-click the form header and click Save.</td>
</tr>
</tbody>
</table>

Results
An input form screen can contain any number of pages. Each page can comprise inputs and reusable sections.
Configure input form sections

Input form sections are reusable components consisting of grouped inputs. You can associate these sections with any number of input form pages.

Before you begin
Role required: admin

Procedure

1. Navigate All > sys_sg_parameter_screen.list.
2. Select an input form screen.
3. In the input form screen, select Advanced.
   The Pages and Sections tabs display.
4. Select the Sections tab to display the input form sections mapped with the listed input form.
5. Select Insert a new row... on the Input Form Sections list.
6. Select **Name** and define a name to reference a section with the input form.

7. Select the green check icon (✓) to save the name.

8. Select **Label** and define a display name for your input form sections, then click the green check icon to save.

9. In the **Input Form Sections**, select the reference lookup icon (🔍) and either create or select a input form section.

<table>
<thead>
<tr>
<th>Option</th>
<th>Procedure</th>
</tr>
</thead>
</table>
| **Select an existing input form section** | a. Select a preconfigured input form section from the list.  
   b. Select the green check icon to save. |
| **Create a new input form section**  | a. Select **New** in the **Input Form Sections**.  
   b. Enter a **Name** for the input form section.  
   c. Select the **Inputs** tab and add input form inputs. For more information, see **Configure input form screen inputs**.  
   d. Select the **Variables** tab and add input form variables. For more information, see **Configure input form screen variables and attributes**.  
   e. Select **Submit**.  
   f. Select the green check icon to save. |

10. Select **Page** and determine which page your input form section displays. Select the green check icon to save.

11. Select **Order** and enter a number to define the placement order of this section within the specified page. Select the green check icon to save.

   **Note:** An input form page displays variables and sections based according to the number defined in their respective **Order** fields. Sections and variables do not have priority over each other.

12. Select **Active** and select **true** to display the section or select **false** to not display the section.

13. Right-click the form header and click **Save** to save the input form record.

**Results**

Your input form can contain any number of inputs and reusable sections within any number of pages.
Mobile fetch types

Fetch type settings determine when data is loaded in your applets. Change your fetch type to optimize load time performance for your applets.

### Mobile fetch types

<table>
<thead>
<tr>
<th>Fetch type</th>
<th>Description</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefetch</td>
<td>Prefetch is the default fetch type for new applets, except form and details. This option pre-loads form applet data while when your user accesses a list, calendar, or form applet. Lists may take longer to load, but form load time is faster.</td>
<td>This fetch type is the default for most applets. Use this fetch type when the form and form segments do not take much additional time to load.</td>
</tr>
</tbody>
</table>
## Mobile fetch types (continued)

<table>
<thead>
<tr>
<th>Fetch type</th>
<th>Description</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background</td>
<td>The app makes a background network request to load embedded screens or form segments. Embedded screens and form segments load instantly once the background request completes.</td>
<td>Use this fetch type when an applet is not the first loaded, but one your users are likely to navigate to. For example, a related list associated with a form.</td>
</tr>
<tr>
<td>On Demand</td>
<td>The app makes a network request to load the app only when your users navigate to it.</td>
<td>Use this fetch type when a screen is not expected to be used often.</td>
</tr>
<tr>
<td>Dynamic</td>
<td>The screens for the first 10 rows load as described in the <code>prefetch</code> type. After 10 first rows, the app loads screens as defined in the <code>on demand</code> fetch type. This option is the default for form and details screens. You can change the number of rows loaded with <code>prefetch</code> by changing the value of the <code>Dynamic prefetch count</code> field.</td>
<td>Use this fetch type when large lists load too slowly using the <code>prefetch</code> fetch type.</td>
</tr>
</tbody>
</table>

### Change the fetch type for an applet
Change the fetch type of an applet to change when the app loads its data.

**Before you begin**
Role required: admin

**About this task**
Applet fetch types typically do not need to be changed. Be sure to test fetch type changes thoroughly before using in a production environment, as fetch type changes can have a significant affect on performance.

**Procedure**
1. Navigate to All > System Mobile > Applets.
2. Open an applet.
3. In the **Fetch Type** field, select a fetch type.
   - If you are in the global scope, the record appear read-only. In this case, you can temporarily switch to the appropriate scope by clicking on the banner notification that appears at the top of the record.
4. Click **Update**.
Enabling bookmarking screens and web pages

Configure the saved views feature to allow users to bookmark screens and web pages. Users can instantly access these saved views via an icon in the navigation bar.

The saved views feature is automatically enabled in the Mobile Agent app. However, you must manually configure and enable it on the Now Mobile app and Mobile Onboarding.

Note:
The Mobile Onboarding app is being deprecated!
With the Now Platform Rome release in September 2021, we started phasing out support for the Mobile Onboarding. Customers may no longer activate it, and we are not offering enhancements or non-critical bug fixes. Mobile onboarding features are available in the Now Mobile app for HR Service Delivery.

You can choose to disable access to this feature on individual screens.

Enable saved views
Make the ability to save views available for specified mobile apps.

Before you begin
Role required: admin

About this task
By default, the saved views feature is enabled in the Mobile Agent. You have to manually enable and configure the feature for the Now Mobile and Mobile Onboarding apps.

Note:
The Mobile Onboarding app is being deprecated!
With the Now Platform Rome release in September 2021, we started phasing out support for the Mobile Onboarding. Customers may no longer activate it, and we are not offering enhancements or non-critical bug fixes. Mobile onboarding features are available in the Now Mobile app for HR Service Delivery.

Procedure
1. Navigate to All > System Mobile > Mobile Branding > Native clients.
2. Select the mobile client on which you want to include the saved views feature.
3. Select the information icon (𝒊) in the Navigation field of the Native Client page. Select Open Record.
4. In the Navigation Tabs section, select New.
5. Select the reference lookup icon (🔍) in the Navigation Tab field.
6. Select the Saved navigation tab.
7. Select the menu icon (≡) and select Save.
8. Select the information icon (𝒊) in the Navigation Tab field. Select Open Record.
9. Make sure that the Active option is selected.
10. Select Update in the Saved Tab form.
11. Select Update in the Navigation Tab Map form.

© 2022 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.
Disable saved views

Disable the saved views feature for some or all of your mobile apps.

Before you begin
Role required: admin

About this task
For privacy, security, or other reasons you might decide not to enable the saved views functionality. Use this same procedure on the Mobile Agent, Now Mobile app, and Mobile Onboarding.

Note:
The Mobile Onboarding app is being deprecated!
With the Now Platform Rome release in September 2021, we started phasing out support for the Mobile Onboarding. Customers may no longer activate it, and we are not offering enhancements or non-critical bug fixes. Mobile onboarding features are available in the Now Mobile app for HR Service Delivery."

Procedure
1. Navigate to All > System Mobile > Mobile Branding > Native clients.
2. Select the native client on which you want to disable or not include the saved views feature.
3. Select the information icon (i) in the Navigation field of the Native Client page. Select Open Record.
4. Select Saved in the Navigation Tab column of the Navigation page.
5. Clear the Active option.
6. Select Update.

Make saved views unavailable on specific pages
You might want to remove the option for users to save specific views and screens for security or privacy reasons.

Before you begin
Role required: admin

Procedure
1. In the web-based UI, enter sys_sg_favorite_deny_list.list in the filter navigator to open the list of global search configurations.
2. Select New in the Mobile Favorites Deny List form.
3. In the Source Table field, enter applet.
4. Select the reference lookup icon (i) from the Source field.
5. In the Select the document screen, within the Document field, select the reference lookup icon (i), then select the screen on which you want the saved views icon not to display.
6. Select OK.
7. Make sure that the Active option is selected.
8. Select Submit.
**Results**
The specified screen no longer displays the saved views icon in the screen header.

**Mobile styles**
Use mobile style elements to change the appearance of your mobile applications.

<table>
<thead>
<tr>
<th>Legacy cards</th>
<th>![Diagram of legacy card elements]</th>
</tr>
</thead>
</table>
| Use legacy cards to show a summary of a record in an easy to read format. Legacy cards are the patterns that define the appearance and location of fields in an applet. Legacy cards are specific to a screen, each screen has its own unique legacy card record. | **Number**  
**Description**  
**Assigned to**  
**Caller**  
**Priority**  
**Avatar** |

Learn more about legacy cards at [Mobile hierarchy](#).

<table>
<thead>
<tr>
<th>UI styles</th>
<th>![Sample of UI styled cards]</th>
</tr>
</thead>
</table>
| Use UI styles to dynamically change legacy card elements such as background color or font color. You can use UI styles to show or hide legacy card elements, and add icons next to an element within an legacy card. | **Closed**  
Unable to connect to email  
3 - Moderate  
False  
**Closed**  
Need Oracle 10GR2 Installed  
4 - Low  
False  
**Closed**  
EMAIL is slow when an attachment is i...  
1 - Critical  
False  
**Closed**  
I can’t launch my VPN client since the I...  
1 - Critical  
True  
**Closed**  
How do I create a sub-folder  
1 - Critical  
True  |

For more detail on UI styles, see [Mobile styles](#).
Mobile UI Rules

Use Mobile UI rules to build rules that modify your mobile UI screens. These rules include options so hide fields, perform field value calculations, and apply UI styling to different UI elements.

In the example to the right, a UI style hides the Impact field based on conditions defined by an administrator.

For more information on mobile UI rules, see Mobile UI Rules.

Icons

Use icons to provide unique visual identifiers for navigation tabs and quick actions in your mobile applications.

To learn more about mobile icons, see Mobile icons.

Note: Custom mobile icons are not supported.

Mobile UI Rules

Use Mobile UI rules to build rules that modify the mobile UI. These modifications can include hiding and displaying fields, making fields mandatory, performing field value calculations, and applying UI styles to different UI elements.

Mobile UI rules and actions

To modify the appearance of your mobile views and input form screens UI, you must create a mobile UI rule, and at least one associated mobile UI action.

Mobile UI rules

A mobile UI rule applies to a single mobile views and input form screen. When you create a rule, you select a record to apply the rule to, as well as conditions that define when your rule applies.

Mobile UI rule actions

Each mobile UI rule uses mobile UI rule actions. These actions define the changes made to your target record. When you create an action, you select an operation, which defines the kind of change you want to make. You then define a value, which gives the rule more detail on the specifics of the change you're making.

For more detail on the kinds of changes you can make with mobile UI rules, see the next section. For detailed steps on creating mobile UI rules and actions, see Create a mobile UI rule.
Available mobile UI rule operations
Allow calculations on input form screen inputs

Use calculations to quickly and accurately fill in field values based on calculations you define. These calculations can include values from other fields. These rules can save time and reduce complexity on your mobile screens.

ℹ️ Note: Calculations can only be applied to input form screens.

Apply UI styling to mobile view elements
Use mobile UI rules to apply UI styling to your screens. Using JSON code, you can set a background color, font color, and text decorator icons.

ℹ️ **Note:** UI styling is only applicable for mobile views.

---

**Disable buttons**

Use mobile UI rules to conditionally disable buttons. De-clutter your interface and avoid mistakes by ensuring buttons are only available in situations where your users need them.

ℹ️ **Note:** The disable buttons option is only applicable for mobile views.

---

**Hide or display inputs or mobile view elements**
Use mobile UI rules to control the visibility of your inputs or elements on your mobile views. Present a clean UI and avoid confusion by displaying inputs only when they are relevant to your users.

Make inputs read-only or mandatory
Make your inputs read-only or mandatory to control which inputs are required and which are uneditable. With these rules, you can ensure that required information is added and prevent unintentional editing of fields that do not require updates.

ℹ️ Note: The option to make inputs read only or mandatory can only be applied to input form screens.

Create a mobile UI rule

Create a mobile UI rule to apply styling or field calculations for your mobile UI elements

**Before you begin**

Role required: admin

ℹ️ Note: This configuration takes place in the native UI outside Studio

**Procedure**

1. Navigate to *All > System Mobile > Mobile UI > Mobile*.
2. In the *Mobile UI Rules* list, click *New*.
3. In the *Mobile UI Rule* form, fill in the fields.

**Mobile UI Rule form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the mobile UI rule</td>
</tr>
<tr>
<td>Application</td>
<td>Application for the UI rule. This field automatically populates with the current application.</td>
</tr>
<tr>
<td>Active</td>
<td>Whether the mobile UI rule is active.</td>
</tr>
<tr>
<td>Field</td>
<td>Value</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Condition</td>
<td>Condition under which the mobile UI rule is used. The rule executes if the condition evaluates as true. Mobile UI rules with no condition always evaluate as true.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> For details on the operators available for your condition, see Operators available for filters and queries.</td>
</tr>
<tr>
<td>Parent Table</td>
<td>The parent table the mobile UI rule applies to. Select either: • Input Form Screen • Mobile View</td>
</tr>
<tr>
<td>Parent</td>
<td>The input form screen or mobile view record to which your mobile UI rule is applied.</td>
</tr>
<tr>
<td>Triggers</td>
<td>When the mobile UI rule is triggered. Select one or both of the following options.                                                       • onLoad: The mobile UI rule triggers when the record loads. • onChange: The mobile UI rule triggers when the record is changed.</td>
</tr>
<tr>
<td>Order</td>
<td>Order in which the mobile UI rule executes. If there are multiple rules affecting the same element, the rules execute according to this value. Lower numbered rules execute before higher numbered rules</td>
</tr>
<tr>
<td>Reverse If False</td>
<td>When this field is selected, the mobile UI rule actions are reverse if the condition evaluates as false.</td>
</tr>
</tbody>
</table>

4. Click **Submit**

**What to do next**
After creating your mobile UI rule, you must create mobile UI rule actions to execute when your rule is used.

**Create a mobile UI rule action**
Create the actions that execute when your mobile UI rule conditions are met.

**Before you begin**
Role required: admin

**Note:** This configuration takes place in the web-based UI outside Studio

**Procedure**
1. Navigate to **All > System Mobile > Mobile UI > Mobile.**
2. Open the mobile UI rule where you want to add actions.
3. In the mobile UI rule form, click **New** in the **Mobile UI rule actions** related list.
4. In the mobile UI rule action form, fill in the fields.

## Mobile UI rule action form

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Rule</td>
<td>The UI rule associated with this action. This field automatically populates with the rule you opened in step 2.</td>
</tr>
<tr>
<td>Application</td>
<td>Application for the UI rule. This field automatically populates with the current application.</td>
</tr>
</tbody>
</table>
| Target    | The element affected by the action. The actions available to select depend on the parent selected in your mobile UI rule.  
- For input form screens, the targets are input form screen inputs  
- For mobile views, targets are elements from the mobile card template JSON. |
| Operation | Select the operation the action performs  
See the table below for details on these actions. |
| Order     | Order in which the action executes. If there are multiple actions for the same mobile UI rule, the actions execute according to this value. Lower numbered rules execute before higher numbered rules |
| Value     | Defines how the action applies to the element in the Target field. This value depends on the selected Operation. See the table below for operation descriptions and the values they use. |

## Mobile UI rule action operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Definition</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable Button</td>
<td>Determines whether the button in the Target field is disabled.</td>
<td>Set the Value to true to disable a button or false to enable it.</td>
</tr>
<tr>
<td>Set Mandatory</td>
<td>Determines if the element in the Target field is mandatory.</td>
<td>Set the Value to true to make the element mandatory or false to make it optional.</td>
</tr>
<tr>
<td>Set Read-only</td>
<td>Determines whether the element in the Target field is read-only.</td>
<td>Set the Value to true to make an element read-only, or false to make the element editable.</td>
</tr>
<tr>
<td>Set UI Style</td>
<td>Applies UI styling to an element.</td>
<td>You can use this value to set a background color, font color, and text decorator</td>
</tr>
<tr>
<td>Operation</td>
<td>Definition</td>
<td>Values</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td></td>
<td>This operation does not reverse when using <strong>Reverse If False</strong> on your mobile UI rule.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Values</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>icon. The value must be in JSON format. This example changes the colors of the background and text:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>`{</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;BackgroundColor&quot;: &quot;#B5EBD4&quot;,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;FontColor&quot;: &quot;#1E4335&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This example sets a text decorator icon.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>`{</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;TextDecorator&quot;:{</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;Type&quot;: &quot;Font&quot;,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;FontName&quot;: &quot;now-mobile-icons-cards&quot;,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;Value&quot;: &quot;IconCode&quot;,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;FontColor&quot;: &quot;#1E4335&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>}</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td></td>
<td>IconCode must be replaced with a valid icon code. For a list of icon codes, see Mobile icons</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td></td>
<td>Use a JSON validator when copying and pasting JSON code from a topic, to make sure all code is correctly applied.</td>
</tr>
<tr>
<td>Set Visibility</td>
<td>Determines the visibility of the element in the <strong>Target field</strong> is read-only.</td>
<td>Set the <strong>Value</strong> to true to display the element, or false to hide the element.</td>
</tr>
<tr>
<td>Simple Arithmetic</td>
<td><strong>Note:</strong> This operation does not reverse when using <strong>Reverse If False</strong> on your mobile UI rule.</td>
<td>Set the <strong>Value</strong> by creating a mathematical operation using inputs from the input form screen associate with your mobile UI rule. For example, if you wanted to calculate travel costs you could use:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>`{</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cost_per_mile*miles_travel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In this example, the values in the cost_per_mile and</td>
</tr>
<tr>
<td>Operation</td>
<td>Definition</td>
<td>Values</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>miles_traveled</td>
<td>inputs in the would be multiplied. For this operation to work, the input form screen would need inputs matching these names.</td>
<td></td>
</tr>
<tr>
<td>Time ago</td>
<td>Applies a time indicator to a record, for example &quot;Updated 3 minutes ago&quot; or &quot;Created 2 months ago&quot;.</td>
<td>The following fields must be set:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• In the <strong>Operation</strong> field, enter Date Formatter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• In the <strong>Target</strong> field, set the target to a mobile view element that uses a DateTime input.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• In the <strong>Value</strong> field, enter timeago.</td>
</tr>
<tr>
<td>Note: This operation does not reverse when using <strong>Reverse If False</strong> on your mobile UI rule.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time and date calculation</td>
<td>Automatically calculates an end time after entering a start time. The end time value can be in the past or the future. For example, you select a meeting to start at 15:00, the default calculation is set to 30 minutes later, so 15:30 displays as the default end time.</td>
<td>The following fields must be set:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• In the <strong>Operation</strong> field, enter Date Calculation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• In the <strong>Target</strong> field, set the target to a input form screen element that uses a DateTime input.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• In the <strong>Value</strong> field, enter the input structure using the following syntax &lt;input name&gt; +/- &lt;duration&gt;.</td>
</tr>
<tr>
<td>Note: This operation does not reverse when using <strong>Reverse If False</strong> on your mobile UI rule.</td>
<td></td>
<td>The duration value is formatted in the following way:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 10d = 10 days</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 5h = 5 hours</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 3m = 3 minutes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 2s = 12 seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For example, an entry of start_time + 1d8h sets the value to be calculated at 1 day and 8 hours after the start time.</td>
</tr>
<tr>
<td>Note: You cannot use month and year in the time and date calculation.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Mobile UI styles

UI styles define the font and background color of fields in your applets.

You can associate one or more UI styles to a legacy card record to change the appearance of the fields in that legacy card. Each UI style applies to a single table. You and only applies under conditions you set within the UI style record.

⚠️ Note: Mobile UI styles can only be applied to legacy cards.

Create UI Styles

Create and modify your UI Styles using the Mobile App Builder. In a UI style, you select a field from your selected table, and define a font and background color for that field. Use may include conditions to determine when a style is applied. For example, you may want to use a colored background on the Priority field to highlight when a record is high priority.

Each UI style record defines the colors for a single field, however multiple UI styles can be added to a legacy card to customize the appearance of multiple fields.

Learn more about creating UI styles at Create mobile UI styles.

Create mobile UI styles

Create UI styles to apply to fields in the mobile app.

Before you begin

Create an app and applets before configuring UI styles.

Role required: admin

Procedure

1. Navigate to All > System Applications > Studio.
2. From the Select Application window, select the application containing your app.
3. In the Application explorer on the left edge of the screen, click UI Styles, and then click the pop-out icon that appears to the right of the text.
4. Click New.
5. Complete the following fields as needed.

<table>
<thead>
<tr>
<th>UI Style fields</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A name for the UI style. Choose a name that is easily identifiable. You use this name to apply the style to a specific field in an applet.</td>
</tr>
<tr>
<td>Table</td>
<td>The name of the table containing the field you want to add a style to. Select the table matching the table for the applet where you intend to apply this style.</td>
</tr>
<tr>
<td>Condition</td>
<td>The 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</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Use item view elements</td>
<td>Enable to select an item view and use the elements from that record.</td>
</tr>
<tr>
<td>Item view</td>
<td>Select an item view. This field appears when the <strong>Use item view elements</strong> option is selected.</td>
</tr>
<tr>
<td>Item view element</td>
<td>This field appears when the <strong>Use item view elements</strong> option is selected. The field contains a list of the elements from the item view selected in the <strong>Item view</strong> field. These elements represent the JSON elements found in the selected <strong>Item view</strong> record.</td>
</tr>
<tr>
<td>Field name</td>
<td>The name of the field you want to add style to. Only fields available on the table you selected appear in this list.</td>
</tr>
<tr>
<td>Style</td>
<td>The style field supports multiple entries. Each entry has a <strong>Name</strong> and <strong>Value</strong> elements. Click the Add (➕) icon to add a new entry to the style field or the remove icon (➖) to remove existing elements. See the following table for a list of the available UI style names.</td>
</tr>
<tr>
<td>Active</td>
<td>Whether the UI style is active.</td>
</tr>
</tbody>
</table>

**Available UI styles**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>font_color</td>
<td>Use this style to change the text color for a field. The value for this style is a color in hexadecimal format.</td>
</tr>
<tr>
<td>background_color</td>
<td>Use this style to change the background color for a field. The value for this style is a color in hexadecimal format.</td>
</tr>
<tr>
<td>is_hidden</td>
<td>Use this style to hide or reveal a field. The value of this style must be <strong>true</strong> or <strong>false</strong>.</td>
</tr>
<tr>
<td>text_decorator_icon</td>
<td>Use this style to add an icon to the left edge of your field. The value of this style must be the Sys ID of a record on the Icon [sys_sg_icon] table.</td>
</tr>
</tbody>
</table>

6. Click **Submit**.

7. Repeat this process for each condition you want the UI style to appear for. For example, if you want the **State** field to have a different color for each value, create a UI style for each state value.
Example: Suggested UI styles
Listed here are suggested UI styles for your instance. These styles are WCAG compliant, and are created with high contrast backgrounds to make the labels easily readable for all users.

### High contrast label styles

<table>
<thead>
<tr>
<th>Label</th>
<th>Background color</th>
<th>Text color</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - Critical</td>
<td>Sweet Pink #FFA4A3</td>
<td>Cherrywood #64201A</td>
</tr>
<tr>
<td>2 - High</td>
<td>Apricot #FBD0B3</td>
<td>Jacko Bean #2F1F0A</td>
</tr>
<tr>
<td>3 - Moderate</td>
<td>Periwinkle #C9C9ED</td>
<td>Gulf Blue #38385B</td>
</tr>
<tr>
<td>4 - Low</td>
<td>Breeze #B5DDE5</td>
<td>Tiber #1A424B</td>
</tr>
<tr>
<td>5 - Planning</td>
<td>Zumthor #D1D6D8</td>
<td>Gable Green #293E40</td>
</tr>
<tr>
<td>Label</td>
<td>Drover #FBF2B4</td>
<td>Jacko Bean #2F1F0A</td>
</tr>
<tr>
<td>Label</td>
<td>Cruise #B5EBD4</td>
<td>Sherwood Green #1E4335</td>
</tr>
<tr>
<td>Label</td>
<td>Iceberg #CFDFD7</td>
<td>Sherwood Green #1E4335</td>
</tr>
<tr>
<td>Label</td>
<td>Marble #D0E6E6</td>
<td>Gable Green #293E40</td>
</tr>
<tr>
<td>Label</td>
<td>Seashell #F1F1F1</td>
<td>Gable Green #293E40</td>
</tr>
</tbody>
</table>

### Conditional formatting styles

<table>
<thead>
<tr>
<th>Label</th>
<th>Background color</th>
<th>Text color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample</td>
<td>Transparent #FFFFFF00</td>
<td>Mahogany #C83C36</td>
</tr>
<tr>
<td>Sample</td>
<td>Transparent #FFFFFF00</td>
<td>Tree Poppy #DC7C40</td>
</tr>
<tr>
<td>Sample</td>
<td>Transparent #FFFFFF00</td>
<td>Chetwode Blue #7070B3</td>
</tr>
<tr>
<td>Sample</td>
<td>Transparent #FFFFFF00</td>
<td>Fountain Blue #5297C4</td>
</tr>
<tr>
<td>Sample</td>
<td>Transparent #FFFFFF00</td>
<td>Oslo Gray #7D8A8C</td>
</tr>
<tr>
<td>Sample</td>
<td>Transparent #FFFFFF00</td>
<td>Earls Green #B09F2C</td>
</tr>
<tr>
<td>Sample</td>
<td>Transparent #FFFFFF00</td>
<td>Sea Green #54AC98</td>
</tr>
</tbody>
</table>
What to do next
Add the field style to the field in an applet.

Create text decorator icons
Use a text decorator icon to selectively highlight list elements that need your users attention.

Before you begin
Role required: admin

Procedure
1. On your instance, navigate to System Mobile > Mobile UI Styles.
2. In the UI Styles list, click the New button.
3. In the UI Style form, fill in the fields as needed.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of your UI style</td>
</tr>
<tr>
<td>Table</td>
<td>Table to which the UI style applies. Use the same table as the applet where you apply your style.</td>
</tr>
<tr>
<td>Condition</td>
<td>Condition under which the icon is visible. Leave the condition field blank to apply the icon to all records.</td>
</tr>
<tr>
<td>Use Item View elements</td>
<td>Whether the item view elements are used. Enable this checkbox to create a text decorator icon.</td>
</tr>
<tr>
<td>Item view</td>
<td>The item view where the icon appears. Use the same item view as the applet where you want to see your icon.</td>
</tr>
<tr>
<td>Item view element</td>
<td>The element of the item view where you want to see your icon.</td>
</tr>
<tr>
<td>Style</td>
<td>The style where you define your icon. For a text decorator icon, use text_decorator_icon in the first box, and the Sys_id of the icon in the second box.</td>
</tr>
</tbody>
</table>

Note: You can find your available icons on the Icon [sys_sg_icon] table. You can right-click any icon on this list and select Copy sys_id from the context menu to copy the sys_id.

4. Click Submit.

Example:
This example demonstrates how to apply a text decorator icon to all records on an incident list in the New state.
This example applies to the **State** element of the **Incident List Screen Item View** UI style. The condition field has been set, so the icon only appears for records in the **New** state. The icon is located on the **Icon** [sys_sg_icon] table.
Add a UI style to an applet

After you create a UI style for a field, you must add it to an applet.
Before you begin
Role required: admin
The process to add UI styles to your applets is different depending on whether the applet is using a pattern from the pattern set 1 or pattern set 2.

Apply a UI style to a pattern from set 1
Use these steps to apply a UI style if you have chosen a pattern from set 1.

Before you begin
Role required: admin

Procedure
1. In Studio, navigate to Mobile Studio > Applets, then click the expand arrow to see the list of applets for your application.
2. Create an applet or open an existing applet where you want to add UI styles.
3. From the applet configuration tab, add the field you want to add styles to, to the primary screen or details screen header. Only certain field locations in the header support UI styles. Make sure 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. In the UI Style Configuration section, only the E1 field is available, and selection is disabled.

4. In the UI Style Configuration section, move styles from the All Styles list to the Selected Styles list to apply these styles. The table selected for the UI style must match the table for the data item associated with your applet. Only UI styles that meet this criteria appear in the All Styles section of the UI Style Configuration section.
5. Click Save.

Apply a UI style to a pattern from set 2
Use these steps to apply a UI style if you have chosen a pattern from set 2.

Before you begin
Role required: admin
Procedure

1. In Studio, navigate to **Mobile Studio > Applets**, then click the expand arrow to see the list of applets for your application.

2. **Create an applet** or open an existing applet where you want to add UI styles.

3. **Optional**: If you have not yet created a UI style for this pattern, you can create one within Studio by clicking the **Create & Edit UI Styles** button.

4. Click **Map UI Styles** to open the item view record.

5. On the **UI Styles** field, click the lock icon (🔒) to open to display the list of UI styles assigned to the item view.
6. On the **UI Styles** field, click the **Add/Remove multiple** icon ( ) to open a list of UI styles.
7. Move UI styles from the list on the left to the list on the right to add them to your applet.
8. Click **Save** when you are finished adding UI styles. You are directed back to the item view record after clicking **Save**.
9. Click **Updated** in the item view record to save your changes.

**Mobile icons**

Use icons to provide unique visual identifiers for navigation tabs and quick actions in your mobile applications.

**Icons for navigation tabs and quick actions**

Applets and applet launchers have a static selection of icons. However, navigation tabs use customizable icons that are stored on the icon [sys_sg_icon] table.

**Icon records**

You can see existing mobile icon records on the icon [sys_sg_icon] table by entering `sys_sg_icon.list` in the navigation filter of your instance. The fields on the icon form define the appearance of your icon.

For quick actions, you can select an icon using the icon picker in Studio.

### Icon form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the icon.</td>
</tr>
<tr>
<td>Icon</td>
<td>Icon field that consists of one or more key and value pairs. Each value defines the appearance of your icon. See the following table for descriptions of these keys.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of icon</td>
</tr>
</tbody>
</table>
Icon form (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Font: Use font type for icons that are used in quick actions.</td>
</tr>
<tr>
<td></td>
<td>• Image: Use image type for icons that are used in navigation bar tabs. Image type icons do not use <strong>BackgroundColor</strong>, <strong>FontColor</strong>, or <strong>Shape</strong> values in the <strong>Icon</strong> field.</td>
</tr>
</tbody>
</table>

Icon field keys

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FontColor</td>
<td>Color for the icon image. The value must be a valid web color. For example, enter #000000 for a black image.</td>
</tr>
<tr>
<td>FontName</td>
<td>Name of the font set that is used for this icon. Enter as the value in the following table.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the font image that is used for this icon. Use this key to define the icon that is used for image type icons. See the following table for the available navigation bar icons.</td>
</tr>
<tr>
<td>Value</td>
<td>Value that determines which icon in the font is used for your icon image. See the following tables for the available font icons and values. Use the font name on the table as the value for this field.</td>
</tr>
</tbody>
</table>

Navigation bar icons

Navigation bar icons consist of a name and an icon value. Your instance theme determines the color of the navigation bar icons and background. The images available for use in the navigation bar icons are displayed in the table in the next section.

You can create icons for your navigation tabs using any of the images that are listed in this table. Depending on the plugins that you activated on your instance, some icon records in the icon [sys_sg_icon] table may be using some of these images. By default, your instance doesn’t have icon records for each image.

To see a list of existing navigation bar icons on your instance:

1. In the navigation filter, enter `sys_sg_icon.list` to open a list of icon records.
2. Filter your list for `<Type><is><Image>`. 
**Note:** You can use existing icons in your custom applications, but do not change or rename existing icons. These icons may be used in other applications in your mobile environment.

For more information on configuring navigation bars, see Configure the navigation bar.

### Available navigation bar icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Icon</th>
<th>Name</th>
<th>Icon</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Cart" /></td>
<td>Cart</td>
<td><img src="image" alt="Comment" /></td>
<td>Comment</td>
<td><img src="image" alt="Inventory" /></td>
<td>Inventory</td>
</tr>
<tr>
<td><img src="image" alt="Home" /></td>
<td>Home</td>
<td><img src="image" alt="Map" /></td>
<td>Map</td>
<td><img src="image" alt="Bag" /></td>
<td>Bag</td>
</tr>
<tr>
<td><img src="image" alt="User" /></td>
<td>User</td>
<td><img src="image" alt="Request" /></td>
<td>Request</td>
<td><img src="image" alt="Group" /></td>
<td>Group</td>
</tr>
<tr>
<td><img src="image" alt="Hardware" /></td>
<td>Hardware</td>
<td><img src="image" alt="Chart Bar" /></td>
<td>Chart Bar</td>
<td><img src="image" alt="Clock" /></td>
<td>Clock</td>
</tr>
<tr>
<td><img src="image" alt="Heart" /></td>
<td>Heart</td>
<td><img src="image" alt="Megaphone" /></td>
<td>Megaphone</td>
<td><img src="image" alt="Project Status" /></td>
<td>Project Status</td>
</tr>
<tr>
<td><img src="image" alt="Star" /></td>
<td>Star</td>
<td><img src="image" alt="Knowledge" /></td>
<td>Knowledge</td>
<td><img src="image" alt="Agile" /></td>
<td>Agile</td>
</tr>
<tr>
<td><img src="image" alt="File" /></td>
<td>File</td>
<td><img src="image" alt="Wrench" /></td>
<td>Wrench</td>
<td><img src="image" alt="Clipboard" /></td>
<td>Clipboard Exclamation</td>
</tr>
</tbody>
</table>
Available navigation bar icons (continued)

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Icon</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Calendar Icon" /></td>
<td>Calendar</td>
<td><img src="image" alt="Bell Icon" /></td>
<td>Bell</td>
</tr>
</tbody>
</table>

Quick action icons

Quick action icons have configurable name, icon, and color values. The images that are available for use in quick action icons appear in the table in the next section.

Select a quick action icon using the icon picker when you create a quick action.

For more details on quick action configuration, see Quick actions.
<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Icon</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="User" /></td>
<td>User</td>
<td><img src="image" alt="Check Circle" /></td>
<td>Check Circle</td>
</tr>
<tr>
<td><img src="image" alt="Group" /></td>
<td>Group</td>
<td><img src="image" alt="Edit 2" /></td>
<td>Edit 2</td>
</tr>
<tr>
<td><img src="image" alt="Add User" /></td>
<td>Add User</td>
<td><img src="image" alt="Wrench" /></td>
<td>Wrench</td>
</tr>
<tr>
<td><img src="image" alt="Cart" /></td>
<td>Cart</td>
<td><img src="image" alt="Phone" /></td>
<td>Phone</td>
</tr>
<tr>
<td><img src="image" alt="Flag" /></td>
<td>Flag</td>
<td><img src="image" alt="Hardware" /></td>
<td>Hardware</td>
</tr>
<tr>
<td><img src="image" alt="Star" /></td>
<td>Star</td>
<td><img src="image" alt="Mail" /></td>
<td>Mail</td>
</tr>
<tr>
<td><img src="image" alt="Comment" /></td>
<td>Comment</td>
<td><img src="image" alt="Calendar" /></td>
<td>Calendar</td>
</tr>
</tbody>
</table>
### Available quick action icons (continued)

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Icon</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Comments" /></td>
<td>Comments</td>
<td><img src="image" alt="Map Pin" /></td>
<td>Map Pin</td>
</tr>
<tr>
<td><img src="image" alt="Bag" /></td>
<td>Bag</td>
<td><img src="image" alt="Megaphone" /></td>
<td>Megaphone</td>
</tr>
<tr>
<td><img src="image" alt="Headphones Mic" /></td>
<td>Headphones Mic</td>
<td><img src="image" alt="Chart Bar" /></td>
<td>Chart Bar</td>
</tr>
<tr>
<td><img src="image" alt="Copy" /></td>
<td>Copy</td>
<td><img src="image" alt="Knowledge" /></td>
<td>Knowledge</td>
</tr>
<tr>
<td><img src="image" alt="Clipboard" /></td>
<td>Clipboard</td>
<td><img src="image" alt="Cloud" /></td>
<td>Cloud</td>
</tr>
<tr>
<td><img src="image" alt="Clipboard Exclamation" /></td>
<td>Clipboard Exclamation</td>
<td><img src="image" alt="Heart" /></td>
<td>Heart</td>
</tr>
<tr>
<td><img src="image" alt="Web Link" /></td>
<td>Web Link</td>
<td><img src="image" alt="Trash" /></td>
<td>Trash</td>
</tr>
</tbody>
</table>
Available quick action icons (continued)

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Icon</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Information Icon]</td>
<td>Information</td>
<td>![File Icon]</td>
<td>File</td>
</tr>
</tbody>
</table>

Create an icon for a mobile navigation tab

Create an icon for a navigation tab in your mobile applications to help your user quickly identify their apps.

**Before you begin**
Role required: admin

**Procedure**

1. In the navigation filter, enter `sys_sg_icon.list` to open a list of Icon [sys_sg_icon] records.
2. Click **New**.
3. On the form, fill in the fields.

**Icon form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Descriptive name for your icon. You may want to include something to identify the icon as a navigation tab icon. For example, you can add &quot;-nav&quot; to the end of the name.</td>
</tr>
<tr>
<td>Icon</td>
<td>Text fields within the Icon field that you can use to define key and value pairs that control the appearance of the icon. In the first field, enter Name. In the second field, enter the name of the icon that you want to use. For a list of the available icons for navigation bar tabs, see the reference in the preceding section.</td>
</tr>
<tr>
<td>Type</td>
<td>Icon type. Select <strong>Image</strong>.</td>
</tr>
</tbody>
</table>

4. Click **Submit**.
5. Navigate to **System Applications > Studio**.
6. Open your mobile application.
7. In the Application Explorer, navigate to **Mobile Studio > Application Menu**, and select the mobile app where you want to use your icon.
8. In the Navigation tabs section of the Navigation form, click the navigation tab where you want to use your icon.
9. In the Tab form, update the **Icon** field with your new icon.
10. Click Update.

**Note:** Changes made to the navigation bar are not visible to your users until they have logged out and logged back into the mobile app.

**Mobile functions**

Configure functions in Studio to determine which actions users can perform in the mobile app.

**Function types**

**Actions**

Use action functions to change data, such as assigning a task to yourself or adding a comment to a record. Action functions require a write-back action item to operate. Configure your actions with input parameters to include user input in the changes you make. For more detail on this function type, see Action functions.
Navigations

Use navigation functions to transition from your current screen to another applet or applet launcher. For example, opening a record from a list, or moving from an employee user profile screen to a manager user profile screen. For more detail on this function type, see Navigation functions.

Smart buttons

Use smart button functions enable your users to take an action using the native capabilities of their mobile device. These buttons give your users quick access to phone, email, locations or navigate to a specific URL. For more detail on this function type, see Smart button functions.

Function context

When you create an action, you must choose whether the function context is record or global. Choose a context based on what you want to function to do.

Record context
Use record context for functions that use the current record. For example, a function on a form used to update or edit the record would use record context.

**Global context**

Use global context the function is not required to act on an existing record. For example, a function to create a new record on a table, or global actions that appear in an applet launcher.

**Function locations**

For each function you create for an application, you must associate it with a specific location. You can associate most functions with a top menu, a swipe, or a specific field. For details on this configuration, see Associate a function with a location in the app.

**Carried parameters**

Parameters are a way of creating a variable or a placeholder that is waiting for input from either the user or the database. The variable then queries the database or the user for more information. For examples detailing how to use parameters in your functions, see Mobile parameter tutorials.

**Action functions**

Action functions make changes to records on your instance.

Use action functions to perform tasks such as assigning a task to yourself or adding a comment to a record. Action items can optionally include an input parameter, which collects input from your user to update a record.

**Action items**

Each action function includes an action item, which defines what the action function is and how it works. Learn more about action items at Configure an action item.

**Action function types**

**Non-parametrized action function**

Actions functions allow the user to change something in the database. For example, making an update or adding a comment to a record requires an action function. Use non-parametrized action function in situations where no user input is required.

For details on configuring these actions, see Configure an action function.
Parametrized action function

Use parametrized action functions to request additional information from your end users when they interact with an action. For example, you can allow your end users to provide a reason for rejecting an approval by adding a text input parameter to a Reject button. When you create an action item that has parameters, you must also configure the UI parameters for the action function.

For details on configuring these actions, see Create an action function with parameters.

Chat launcher action function

Create a contextual link in your mobile application so that your users can connect to Virtual Agent or Live Agent, and receive information that pertains to their issues. By using a contextual link, your users automatically receive the information that you define about the relevant record.

For details on configuring these actions, see Configure a contextual link to Virtual Agent.

Note: Connect chat is not supported in the native chat launcher.

Configure an action function

Action functions enable the user to change something in the database. For example, making an update or adding a comment to a record requires an action function.

Before you begin

Role required: admin
Procedure
1. In the Application Explorer, navigate to **System Applications > Studio**.
2. Select your application.
3. Navigate to **Mobile Studio > Functions > Actions**.
4. Click the popout icon (MBED) to open the Actions list in a tab.
5. In the upper right corner of the list, click **Create new**.
6. On the form, fill in the fields.

**Actions function fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the action. Choose a name that is easy to identify.</td>
</tr>
<tr>
<td>Description</td>
<td>Additional information about the action.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of action. Select <strong>Action item</strong>.</td>
</tr>
<tr>
<td>Context</td>
<td>The level to apply the action to.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Record</strong>: Applies 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 <strong>Input source</strong> field set to Auto fill, you must specify the table in the <strong>Table</strong> field.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Global</strong>: Applies an action at the global, or list level. For example, use this option to add a button that creates a record. Do not use the Global setting for actions that are in the context of a record, or actions that include a UI parameter with the <strong>Input source</strong> field set to Auto fill. For more information on button locations, see <strong>Associate a function with a location in the app</strong>.</td>
</tr>
<tr>
<td>Action item</td>
<td>Action item to associate with the action. For more information, see <strong>Configure an action item</strong>.</td>
</tr>
<tr>
<td>Allow images upload</td>
<td>Allows images to be uploaded. Images are stored in the attachment [sys_attachment] table.</td>
</tr>
<tr>
<td>Show signature field</td>
<td>Requires a user signature before submitting the action. Users can either sign with their fingers or type their names in the signature field. If <strong>Allow images upload</strong> is selected, an image of the signature is sent to the Signature Images [signature_image] table.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>when the action executes. If selected, the signature form displays on a separate page. To overlay the signature form, use the <strong>Signature</strong> option in the <strong>Preconditions</strong> field.</td>
</tr>
<tr>
<td>Use Overlay</td>
<td>Overlays a text input parameter on the current details screen. Exactly one text input parameter must be defined for the action. If this option is not defined, the input parameter displays on a separate screen.</td>
</tr>
<tr>
<td>Preconditions</td>
<td>Select an option to require user confirmation before submitting the action.</td>
</tr>
<tr>
<td></td>
<td>• <strong>None</strong>: Do not require user confirmation. This option is the default.</td>
</tr>
<tr>
<td></td>
<td>• <strong>HR Task Signature</strong>: Require a user signature before submitting the action. Users can either sign with their fingers or type their names in the signature field. If <strong>Allow images upload</strong> is selected, an image of the signature is sent to the Signature Images [signature_image] table when the action executes. When this option is selected, the signature form overlays the current screen.</td>
</tr>
<tr>
<td></td>
<td>• <strong>HR Task Credentials</strong>: Require users to input their credentials before submitting the action. This option is supported only with local authentication.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Re-authentication</strong>: Requires users to re-authenticate their credentials before executing an action. For more information, see Configure mobile re-authentication system properties.</td>
</tr>
</tbody>
</table>

**Condition tab fields**

<table>
<thead>
<tr>
<th>Condition Type</th>
<th>Type of condition to use.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Declarative</strong>: Adds a condition builder to the form. For more information, see <a href="#">condition builder</a></td>
</tr>
<tr>
<td></td>
<td>• <strong>Script</strong>: Adds a script condition field to the form.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Displays only if <strong>Context</strong> is set to <strong>Record</strong>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table</th>
<th>The table that the action applies to.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td>Conditions that must be met to use the action. For example, you could prevent users from resolving an incident that is in a state of closed, resolved, or canceled.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Roles</strong></td>
<td>Limit user access to an action by role.</td>
</tr>
<tr>
<td><strong>Messages tab fields</strong></td>
<td></td>
</tr>
<tr>
<td>Show confirmation message</td>
<td>Displays a confirmation message to verify whether a user wants to continue with the action.</td>
</tr>
<tr>
<td>Confirmation message</td>
<td>The confirmation message to display.</td>
</tr>
<tr>
<td>Note: Displays only if Show confirmation message is selected.</td>
<td></td>
</tr>
<tr>
<td>Confirm label</td>
<td>Label of the button to confirm the action.</td>
</tr>
<tr>
<td>Note: Displays only if Show confirmation message is selected.</td>
<td></td>
</tr>
<tr>
<td>Cancel label</td>
<td>Label of the button to cancel the action.</td>
</tr>
<tr>
<td>Note: Displays only if Show confirmation message is selected.</td>
<td></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, {{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, {{number}} could not be updated.</td>
</tr>
<tr>
<td><strong>Acknowledgment Messages tab fields</strong></td>
<td>Require user acknowledgment before submitting the action. If this field is enabled, the Context field must be Record.</td>
</tr>
<tr>
<td>Show acknowledgment text</td>
<td>Field that includes the acknowledgment text you want to display to the user. Select the table for the field in the Condition tab. Only String fields are supported.</td>
</tr>
<tr>
<td>Note: Displays only if Show acknowledgment text is selected.</td>
<td></td>
</tr>
<tr>
<td>Confirm label</td>
<td>Label for the confirmation button that appears below the acknowledgment text. The default value is I Agree.</td>
</tr>
<tr>
<td>Note: Displays only if Show acknowledgment text is selected.</td>
<td></td>
</tr>
<tr>
<td><strong>Offline Properties tab fields</strong></td>
<td>Option for displaying and defining the offline features available.</td>
</tr>
<tr>
<td>Offline</td>
<td>Hide fields after the user performs the action in offline mode. For example, if users</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Field</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Show fields</td>
<td>Display fields after the user performs the action in offline mode. For example, if users assign a task to themselves, you could show the <strong>Work notes</strong> field.</td>
</tr>
<tr>
<td>Hide functions</td>
<td>Hide functions after the user performs the action in offline mode, for example, when a user taps the Start Work function in offline mode.</td>
</tr>
<tr>
<td>Show functions</td>
<td>Display functions after the user performs the action in offline mode, for example, when a user taps the Start Work function in offline mode.</td>
</tr>
<tr>
<td>Mark as zombie on screen</td>
<td>Display the record in a lighter color (grayed out) to indicate that the user changed a record until the mobile app returns to online mode and the changes are saved.</td>
</tr>
<tr>
<td><strong>Action Completion tab fields</strong></td>
<td></td>
</tr>
<tr>
<td>Refresh full screen after action</td>
<td>Option for the current screen to automatically refresh when an action completes.</td>
</tr>
<tr>
<td></td>
<td>This option is available only when <strong>Context</strong> is set to <strong>Global</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The refresh option is supported for screens (applets) but not applet launchers.</td>
</tr>
<tr>
<td>Jump to previous screen</td>
<td>Redirects the user to the previous screen after completing an action.</td>
</tr>
<tr>
<td>Show refresh on previous screen</td>
<td>Shows a New Update message after the user completes the action and redirects back to previous screen.</td>
</tr>
<tr>
<td>Jump to screen after successful action completion</td>
<td>Displays additional screen navigation configuration options after the user completes an action. Use this field to determine whether user interaction is required before navigating to a specified screen.</td>
</tr>
<tr>
<td></td>
<td>This option is available only when a table is selected in the <strong>Condition</strong> tab.</td>
</tr>
<tr>
<td>Destination screen</td>
<td>The screen the user is redirected to after successful action completion. This screen displays newly created records, if available.</td>
</tr>
<tr>
<td></td>
<td>This option is available only when <strong>Jump to screen after successful action completion</strong> is selected.</td>
</tr>
<tr>
<td>Jump to screen without user interaction</td>
<td>Sends the user to the destination screen.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Auto-populate screen parameters</td>
<td>Automatically populates parameters in the destination screen. This option is available only when <strong>Jump to screen after successful action completion</strong> is selected.</td>
</tr>
</tbody>
</table>

**Note:** While all the options in the **Action completion tab** are selectable, their actions may overlap with each other. Make sure to test the settings so the results produce the outcomes you expect.

**What to do next**

After you create an action function, you must 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**.

**Configure an action item**

For an action function to work, you must create an action item to associate with the action function. Action items define what the action function is and how it works.

**Before you begin**

Before creating an action item, create an action function.

Role required: admin

**About this task**

Most action items use parameters.

Use action items to define what an action function does when a user uses that function.

The following steps detail creating an action without parameters. To create a parametrized action item, see **Configure an action item with parameters**.

**Note:** ServiceNow mobile apps are unable to perform any actions that cannot be performed in the platform web-based interface. For example, if you use 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 the same requirements. Keep this in mind when creating actions, so that you can add the correct parameters.

Role required: admin
Procedure
1. In Studio, navigate to Mobile Studio > Action Items.
2. Click the pop out icon ( ) to open the Actions items list in a tab.
3. Click Create a new action item.
4. Complete the following fields as needed.

**Action item fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>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.</td>
</tr>
<tr>
<td>Description</td>
<td>More information to help you identify the action item.</td>
</tr>
</tbody>
</table>
| Type                 | The kind of action item. Choose from the following:
  • New
  • Update
  • Delete
  • Script
  Different fields appear on the action item form depending on the type of action you select. |
<p>| Table                | The table the action item applies to, for example, Incident. |
| Execution Script     | The script executed by the action. This field only appears if you select Script as the type. For more information, see the example below. To make use of an input from a parameter screen in your scripts, use parm_input.&lt;InputName&gt; To make use of a variable from a parameter screen in your scripts, use parm_variable.&lt;VariableName&gt; |
| Use current record as condition | Whether you want a separate set of query conditions for the action item. If selected, the Query conditions field is disabled. For update or delete actions, you must define the record 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. |
| Query Condition      | Filter conditions that apply to the action item. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set field values</td>
<td>Determine the field values for an action. For example, if you want to create an action that updates an incident with a state of Resolved, you may the field values for State = Resolved. You can also create parametrized items to pass into the field value.</td>
</tr>
<tr>
<td>Parameter screen</td>
<td>Select a parameter screen to use for this action item.</td>
</tr>
</tbody>
</table>

5. Click **Submit**.

**Example:**
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 assigns 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);
```

The following example uses a script to perform a navigation completion functionality after an action is performed. Enter `actionResult` as the function and then define `setRedirectionInfo(gr.getUniqueValue(), gr.getTableName())` to specify where to navigate to, once the action is performed.

```javascript
(function WriteBackAction(input, actionResult) {
    var gr = new GlideRecord('incident');
    gr.get(input.sys_id);
    gr.short_description = 'Updated by Scripted Action';
    gs.addInfoMessage(gs.getMessage("This is the First success message"));
    gs.addInfoMessage(gs.getMessage("This is the Second success message"));
    gs.addInfoMessage(gs.getMessage("This is the Third success message"));
    gr.update();

    actionResult.setRedirectionInfo(gr.getUniqueValue(), gr.getTableName());
})(input, actionResult);
```

If you use parameters for the action item, you can call them in the script. The call in the script must match the parameter name exactly. For example, if the parameter name is `wb_wot_reject_work_note`, as in the first script above, you can call it in the script using `gr.work_notes = input.wb_wot_reject_work_note;`.
What to do next
Associate the action item with an action function.

Configure an action item with parameters
Parameters determine the information you are passing into the action to ensure you are changing the correct record and to enforce required fields as needed. Create an action item with parameters to define the changes being made to an action and how the changes get made.

About this task
Use action items to define what an action function does when a user uses that function. The following steps detail creating an action with parameters. To create a non-parametrized action item, see Configure an action item.

Procedure
1. In Studio, in a mobile application, navigate to Mobile Studio > 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 Configure 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 distinguish easily.
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 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
   
   ❗ Note: Making a Boolean mandatory has no effect. Boolean fields are always considered to have a value. A selected check box has a value of true and an unselected check box has a value of false. Either of these values satisfies the requirement of a mandatory field.
   • 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 you created as a condition in the condition builder for the action item.
Create an action function with parameters

Request additional information from your end users when they interact with an action. For example, you can allow your end users to provide a reason for rejecting an approval by adding a text input parameter to a Reject button. When you create an action item that has parameters, you must also configure the UI parameters for the action function.

About this task
The UI parameter fields do not appear on the action function unless you have associated an action item with parameters to the action function.

Procedure
1. In Studio, in a mobile application, navigate to Mobile Studio > 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 Configure an action function.
3. In the UI Parameters related list, click New.
4. Complete the following fields as needed.

   **UI Parameter 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 that you choose something you can easily discover later. If the parameter requires user input, use a user-friendly name.</td>
</tr>
<tr>
<td>Parameter type</td>
<td>Choose Screen or Button.</td>
</tr>
<tr>
<td>Field</td>
<td>Definition</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Button</td>
<td>Choose a button to associate with this parameter. This field only appears when the <strong>Parameter type</strong> field is set to <strong>Button</strong>.</td>
</tr>
<tr>
<td>Screen</td>
<td>Choose a screen to associate with this parameter. This field only appears when the <strong>Parameter type</strong> field is set to <strong>Screen</strong>.</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>The source for the parameter input. Choose <strong>User Input</strong> or <strong>Auto fill</strong>.</td>
</tr>
<tr>
<td>Input type</td>
<td>The UI used to complete the variable. The available options in this field depend on the choice entered in the <strong>Input source</strong> field. Choose from one of the following options.</td>
</tr>
</tbody>
</table>

**User input options:**

**Text**

Provides a simple text field. This option works best for fields that require 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.

**Note:** The list input returns a maximum of 1000 results. For references that require more than 1000 results, use the **Searchlist** input type.

**SearchList**

Provides a search bar so that users...
<table>
<thead>
<tr>
<th>Field</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>can search in a list. Select a reference field on a table for users to search on.</td>
<td></td>
</tr>
<tr>
<td>QR/Barcode</td>
<td>Provides the option to search by QRC or barcode. Learn more about this feature in Mobile bar-code scanning.</td>
</tr>
<tr>
<td>Auto fill options</td>
<td></td>
</tr>
<tr>
<td>GPS Location</td>
<td>Auto-fills with the mobile user's location when the action is used. For detail on this feature, see Mobile location tracking configuration.</td>
</tr>
<tr>
<td>Date</td>
<td>Auto-fills with the date and time stamp of when the action is used.</td>
</tr>
<tr>
<td>Constant</td>
<td>Auto-fills with a constant value. The <strong>Constant value</strong> field appears on the form when this choice is selected.</td>
</tr>
<tr>
<td>Source field</td>
<td>Auto-fills from a specified field. The <strong>Button parent table</strong> and <strong>Source field</strong> fields appear on the form when this choice is selected.</td>
</tr>
<tr>
<td>User</td>
<td>Auto-fills with the mobile user's user record.</td>
</tr>
<tr>
<td>Field</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Input style</td>
<td>How the user interacts in the UI. Choose from inline or pop-up.</td>
</tr>
<tr>
<td>Constant value</td>
<td>Enter a static value to use for this parameter. This field only appears if you use the Constant input type.</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>
<tr>
<td>Table name</td>
<td>The name of the table you want to pull information from. This field only 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 table, select the incident table for the UI parameter as well.</td>
</tr>
<tr>
<td>Field name</td>
<td>The name of the field you want to pull information from. This field only appears if you use the List or the SearchList input type. Select a reference field from the table you selected. For example, select the Assign to field.</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 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. If you select this option, a table name and field are required.</td>
</tr>
<tr>
<td>Button parent table</td>
<td>The parent table for the source field. This field only appears if you use the Source field input type.</td>
</tr>
<tr>
<td>Source field</td>
<td>The field used for the source field. This field only 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 you created for the action item. For example, Assignee.

c. In the UI parameter field, search for the name of the UI parameter you created.

d. Click Save.

7. From the action function, click Update.

**What to do next**

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.

**Define banner display persistence**

Define if a message or banner displayed on a mobile device requires the user to acknowledge the receipt of the message.

**Before you begin**

Role required: admin

**About this task**

Configure specific messages to remain on a screen until the user actively dismisses the message. By default, messages fade after a few seconds. When a function button has multiple messages associated to it, then all associated messages are consolidated into a single message. If one of these messages contains a persistence attribute, then the single consolidated message adopts the persistence behavior.

Only one banner message is displayed at a time. Newer banner messages replace existing banner messages. This replacement occurs even if the original existing banner contains persistent attributes.

**Procedure**

1. Navigate to All > System Applications > Studio.
2. Select your application.
3. In Application Explorer, navigate to Mobile Studio > Functions > Actions.
4. Select a function where you want to add the persistence banner behavior.
5. Make sure Action item is selected for the Type field.
6. Select the Messages tab and enter text in the Success Message field.
7. Add the persistent behavior attribute.
   a. Select the Attributes tab.
   b. Select the Insert a new row area in the Name column.
   c. Select the reference lookup icon ( ) and select the label Alerts Require Dismissal from the list.
   d. Select the green check mark (✓) to confirm the selection.
   e. Select the area under the Value column to open the value text area.
   f. Enter the value true and select the green check mark to confirm the selection.
8. Right-click in the header and select Save.
Results
A banner remains on the screen until the user actively dismisses it.

Reference qualifiers for ServiceNow mobile
Admins use reference qualifiers to create filters that streamline data returned for a reference field. The results of configured reference qualifiers are viewed within your mobile device. Reference qualifiers facilitate the filtration of data extracted from reference tables, ensuring that only the desired values are returned. You can use one of three types of reference qualifiers, simple, dynamic, and advanced.

Note: Before you configure a reference qualifier to display filtered results in a mobile device, create an action function. For more information, see Create an action function with parameters.

The images show a comparative example, within the mobile platform, where a reference qualifier is used and not used. The image with the lower results found and different results listed, indicate that a reference qualifier configuration was implemented.
For more information

For information on the difference between simple, dynamic, and advanced reference qualifiers and how to configure them, see the following documentation topics:

Reference qualifiers
Configure reference qualifiers

Configure a contextual link to Virtual Agent

Use Studio to create a contextual link in your mobile application so that your users can connect to Virtual Agent and receive information that pertains to their issues. By using a contextual link, your users automatically receive the information that you define about the relevant record.

Before you begin
Role required: admin
Before you can create a virtual link to Virtual Agent feature, you must have the Glide Virtual Agent plugin (com.glide.cs.chatbot) installed on your instance. For more details, see Virtual Agent.

Perform the following steps in Studio.

Procedure
1. Navigate to All > Mobile Studio > Functions > Actions.
2. Click the pop-out icon ( disclosing button) to open the Actions list in a tab.
3. In the upper right corner of the list, click Create new.
4. On the form, fill in the fields.

Action function fields form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the action. Choose a name that is easy to identify.</td>
</tr>
<tr>
<td>Description</td>
<td>Descriptive information about the action.</td>
</tr>
</tbody>
</table>
5. Click Submit.

Create UI parameters for your Virtual Agent link

Create UI parameters to pass information from your record into Virtual Agent so that your users get the information they need for their issues.

Before you begin
Role required: admin

Procedure
1. In the UI parameters tab of your action function, click New.
2. On the form, fill in the fields.
   If a field isn’t mentioned in the following table, use the default values for that field.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name that describes the information that is being passed to Virtual Agent. For example, you might use incident_caller for a UI parameter that contains the caller for an incident record. Because these names are used in scripts, do not use spaces in the names.</td>
</tr>
<tr>
<td>Parameter type</td>
<td>Parameter type that can be a button or screen. Select Button.</td>
</tr>
<tr>
<td>Button</td>
<td>Action that you created in the previous steps.</td>
</tr>
<tr>
<td>Input source</td>
<td>Source where the UI parameter gets its value. Select Auto fill.</td>
</tr>
<tr>
<td>Input type</td>
<td>Location where the user inputs data for the parameter. Select Source field.</td>
</tr>
<tr>
<td>Parent button table</td>
<td>Table that is used in the Table field of the action that you created in the previous steps.</td>
</tr>
<tr>
<td>Source field</td>
<td>Field that contains the information that is being passed to Virtual Agent. For example,</td>
</tr>
</tbody>
</table>
Field | Description
--- | ---
select **Caller** if you are passing the value of the **Caller** field on an incident form.

**Note:** The table and sys_id of your record are automatically passed to Virtual Agent, so you don’t need to create UI parameters for these values.

3. Click **Save**.

4. **Optional:** Repeat steps 1 through 3 to create additional UI parameters for any other information that you want to pass to Virtual Agent.

You can direct your mobile users to a specific Virtual Agent topic by creating a **search_text** UI parameter. You can create this parameter using the preceding steps, with the following changes:

<table>
<thead>
<tr>
<th>UI parameter form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the UI parameter. For this example, enter <strong>search_text</strong>.</td>
</tr>
<tr>
<td>Input Type</td>
<td>Location where the user will input data for the parameter. For this example, select <strong>Constant</strong>.</td>
</tr>
<tr>
<td>Constant Value</td>
<td>Value that matches a keyword from one of your Virtual Agent conversations. When your users tap the chat button, they are directed to the matching topic.</td>
</tr>
</tbody>
</table>

**What to do next**
The UI parameters that you created are available to your Virtual Agent topics within the vaContext object. You can access these parameters in Virtual Agent Designer by entering vaContext, followed by the name of your UI parameter. For example, you can access the incident_caller UI parameter by entering vaContext.incident_caller. For more details on using context variables in Virtual Agent, see Virtual Agent scripts.

**Enable Virtual Agent for mobile applications**
Give your users the ability to chat with a virtual agent through a ServiceNow mobile application.

**Before you begin**
Role required: admin
To include virtual agent functionality in your mobile applications, you first must activate and configure virtual agent. For details on this process see Planning and configuring Virtual Agent.

**Procedure**
1. Navigate to **All > System Applications > Studio**.
2. In the application explorer on the left edge of the screen, select **Functions > Actions**, and click the pop-out icon that appears to the right of **Actions**.
3. Click the **Create New** button to create a new function.
4. In the **Type** field, select **Chat Launcher**.
5. In the **Context** field, select **Global**.
6. Click **Submit**.
7. On your instance, outside of Studio, navigate to **System Mobile > Applet Launchers**.
8. Open the record for the applet launcher where you want to add your Virtual Agent quick action.
9. In the **Body** section, click **Insert a new row** under the **Quick Actions Menu Maps** list.
10. Select the function created in the previous steps.
11. Click **Update**.

**Navigation functions**

Navigation functions transition users from their current screen to another applet or applet launcher.

Use navigation functions to transition from your current screen to another applet or applet launcher. For example, opening a record from a list, or moving from an employee user profile screen to a manager user profile screen. Navigation functions use the Global or Record context.

**Navigation function contexts**

**Global Context**

Use global context navigation functions in situations where the navigation does not depend on information from a record. For example, to navigate to a specific applet or applet launcher, you can use a global context navigation function. For examples of global context navigation configuration, see:

- Configure a navigation to an applet
- Configure navigation to an applet launcher

**Record Context**

Use record context navigation functions in situations where the navigation depends on information from a record. For example, you want to navigate from the **Assigned to** field in an incident record to the assignee's user record. In this case, you would use a record context navigation. For an example of record context navigation configuration, see Tutorial: Configure navigation from a list applet to another list applet.

**Configure a navigation to an applet**

Navigation functions enable you to define simple ways for end users to navigate within the mobile platform, for example, navigating to a record from a field on another record.
Before you begin
Before you create a navigation function, you should have a source applet and a destination applet for your navigation.
Role required: admin

Procedure
1. In Studio, navigate to Mobile Studio > Functions > Navigation.
2. Click the pop-out icon.
3. In the Navigations list, click Create New.
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 Description field, provide a description for your navigation function.
6. In the Destination Type field, select whether you are navigating to an applet or applet launcher.
7. In the Destination field, select the applet you want to navigate to.
   Depending on the destination screen selected in your Destination field, you might need to include parameter settings.
8. In the Context field, select whether the action should be available for a specific or global context. For an applet, select Global.
9. Set the Available Offline option to determine whether the navigation function is available offline.
The applet and the application containing the navigation function must be marked as available offline for the navigation function to work offline. For more information on offline mode, see Offline mode.
10. If you added a destination applet that has a parametrized data item, in the Parameter Setting section, update the redirection parameter fields.
   a. Click the value in the parameter name field.
   b. Complete the parameter settings.
      Note that some of the fields vary depending on the value you select for the Type field.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination UI parameter</td>
<td>The name of the parameter you created for the data item. If the parameter Type value is User Input, use a user-friendly name.</td>
</tr>
<tr>
<td>Type</td>
<td>Source of information for the parameter. The available options are:</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 uses a constant value.</td>
</tr>
<tr>
<td></td>
<td>• Source UI parameter</td>
</tr>
</tbody>
</table>

© 2022 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>Source field</td>
<td>Source for the parameter. This field appears only if Field is the redirection parameter type. The table defined in the data item determines the available fields.</td>
</tr>
<tr>
<td>Constant value</td>
<td>Value to appear for the data item parameter. This field appears only if Constant is the redirection parameter type.</td>
</tr>
<tr>
<td>Source UI parameter</td>
<td>UI parameter for the source screen. This field appears only if Source UI Parameter is the redirection parameter Type value.</td>
</tr>
</tbody>
</table>

c. Click Save.

11. In the Advanced Configurations section, determine the display conditions and the roles required for the navigation function.
For example, you could define a different function or action for the same button according to user roles.

a. Select a table in the Display Conditions section.

b. In the condition builder, select filter conditions to limit when the navigation function is available.
For example, if you want users to be able to navigate to a problem from an incident form, you can configure the navigation function to appear only when the problem field has a value.

c. Add the roles that can use the navigation function to the Select Roles section in the Roles Permission section.

12. Click Save.

**What to do next**
After you create a navigation function, you must associate it with a specific location in the mobile app. For more information, see Associate a function with a location in the app.

**Tutorial: Configure navigation from a list applet to another list applet**

Configure a navigation function that directs your users to another list applet when they tap a record from a list.

Watch this video on how to configure mobile studio.
When you create a list applet, you can include a form applet that lets your users display record details when they tap a record. However, you can instead use a navigation function to direct your users to any other type of applet when they tap a record from a list. In this example, you can see a navigation function that directs users to a second list.
You can create incident priorities that are displayed to your users in an applet or within an applet launcher. When a user taps a priority entry in the primary list, they are directed to a second list showing all incidents that match that priority. Users can then tap an item from the secondary list to see that record in a form screen.

To create an incident priority that you can display in an applet or within an applet launcher, you create:

- A primary list that shows your incident priorities.
- A secondary list that shows the incidents matching the selected priority.
- A navigation function to list the primary and secondary lists.

Create a primary list

Create your first list. This list displays the priorities for your incidents. Your users can tap on any priority to see records that match that priority.

Before you begin

Role required: admin

About this task

Procedure

1. Navigate to All > System Applications > Studio.
2. Select a scoped application where you want to create your applet.
3. In the Application Explorer, navigate to Mobile Studio > Applets and select Applets.
4. Click the pop-out icon ( ) that appears to the right of Applets.
5. In the applet list, click Create an applet.
6. In the Name field, enter a name for your applet.
7. From the choose a template section, select List.
8. Deselect the option to include a form with this list.
   This list uses a navigation function to direct to another list.
9. Click Create New.
   The list applet is created. The configuration screen for your new applet displays in Studio.
10. In the Data Item field, click Add (+) to create a new data item for your list.
11. In the Name field, enter a name for the data item.
12. In the **Table** field, select *Incident [incident]*.

13. In the **All of these conditions must be met** section, enter `<Priority> <is one of> <1-Critical><2-High><3-Moderate>`.

14. In the **Group By** field, select *Priority*.

15. Click **Save**.

16. Click the tab that contains your list applet to return to the applet configuration screen.

17. In the **Data Item** field, select the data item that you created in the previous steps.

18. Click **Change Pattern**.

19. Select pattern 11 from the available patterns.

20. Click **Done**.

21. In the **Field Configurations** section, select *priority* for your field.

22. Click **Save**.

You have a list applet that shows an entry for each incident priority. Currently, nothing happens when you tap these entries. In the next steps, you'll create a second list to show the incidents that match a priority. Then, you'll create a navigation function to navigate to the list that shows priorities to the second list that shows the incidents matching that priority.
Create a secondary list
Create a second list of incidents to match the categories that your users selected in the primary list. This list uses a parametrized data set to receive the chosen priority from the primary list.

Before you begin
Role required: admin

Procedure
1. In the Application Explorer, navigate to Mobile Studio > Applets and select Applets.
2. Click the pop-out icon (ogeneity_button) that appears to the right of Applets.
3. In the applet list, click Create an applet.
4. In the Name field, enter a name for your applet.
5. Select List from the Choose a template section.
6. Select the option to include a form with this list.
   This list uses a navigation function to direct to another list.
   This second list shows your incidents. Selecting the form option lets your users see the incident form when they tap an incident.
7. Click Create New.
   The list applet is created. The configuration screen for your new applet appears in Studio.
8. In the Data Item field, click Add (ديدة) to create a new data item for your list.
9. In the Name field, enter a name for the data item.
10. In the Table field, select Incident [incident].
11. Click Save.
12. In the Parameter Definition section, click the add button (ديدة) to add a new parameter.
    The Parameter Definition pop-up window appears.
13. In the Name field, enter priority.
14. In the Type field, select String.
15. Click Save.
    The Parameter Definition pop-up window closes.
16. In the All of these conditions must be met section, enter <Priority> <is>.
17. Click the reference value button (ديدة), and select priority.
18. Click Save.
    You have created a parametrized data item to use in your second list. This data item uses a parameter to pass the priority from the first list that you created in previous steps.
19. Return to the configuration form for your second list.
20. In the Data Item field, select the parametrized data item that you created in the previous steps.
21. In the User Input Parameter Definition section, click the add button (ديدة) to add a new parameter.
22. In the Name field, enter priority.
    Leave all other fields at their default value.
23. Click Save.

24. In the Screen UI Parameter Mapping section, click the add button (➕) to add a new parameter mapping.

25. In the Item parameter field, select priority.

26. In the UI parameter field, select priority.

27. Click Save.

The list is now configured to use the priority that you selected in the first list. This information passes between lists using a parameter.

28. In the Field Configurations section, select the incident fields that you want to appear on the incident list.

29. At the top of the list configuration screen, click the Form Screen tab.

30. In the Field Configurations section, select the incident fields that you want to appear on the incident form.

31. Click Save.

You now have a second list, which displays the incidents in the category that your user selected from the first list. In the final steps, you’ll create a navigation function that can handle the navigation between the two lists.

Create a navigation function

Create a navigation function to connect your two lists. The navigation function handles the transition from one list to the next when the user taps a record on the list.

Before you begin

Role required: admin

Procedure

1. In the Application Explorer, navigate to Mobile Studio > Functions > Navigations.

2. Click the pop-out icon (的功能) that appears to the right of Navigations.

3. Click Create New.

4. In the Name field, enter a name for your navigation function.

5. In the Destination Type field, select Applet.

6. In the Destination field, select the parametrized list that you created in previous steps.

7. In the Context field, select Record.

8. In the Table field, select Incident [incident].

9. In the Parameter Name section, click the parameter. There should be a single parameter named priority. The Redirection Parameter pop-up window appears.

10. In the Source field field, select Priority.

11. Click Save.

You have configured a navigation function to direct to your second list, and used the parameter settings to pass the priority field to your second list.

12. In the Application Explorer, return to Mobile Studio > Applets and select your first list applet.

13. Click the Functions tab to display the functions configuration section.

14. Under List Item Function, click the add button (➕).
15. In the **Function** field, select the navigation function that you created in the previous steps.

16. **Click Save.**

Now that your navigation function is configured, you can tap a priority in your first list, and see a list of all incidents with that priority. You can also tap an incident in the second list to see the form for that incident.

**Configure navigation to an applet launcher**

Use a navigation function to direct your users to an applet launcher page. Use this navigation function in a quick action, footer button, or top menu.

Navigation functions direct your users to from one screen to another. In this example, you create a navigation function to direct your users to an applet launcher. After you have created this function, you can put it to use. You can use your navigation function anywhere you can normally use functions. This example shows how to include this function in a screen’s footer, a quick action, or a screen’s top menu.

**Create a navigation function**

Create a navigation function to direct to an applet launcher.

**Before you begin**

Role required: admin

**About this task**

In the following steps you create a navigation function to direct your users to an existing application launcher. You will need to have an existing applet launcher to use as a destination for this function. For detail on creating applet launchers, see [Launcher screens](#).

**Procedure**

1. In Studio, navigate to **Mobile Studio > Functions > Navigations**.

2. Click the pop-out icon ( ) that appears to the right of **Navigations**.

3. In the **Navigations** list, click **Create a new navigation**.

4. On the navigation form, fill in the fields.

**Navigation form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for your navigation function.</td>
</tr>
<tr>
<td>Destination Type</td>
<td>The destination type for this navigation. For this example, select <strong>Applet Launcher</strong>.</td>
</tr>
<tr>
<td>Description</td>
<td>Description for your navigation function.</td>
</tr>
<tr>
<td>Destination</td>
<td>The applet or applet launcher users will see after using this function. Select an applet launcher from a list of your available applet launchers.</td>
</tr>
<tr>
<td>Context</td>
<td>Whether the function applies at the record or table level. Select <strong>Global</strong>.</td>
</tr>
<tr>
<td>Available Offline</td>
<td>Whether the action will be available in offline mode.</td>
</tr>
</tbody>
</table>

5. **Click Save.**
Use your navigation function as a quick action

Use your navigation function as a quick action if users need to frequently access the launcher.

Before you begin
Role required: admin
Quick actions are appear on applet launcher pages, and serve as a quick way to provide shortcuts to commonly used applets and actions. Add your navigation function as a quick action to give your users an easily accessible shortcut to the launcher defined in your navigation function.

Procedure
1. In Studio, navigate to Mobile Studio > Applet Launchers.
2. Click the pop-out icon (⋮) that appears to the right of Applet Launchers.
3. Open an applet launcher where you want to add your quick action.
4. In the Quick Actions section, click the add button (+) to create a new quick action.
5. In the Create a Quick Action pop-up. Fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>The label for this quick action. Use a value to inform your users of what the action does. In this case, you could use the name of the destination applet launcher.</td>
</tr>
<tr>
<td>Icon</td>
<td>The icon that appears for this quick action.</td>
</tr>
<tr>
<td>Function</td>
<td>The function used by this quick action. Select the function you created in previous steps.</td>
</tr>
</tbody>
</table>

6. Click Save.
7. In the applet launcher form, click Save.

What to do next
Test the navigation using your mobile app. After completing the steps, you can see a quick action in your applet launcher with the label and icon you defined in these steps. Tapping the quick action will direct you to the applet launcher you defined in your navigation function.

Use your navigation function as a top menu selection
Use your navigation function as a top menu selection to give your users access to the application launcher from within an applet.

Before you begin
Role required: admin
Your users access the top menu function by tapping the icon in the upper right corner of an applet screen. In these steps you add your navigation function to a top menu function of an existing applet. If you have not configured any applets, you need to do that first. For detail on that process see Create a screen.
Procedure
1. In Studio, navigate to Mobile Studio > Applets.

2. Click the pop-out icon (🔍) that appears to the right of Applets.

3. Open an applet where you want to add your navigation function as a top menu action.

4. In the applet form, click Functions to access the applet’s functions.

5. In the Top Menu Functions section of the form, click the add button (+) to create a new top menu function.

6. In the Top Menu Function pop-up, Enter a label for your menu item in the Label field, and select the navigation function you created in earlier steps in the Function field.

7. Click Save.

8. On the applet form, click Save.

What to do next
Test the top menu action using your mobile app. After completing the steps, you can see a menu icon in the upper right corner of your applet. Tapping the menu icon displays the items in the menu. Your navigation function appears in this list using the name you provided in the Label field in step 6.

Use your navigation function as a footer function
Footer functions enable your end users to take an action on a details segment of a form. You can use the navigation function you created to give your users access to the application launcher from within the details segment of your forms. Unlike the top menu function, the form footer button is visible at the bottom of the form.

Before you begin
Role required: admin

Procedure
1. In Studio, navigate to Mobile Studio > Applets.

2. Click the pop-out icon (🔍) that appears to the right of Applets.

3. Open an applet where you want to add your navigation function as a top menu action.

   Note: The applet you select should be a form applet, or another applet that contains a form screen, such as a list, calendar, or map applet.

4. In the applet form, select the Body section.
   Click the Form Screen tab if you are accessing a form screen within another applet, such as a list.
5. In the footer functions section at the bottom of the form, click the add button (➕) to create a new footer function.

6. In the Footer Function pop-up, fill in the fields.

**Footer Function form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>The label for this footer function. Use a value to inform your users of what the action does. In this case, you could use the name of the destination applet launcher.</td>
</tr>
<tr>
<td>Function</td>
<td>The function used by this footer function. Select the function you created in previous steps.</td>
</tr>
<tr>
<td>Button Emphasis</td>
<td>The button emphasis for the footer function. This selection affects the color of the button. For a navigation function, select <strong>Primary</strong> or <strong>Secondary</strong>.</td>
</tr>
</tbody>
</table>

7. Click **Save**.

8. In the applet form, click **Save**.

**What to do next**
Test the footer function using your mobile app. After completing the steps, you can see a footer function button at the bottom of your form applet. Tap the button to navigate to the applet launcher you defined in the previous steps.

**Smart button functions**
Use smart buttons to interact with native applications on your mobile device, such as your phone, map, or email applications.
Use smart buttons to quickly perform actions you specify outside the app. These actions can include navigating to a location on a map, sending a text message or email to a contact, placing a phone call, or opening a URL in a browser. You can choose from any of the following options.
### Smart button types

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>Use the <strong>Address</strong> smart button type to navigate to an address on your instance using your mobile device's navigation software. For example, you can configure a button to direct your field technician to their next work location.</td>
</tr>
<tr>
<td>Email</td>
<td>Use the <strong>Email</strong> smart button type to send an email. The recipient can be a static email address, or an email address stored on a table on your instance. For example, you can configure a button to send an email with a preset subject from within an incident record.</td>
</tr>
<tr>
<td>Phone</td>
<td>Use the <strong>Phone</strong> smart button type to place a call or send an SMS text message. As with email, the recipient can be a static, or a number stored on an instance record. For example, you can configure a button on an incident record to call your customer. You can configure your smart button to automatically use the phone number in your customer's user record.</td>
</tr>
<tr>
<td>URL</td>
<td>Use the <strong>URL</strong> smart button to navigate to a web address. For example, you can configure a button to open your company's website. URLs can be relative or external. Relative URLs display within the app, while external URLs open in the mobile device's default browser.</td>
</tr>
</tbody>
</table>

### Smart button context

Context determines whether a smart button uses information in a record, or static information you define when creating the smart button.

**Record Context**

Use record context when you want to use information from the record where you have included your smart button. For example, you want to create a smart button to call the user listed in the **Caller** field of an incident. In this case you would select **Record** in the smart button’s context field. The number used for the phone call is specified by selecting a table and field where the caller's phone number is stored. Using this method, the number called changes dynamically when accessing incidents with different callers.

**Global Context**

Use global context when you want to create a button that does not depend on information in the record. For example, you want to create a smart button that
calls your company’s support number. This number remains the same no matter where the smart button is placed in your application. When creating a smart button with the global context, you have a **Phone Number** field where you can input the number.

For examples of configuring smart buttons, see [Configure a smart button](#).

**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 admin 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 is visible.</td>
</tr>
<tr>
<td>Roles Permission</td>
<td>Roles that can see the smart button. If no roles are selected, the button is visible to all users.</td>
</tr>
</tbody>
</table>

**Deep linking to third party applications**

ServiceNow mobile supports deep linking to third party applications using the **URL** smart buttons. To make use of this feature, enter a properly formatted URI in the **URL** field. The format for a deep linking URI varies depending on the target app. For information on these URI formats, refer to deep linking documentation provided by the third party vendors.

Administrators can use the `glide.sg.allowed_external_deeplinks` property to define which third party apps are available for deep linking. For details on this property, see [Configure which external apps are available for deep linking](#).

**Configure a smart button**

Smart buttons are actions that allow you to perform another action, such as sending an email, making a phone call, pulling up a location, or navigating directly to a URL.

**Before you begin**

Role required: admin

**Procedure**

1. In Studio, navigate to **Mobile Studio > Functions > Smart Buttons**.
2. Click the pop out icon (🔗) to open the Smart buttons list in a tab.
3. Click **Create New**.
4. In the new smart button form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for your smart button</td>
</tr>
</tbody>
</table>
| Type       | Type of smart button. Select from:
  • Address
  • Email
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
</table>
| Phone               | • Phone  
|                     | • URL  
|                     | See the table below for detail on button types.                                                                                       |
| **Note:** Choose a field appropriate to the type of smart button you are creating. For example, if you are creating a smart button with the type **Phone**, select a field that includes a phone number. |
| Description         | Unique description for your smart button to make it easy to identify.                                                                   |
| Context             | Context for your smart button. Select from:                                                                                           |
|                     | • Record  
|                     | • Global  
| **Note:** Smart buttons using the **Address** type must use the **Record** context. The **Context** field becomes read only when you select that type. |
| Use source value from field | Use a specific field from a table as the source of a URL.                                                                                 |
|                     | This field is only visible when you select **URL** in the **Type** field.                                                                |
| Table               | Table where you want to use your smart button.                                                                                         |
| fields              | Field where you want to add your smart button. The table selected in the **Table** field determines which fields are available.         |
| Phone Number        | Phone number called by the smart button. This field is only visible when you select **Phone** in the **Type** field, and **Global** in the **Context** field. |
| **Note:** If you have selected **Record** in the context field, the phone number called is the number in the **Fields** field. Be sure to select a field that contains a phone number. |
| Phone Type          | Type of phone the smart button will communicate with. Select from:                                                                     |
|                     | • Cellular  
|                     | • Land Line  
<p>|                     | This field is only visible when you select <strong>Phone</strong> in the <strong>Type</strong> field.                                                             |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS</td>
<td>The SMS message sent by the smart button. This field is only visible when you select <strong>Cellular</strong> in the <strong>Phone Type</strong> field.</td>
</tr>
<tr>
<td>Email Address</td>
<td>Email address to which the email is sent. This field is only visible when you select <strong>Email</strong> in the <strong>Type</strong> field, and <strong>Global</strong> in the <strong>Context</strong> field.</td>
</tr>
<tr>
<td><strong>Note:</strong> If you have selected <strong>Record</strong> in the context field, the email is sent to the email address in the <strong>Fields</strong> field. Be sure to select a field that contains an email address.</td>
<td></td>
</tr>
<tr>
<td>Email Subject</td>
<td>Subject for the email message. This field is only visible when you select <strong>Email</strong> in the <strong>Type</strong> field.</td>
</tr>
<tr>
<td>Email Content</td>
<td>Content of the email message. This field is only visible when you select <strong>Email</strong> in the <strong>Type</strong> field.</td>
</tr>
<tr>
<td>URL Label</td>
<td>The visible label of your URL link. This field is only visible when you select <strong>URL</strong> in the <strong>Type</strong> field.</td>
</tr>
<tr>
<td>URL Link</td>
<td>The URL to which your smart button will navigate. This field is only visible when you select <strong>Email</strong> in the <strong>Type</strong> field, and <strong>Global</strong> in the <strong>Context</strong> field.</td>
</tr>
<tr>
<td><strong>Note:</strong> If you have selected <strong>Record</strong> in the context field, and enabled the <strong>Use source value from field</strong> field, the button will use the value in the <strong>Fields</strong> field for the URL. Be sure to select a field that contains a URL.</td>
<td></td>
</tr>
</tbody>
</table>

5. **Optional:** In the **Advanced Configuration** 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 require a value in the phone number field.

**Results**

**What to do next**
After you create a smart button function, you must 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 parametrization to include record specific information in your smart buttons.

**Before you begin**
Role required: admin
About this task
This example demonstrates how parameters are used to improve the functionality of smart buttons. In this case, the smart button provides a 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 ServiceNow instance. Demonstrates how to find relative links in a ServiceNow instance

Procedure
1. Navigate to All > System Applications > Studio and open your mobile application.
2. Navigate to Mobile Studio > Functions > Smart Buttons and click the pop out icon (/gif) to open the Smart buttons list in a tab.
3. Click New to create a new smart button.
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>Description</td>
<td>Enter a description for your smart button</td>
</tr>
<tr>
<td>Context</td>
<td>Record</td>
</tr>
<tr>
<td>Use source value from field</td>
<td>Deselect</td>
</tr>
<tr>
<td>Table</td>
<td>Incident [incident]</td>
</tr>
<tr>
<td>URL Link</td>
<td>sp?id=search&amp;spa=1&amp;t=kb&amp;q={short_description}</td>
</tr>
<tr>
<td>URL Label</td>
<td>Show KB</td>
</tr>
</tbody>
</table>

5. Click Save.

Results
Tapping this button displays a list of knowledge base articles, using the short description for the incident as a search term.

Associate a function with a location in the app
For each function you create for an app, you must associate it with a specific location. You can associate most functions with a top menu, a swipe, or a specific field.
Applet launcher header functions allow your users to navigate to a specific parametrized form screen. Some common examples include navigating to a user's profile, or navigating to a cart.

For details on creating parametrized forms, see [Configure a parametrized form applet](#).

For details on how to use these forms in an applet launcher header, see [Configure a launcher screen header](#).
Chart header functions enable your users to navigate to a specific set of records from a chart screen. When a user is viewing a chart, they can click on the chart header function on the top right corner of the screen, to view a list of all the records that are within the chart. For details on using chart header functions, see Configure a navigation from a chart to a list applet.
Field functions

Field functions enable your end users to change the value of a field. Field functions only apply to items on the details screen in the body displayed area.

1. In Studio, open a form screen that you want to add a field function to. Make sure that there are elements in the body display area that you can add field functions to.

2. In the Details Segment of the Body tab, add a function under Field Functions.

3. Select the field that you want to add the function to and select the function. This function is the behavior that the system performs when a user taps the field function.

4. Click Save.

<table>
<thead>
<tr>
<th>Field function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Item</td>
</tr>
<tr>
<td>Category</td>
</tr>
<tr>
<td>Subcategory</td>
</tr>
<tr>
<td>Impact</td>
</tr>
<tr>
<td>Urgency</td>
</tr>
<tr>
<td>Assignment group</td>
</tr>
<tr>
<td>Assigned to</td>
</tr>
<tr>
<td>Parent Incident</td>
</tr>
</tbody>
</table>

© 2022 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.
Footer functions

Footer functions enable your end users to take an action on a details segment of a form.

Supported screens for the footer function are: form details screen, form embedded section screen, dynamic segments in a form screen, and list screens.

1. In Studio, open a form screen that you want to add a footer function to.
2. In the Details Segment of the Body tab, add a function under **Footer Functions**.
3. Add a label and a function. This function is the behavior that the system performs when a user taps the footer function.
4. Add a color option. Choose from **Primary**, **Destructive**, or **Secondary**.
5. Click **Save**.

List item functions

Use the list item function location to trigger a function when a user taps on a record in a list. This function location can for example, navigate a user to another applet when they tap a record.

1. Navigate to a list applet that you want to add a list item function.
2. Switch to the Functions tab.
3. In the List Item Functions section, click the Add icon (➕).
4. 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.
5. Click **Done**.
For an example of using this function location to navigate from one list to another, see Tutorial: Configure navigation from a list applet to another list applet.

Media section functions

You can associate a function to one of your media sections in an applet launcher. Use this function to navigate from your media to a applet or applet launcher, or URL.

Note: The Check in now text in the image represents the media section function.

1. In your instance, navigate to System Mobile > Applet Launchers.
2. Open the applet launcher containing your media section.
3. In the Body tab, find your media section in the Applet Launcher Sections list.
4. Click the title of your media section to open the media section record.
5. In the Function instance field, select the function you want to use in your media section.
6. Click Update.
Top menu functions

Use top menu function location for less frequently used actions, and on forms where you have more actions than will fit conveniently in the footer.

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 you want to add to the top menu, then select the name of the function you created.
6. Click Done.
Quick action functions

Use the quick action item location to provide your users with a shortcut to an item or action in your mobile apps. Quick actions appear under a plus icon (+) at the bottom of the applet launcher page in Android. On iOS devices, quick actions appear under the ellipsis icon (…) in the applet launcher header.

Configure quick actions on your applet launcher form in Studio. For details on configuring quick actions, see Quick actions.

Swipe functions

The swipe function location only applies 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.

Mobile properties

Mobile properties enable admins to turn on or turn off features in ServiceNow apps without upgrading the ServiceNow instance.

The ServiceNow Platform server controls the features that are available on your mobile app. The server instance is released with new features for every family release. New versions of the mobile apps are released to the Apple Store and the Google Play Store every month. These monthly releases of mobile apps usually contain bug fixes but can also contain new features. These new features can be turned on or off with mobile properties.

Mobile properties can be configured to set flags on your ServiceNow instance. These “flags” are records structured as key/value pairs in the sys_sg_properties table. They can turn on or turn off features on your mobile client apps. The ServiceNow instance sends the mobile property to mobile client apps after authentication.

Mobile properties are available starting with the San Diego release.

Application scope and mobile properties

When you create a mobile property, you select the application scope for which the property is available in the Mobile Properties New record form. To change the list of available application scopes, select the globe icon (🌍) on the instance banner and then select Application scope: application_scope.

Application scope precedence

When multiple mobile properties of the same name are defined for multiple application scopes, the system applies precedence rules to select a mobile property to use. The following scenarios use the clientRefresh mobile property as an example to show how the precedence rules work.

**Scenario 1: Global application scope always takes precedence**

If a mobile property is defined for multiple application scopes and one of those is the global application scope, then the property defined for global application scope always takes precedence. In this case, the following actions occur:

1. The system calls for the clientRefresh property on an instance.
2. The system finds three clientRefresh properties defined for this instance:

<table>
<thead>
<tr>
<th>Property name</th>
<th>Application scope</th>
<th>Updated (date and time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>clientRefresh</td>
<td>Global</td>
<td>2021-12-06 10:41:00</td>
</tr>
<tr>
<td>clientRefresh</td>
<td>Now Mobile</td>
<td>2021-12-06 10:38:41</td>
</tr>
<tr>
<td>clientRefresh</td>
<td>Agent Workspace</td>
<td>2021-12-06 10:42:06</td>
</tr>
</tbody>
</table>

3. The system chooses the clientRefresh property defined for the global application scope.

In scenario 1, the clientRefresh property defined for the global application scope takes precedence.
Scenario 2: If the mobile property isn't defined for global application scope, then the most recently updated property takes precedence

If a mobile property is defined for multiple application scopes but if there is no property with global application scope, then the system uses the most recently updated property. In this case, the following actions occur:

1. The system calls for the clientRefresh property on an instance.
2. The system finds three clientRefresh properties defined for this instance:

<table>
<thead>
<tr>
<th>Property name</th>
<th>Application scope</th>
<th>Updated (date and time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>clientRefresh</td>
<td>Now Mobile</td>
<td>2021-12-06 10:38:41</td>
</tr>
<tr>
<td>clientRefresh</td>
<td>Asset Management for mobile</td>
<td>2021-12-06 10:42:06</td>
</tr>
<tr>
<td>clientRefresh</td>
<td>Agent Workspace</td>
<td>2021-12-06 12:06:20</td>
</tr>
</tbody>
</table>

3. The system chooses the clientRefresh property that was most recently updated.

In scenario 2, the clientRefresh property defined for the agent workspace application scope was the most recently updated. All three mobile properties were updated on 2021-12-06. The property defined for the agent workspace application scope was updated at 12:06:20. That time is almost one and a half hours after the other two. The system chooses the clientRefresh property defined for the agent workspace application scope because it's the most recently updated clientRefresh property.

Turn on or turn off automatic app refresh

Use the clientRefresh mobile property on your ServiceNow instance to turn on or turn off automatically refreshing your mobile apps. You can also use the clientRefresh property to adjust the scope of refresh on your mobile app.

Before you begin
Role required: admin

About this task
The clientRefresh mobile property is enabled by default on the ServiceNow platform server. Mobile apps that are managed by San Diego and later versions of the server instance automatically refresh after an end user runs action items that change data in records. The data change is immediately visible to the user. Users don’t need to refresh their mobile apps. The New Update image no longer appears to prompt an app refresh when this feature is enabled. This feature causes no impact on mobile app performance. Mobile apps only refresh screens that have been previously visited. If you want to turn off this behavior, use the steps described in this task.

Mobile properties are configured as records in the sys_sg_properties table on the ServiceNow Platform server. The ServiceNow instance sends the mobile property to mobile apps after authentication. Then the property turns on or turns off a feature on the mobile app. The clientRefresh mobile property is configured using JSON scripts.

The clientRefresh property is a JSON object with two keys: isEnabled and refreshScope, which are described in the Mobile properties new record form.

Important: Starting with the San Diego release of the ServiceNow server, to turn off refreshing behavior, you must create a new clientRefresh JSON object and set the isEnabled key to false in the mobile property definition.
Procedure

1. Navigate to All > sys_sg_properties.list.
The Mobile Properties list appears.

2. In the Mobile Properties list, select New.
The Mobile Properties form appears.

3. On the form, fill in the fields.

### Mobile properties new record form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for the mobile property record. Enter clientRefresh to make the automatic app refresh feature configurable. Although this feature is turned on by default, you must create it to configure it. It isn't visible on the instance until you create it.</td>
</tr>
</tbody>
</table>

**Note:** The record name is case-sensitive and must be entered as clientRefresh.

<table>
<thead>
<tr>
<th>Application</th>
<th>Application scope where the mobile property is applied. To select a different application scope, select the globe icon on the instance banner. Then select Application scope:application_scope. For information about how mobile properties are affected by application scope, see Mobile properties.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Description of the mobile property. Enter a description of the property.</td>
</tr>
<tr>
<td>Type</td>
<td>Data type of the mobile property record. You must select JSON for the clientRefresh mobile property.</td>
</tr>
<tr>
<td>Value</td>
<td>The JSON code that configures this feature.</td>
</tr>
</tbody>
</table>

- To disable the feature, enter the following:

  ```json
ds{"isEnabled":false,"refreshScope":"all_tabs"}
```

- To enable the feature, enter the following:

  ```json
ds{"isEnabled":true,"refreshScope":"all_tabs"}
```

- To increase the scope of data that is refreshed, enter the following:

  ```json
ds{"isEnabled":true,"refreshScope":"all_tabs"}
```

Using "all_tabs" refreshes all screens across all navigation bar tabs after an action is run in the mobile app that changes data in a record.

- To limit the scope of data that is refreshed, enter the following:

  ```json
ds{"isEnabled":true,"refreshScope":"current_tab"}
```

Using "current_tab" refreshes just the current navigation bar tab after an action is run in the mobile app.

**Note:** Use a JSON formatter/validator to make sure that there are no syntax issues.

<p>| Active | Option to activate the mobile property. If the check box is cleared, then the mobile property isn't activated for use. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile App Config</td>
<td>Mobile app configuration for which you want to use the mobile property. This setting limits the mobile property behavior to users who have access to this mobile app configuration. Other users don’t have access to this mobile property and instead experience default behavior. To enter a value, either start typing the configuration name or select the search icon. The search dialog box lists the configurations from which you can choose. Select a configuration in the Mobile app configs dialog box.</td>
</tr>
<tr>
<td>Mobile Application</td>
<td>Mobile application to which you want to send the mobile property. This setting limits the mobile property behavior to users who have access to this mobile app. Other users don’t have access to this mobile property and instead experience default behavior. To enter a value, either start typing the application name or select the search icon. The search dialog box lists the applications from which you can choose. Select an application in the Mobile Applications dialog box.</td>
</tr>
</tbody>
</table>

4. Select Submit.

What to do next
After updating this property, the end user must pull to refresh and get the latest feature configurations.

Mobile push notifications
Mobile push notifications are configurable pushed messages that appear directly in a mobile app. These push notifications are non-actionable or actionable, and are configured by administrators to meet the needs of their users.

Non-actionable notifications are created by configuring a mobile push notification and then using the mobile deep link generator to generate links that navigate users into any screen type within a mobile app. To learn more about the mobile deep link generator, see Deep linking for mobile.

Actionable push notifications require you to perform additional steps, such as mapping functions for each push action. For more information about actionable notifications, see Configure actionable push notifications.

You can target mobile push notifications for any of the specific ServiceNow Mobile Platform mobile apps.

Push notifications structure overview

© 2022 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.
You must configure mobile push notifications in your ServiceNow instance before they can be sent to users on their mobile devices.

**Mobile push notification components**

Configure mobile push notifications to target a specific screen in one of the ServiceNow Mobile Platform applications.

This configuration determines the structure of the notification, the text the user receives, who should receive the message, and what triggers the delivery of a notification. Users tap the notification and the associated deep link then directs the user to the desired screen.

**Example push notification content and message**

![Sample Push Notification Message]

1. Push notification message content
2. Push notification message

Configure each component of the push notification because each component is dependent on the another. For example, you must configure push notification content to create a push notification message. Likewise, a push notification event requires a push notification message.

**Mobile push notification components**

<table>
<thead>
<tr>
<th>Notification component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push notification message content</td>
<td>The push notification message content contains the location users are directed to when they tap on the notification, as well as the appearance and layout of the notification. For more information on content configuration, see <a href="#">Configure push notification message content</a>. Optionally, you can display up to three additional fields in a notification. These fields are visible within the <strong>Notifications</strong> navigation bar tab in a mobile app.</td>
</tr>
</tbody>
</table>

**Note:**

Mobile UI styles are available for these additional fields. For more information around styling, see [Mobile UI styles](#).

© 2022 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.
Mobile push notification components (continued)

<table>
<thead>
<tr>
<th>Notification component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push notification message</td>
<td>The push notification message contains the string message or text that users see when they receive a notification. The message displays in the device’s notification center (based on your configuration for the user) and in the Notifications tab. For more information on message configuration, see Configure a push notification message.</td>
</tr>
<tr>
<td>Push notification event</td>
<td>The push notification event determines when the mobile push notification is sent and to whom it should be sent to. For more information on event configuration, see Configure a push notification event.</td>
</tr>
</tbody>
</table>

Configure push notification message content

Configure push notification message content and select which mobile application receives the notification.

Before you begin

To configure push notification message content, you must generate a deep link using the mobile deep link generator. For more information, see Deep linking for mobile.

Role required: admin

About this task

Complete the Push Message Content form to:

- Name the notification
- Select the mobile app that the notification is sent to
- Optionally customize the appearance of the notification message
- Determine the deep link used when users tap the notification

Push notification message content supports up to three additional fields. If these fields are used as optional text slots, the slots can display fields from the source record.

Note: The text slots are fixed on the mobile app, so you cannot change their location or size. Each slot maps to a Status, Identifier, and Description within the JSON object in the push notification message content script.

Mobile UI styles are also available for additional fields. For more information around styling, see Mobile UI styles.

Procedure

1. Navigate to All > System Notification > Push > Push Message Content.
2. Select New.
3. In the Push Message Content form, fill in the following fields:
# Push Message Content form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String value for this record.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This name is not visible in the notification, it is only a reference for use during configuration.</td>
</tr>
<tr>
<td>Push app</td>
<td>References the mobile app you want to send a notification to.</td>
</tr>
<tr>
<td></td>
<td>• Use ServiceNow mobile Application for Mobile Agent.</td>
</tr>
<tr>
<td></td>
<td>• Use ServiceNow Request Application for Now Mobile app.</td>
</tr>
<tr>
<td>Push Message Generation</td>
<td>The script block that contains the notification link. In the script block, the JSON object must return a link and a Link key must exist. The key should have a corresponding deep link value generated through the mobile deep link generator.</td>
</tr>
</tbody>
</table>

### Example

This example shows the completed form and a JSON object with the appropriate Link key and a corresponding deep link value.

```
var layoutGen = new global.NotificationLayoutFieldGenerator();
```

4. **Optional:** Add additional fields to the notification content, such as a text slot.

(Optional) In the **Push Message Generation** script block, create a layout field object for each field.

a. Instantiate a notification layout object.

   **Example**

   ```javascript
   var layoutGen = new global.NotificationLayoutFieldGenerator();
   ```

b. Use the notification layout script and include the global application scope to generate layout fields.

   To learn more about script includes, see **Script Includes**.

   **Note:** Do not modify the notification layout generator script include.

c. Create the layout field.

   Each layout field requires a Table, Sys_Id, and a Column.

   **Note:** The Table and Sys_Id values should come directly from the current record using `current.getName()` and `current.sys_id`. The Column value must exist for the desired table and match the dictionary column name exactly.

   ```javascript
   var text_01 = layoutGen.layoutField(current.getName(), current.sys_id, "priority");
   ```
d. Pass the layout fields into the original JSON object as another object.

(Optional) The mobile applications listen for the optional Layout key, which is an object that contains the desired additional fields. In the Layout object, associate each additional field with the Status, Identifier, or Description keys. The following example is available as a reference.

Example

```
(function buildJSON( /*GlideRecord*/ current, /*String*/ message, /*Object*/ attributes) {
    var layoutGen = new global.NotificationLayoutFieldGenerator();
    var text_01 = layoutGen.layoutField(current.getTableName(), current.sys_id, "priority");
    var text_02 = layoutGen.layoutField(current.getTableName(), current.sys_id, "state");
    var text_03 = layoutGen.layoutField(current.getTableName(), current.sys_id, "short_description");

    var deepLinkGenerator = new global.MobileDeepLinkGenerator("Request");
    var link =
        deepLinkGenerator.getFormScreenLink("b71040b1533a33007bc2ddeff7b1221", current.getTableName(), current.getValue("sys_id"));

    var json = {
        "Link": link,
        "Layout": {
            "Status": text_01,
            "Identifier": text_02,
            "Description": text_03
        }
    };

    return json;
})(current, message, attributes);
```

Results
You created a push notification message content record that contains a deep link.

What to do next
After configuring push notification message content, it must be paired with a push notification message record. For more information on configuring a push notification message, see Configure a push notification message.

Configure a push notification message
Configure the message displayed to users when they receive their push notification.
Before you begin
To create a push notification message, you must first configure the push notification message content. For more information, see Configure push notification message content.
Role required: admin

Procedure
1. Navigate to All > System Notification > Push > Push Messages.
2. Click New.
3. In the Push Message form, fill in the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The string value for this record.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>This name is not visible in the notification, it is only a reference for use during configuration.</td>
</tr>
<tr>
<td>Push app</td>
<td>References the mobile app you want to send a notification to.</td>
</tr>
<tr>
<td>• Use ServiceNow mobile Application for Mobile Agent.</td>
<td></td>
</tr>
<tr>
<td>• Use ServiceNow Request Application for Now Mobile app.</td>
<td></td>
</tr>
<tr>
<td>Push message</td>
<td>References the push message content record.</td>
</tr>
<tr>
<td>Table</td>
<td>Table selector to choose the table the message originates from. For example, if you want to send notifications when new incidents are created, select the Incident [incident] table.</td>
</tr>
<tr>
<td>Message</td>
<td>The string value used as the message. This field can contain a simple string, but also supports displaying field values from the source table.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>To use a field value, wrap the desired column name with ${}, such as &quot;Change ${number} has been assigned to you.&quot;</td>
</tr>
</tbody>
</table>

4. Click Submit to create the record.

Results
You configured a push notification message and paired it with a push notification message content record.

What to do next
To complete the notification, an event is required to determine when the push notification is sent and who receives it. For more information, see Configure a push notification event.

Configure a push notification event
Configure when a push notification event is triggered and sends a mobile push notification to users.

Before you begin
To create a push notification event and complete the mobile push notification, a push notification message is required. For more information, see Configure a push notification message.
Role required: admin

Procedure
1. Navigate to All > System Notification > Push > Create Push Notification.
2. In the Push Notification form, complete the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The string value for this record. Note: This name is visible to users when they are managing their notification preferences in their personal user settings.</td>
</tr>
<tr>
<td>Table</td>
<td>Table selector to choose the table the message originates from. For example, if you want to send notifications when new incidents are created, select the Incident [incident] table.</td>
</tr>
<tr>
<td>Category</td>
<td>Select where to store notification records if a user manages their platform notification preferences in their personal user settings.</td>
</tr>
</tbody>
</table>

3. In the When to send tab, complete the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inserted</td>
<td>When set to true, the notification is sent when a record is inserted into the specified table.</td>
</tr>
<tr>
<td>Updated</td>
<td>When set to true, the notification is sent when a record is updated.</td>
</tr>
<tr>
<td>Conditions</td>
<td>Condition builder for the specified table. When conditions are added, they must be set to true for the notification to send. These conditions work with the Inserted and Updated fields. Note: If the Inserted field is set to true, the condition must also be set to true for the notification to send.</td>
</tr>
</tbody>
</table>

4. In the Who will receive tab, complete the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>List of users from the User [sys_user] table who should receive this notification.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Users/Groups in fields</td>
<td>Users or Groups from one of the fields on the specified table who should receive this notification</td>
</tr>
<tr>
<td>Groups</td>
<td>List of groups from the Group [sys_user_group] table who should receive this notification.</td>
</tr>
</tbody>
</table>

5. In the **What to send** tab, select the desired **Push Message**.

6. Click **Submit** to save the configuration record.

**Results**

A push notification event is configured and paired with the push notification message. The mobile push notification is available to send to users.

**What to do next**

You can add a push registration so that mobile users can receive this notification without requiring them to opt in. For more information, see Register the standard push notification.

**Register the standard push notification**

Register your standard push notification in a push application to use it in your mobile apps.

**Before you begin**

Role required: admin

The push application handles sending notifications to mobile devices. Add your new notification to the **ServiceNow Mobile Application** push application so your mobile app users start receiving this notification.

**Procedure**

1. Switch to the **Global** scope.

2. Navigate to **System Notification > Push Application**.

3. Open the record for the push application you want to modify.

   • Open **ServiceNow Mobile Application** for the Mobile Agent app.
   
   • Open **ServiceNow Request Application** for the Now Mobile app.
   
   • Open **ServiceNow Onboarding Mobile Application** for the Mobile Onboarding app.

   **Note:** If you are using a branded mobile application, select the record for the branded app. For example, select **ServiceNow Onboarding Intune Mobile Application** if you are using the Microsoft Intune branded application.

   **Note:** The Mobile Onboarding app is being deprecated!

   With the Now Platform Rome release in September 2021, we started phasing out support for the Mobile Onboarding. Customers may no longer activate it, and we are not offering enhancements or non-critical bug fixes. Mobile onboarding features are available in the **Now Mobile app for HR Service Delivery.**

4. Right-click the record header and choose **Configure > Related Lists** from the menu.

5. Add **Default Push Registration > Push App** to the list on the right.

6. In the **Push Default Registrations** related list, click the **New** button to create a new **Push Default Registration** record.
7. In the Notification field, select the notification you created in the previous steps.

8. Click Submit.
   Your push message is rendered on mobile devices that it is registered to. To see how push notifications appear to end users, see Mobile push notifications.

   Note: Changes to notification settings are not visible to your users until they refresh their meta-data. Users can refresh meta-data by refreshing any applet launcher in the app

Configure push applications for iOS branded apps

Install an iOS push certificate to use push notifications on your branded ServiceNow mobile apps for iOS.

Before you begin
Role required: admin
To use push notifications on your branded ServiceNow mobile apps for iOS, you must upload your Apple certificates p12 file, and enter your key store password into your instance.

   Note: These steps are not necessary if you are not using a custom branded mobile apps. For more detail on custom branding, see Request a branded mobile app.

Procedure
1. Navigate to All > System Notification > Push > Push Application.
2. Open the record for the push application you want to modify.
   - Open ServiceNow Mobile Application for the Mobile Agent app.
   - Open ServiceNow Request Application for the Now Mobile app.
   - Open ServiceNow Onboarding Mobile Application for the Mobile Onboarding.

   Note: The Mobile Onboarding app is being deprecated!
   With the Now Platform Rome release in September 2021, we started phasing out support for the Mobile Onboarding. Customers may no longer activate it, and we are not offering enhancements or non-critical bug fixes. Mobile onboarding features are available in the Now Mobile app for HR Service Delivery.

   Note: If you are using a branded mobile application, select the record for the branded app. For example, select ServiceNow Onboarding Intune Mobile Application if you are using the Microsoft Intune branded application.
3. In the Push field, change the value from REST API to Direct.
4. In the Apple tab, click the reference icon ( ) next to the Certificate field.
5. In the X.509 Certificates list pop-up, click New.
6. In the X.509 Certificates form, fill in the fields.

   X.509 Certificate form

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name for your certificate.</td>
</tr>
<tr>
<td>Field</td>
<td>Value</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Notify on expiration</td>
<td>User to receive a notification when the certificate expires.</td>
</tr>
<tr>
<td>Type</td>
<td>Certificate type. Enter <strong>PKCS12 Key Store</strong> in this field.</td>
</tr>
<tr>
<td>Key store password</td>
<td>Your Apple key store password.</td>
</tr>
<tr>
<td>Warn in days to expire</td>
<td>Number of days before certificate expiration to receive an expiration warning.</td>
</tr>
<tr>
<td>Short Description</td>
<td>Description for your certificate.</td>
</tr>
</tbody>
</table>

7. Click the attachment icon (_attach).  
9. Click **Save**.  
10. **Optional**: Re-open your X.509 Certificates record and click the **Validate Stores/Certificates** link at the bottom of the form to validate your certificates and stores.  
11. On the push application form, click **Update**.

### Configure push applications for Android branded apps

Enter your Google Firebase Cloud Messaging legacy server key in your push application record to use push notifications on your branded ServiceNow mobile apps for Google Android.

**Before you begin**

Role required: admin

To use push notifications on your branded ServiceNow mobile apps for Android, you must enter your Google Firebase Cloud Messaging legacy server key in your push application record.

⚠️ **Note**: These steps are only necessary if you are using a custom branded mobile apps.  
For more detail on custom branding, see [Request a branded mobile app](#).

### Procedure

1. Navigate to **All > System Notification > Push > Push Application**.  
2. Open the record labeled **ServiceNow Mobile Application**.  
3. In the **Push** field, change the value from **REST API** to **Direct**.  
4. In the **Google** tab, enter your Google Firebase Cloud Messaging server key in the **API Key** field.  
5. Click **Update**.

### Configure actionable push notifications

Include actions with your push notifications. Users can perform push notification actions without opening the app.
Up to three actions can be associated with a push notification. These actions must refer to an existing mobile function. The following function types of actions are supported:

- Action item
- Navigation
- URL
- Chat launcher

### Creating actionable push notifications

Create actionable push notifications using the following process:

**Create a push notification**

Create a push notification your users will see on their mobile devices.

**Add a push action category**

Select a push category to determine what actions your users can take in a notification.

**Create functions for each push action**

Create mobile function for each function in the selected action category. These actions perform tasks on your instance based on what the user selects in the notification.

**Map functions with push actions on the actionable push notification**

Associate functions to the actions in your notification so the instance uses the correct function for each action.

**Create push message content**

Create a record to determine what information the notification displays to your users.

**Create a standard notification**

Create a standard notification your instance using the platform notifications.

**Create an actionable push notification**

Create a push notification your users will see on their mobile devices.

### Before you begin

**Role required:** admin

### Procedure

1. Navigate to **All > System Mobile > Mobile Push Notifications > Push Notifications**.
2. Click **New**.
3. In the push notification form, fill in the fields.
Push notification form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of your push notification.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This name does not appear to users viewing the notification.</td>
</tr>
<tr>
<td>Application</td>
<td>The scoped application for your notification. This field automatically uses the current application scope.</td>
</tr>
<tr>
<td>Actionable</td>
<td>Whether the push notification is actionable. Click to enable this field.</td>
</tr>
<tr>
<td>Screen</td>
<td>An optional screen associated with your notification. The notification uses this screen. This screen displays to a user when they tap on the notification.</td>
</tr>
<tr>
<td>Category</td>
<td>Category of your notification. This field is visible only when the Actionable field is enabled. This category defines which actions are available in the push notification. You will fill in this field in later steps.</td>
</tr>
</tbody>
</table>

4. Click **Save**.
   Your notification record is created. After saving, the **Push Action Instances** related list appears on the form.

**Add a push action category**
Select a push category to determine what actions your users can take in a notification. This category defines which actions your users can take when viewing a notification.

**Before you begin**
Role required: admin

**Procedure**
1. In the **Push Notification** record, select a category in the **Category** field. There are 12 base system categories to choose from on your instance.

**Push action categories**

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accept-reject-back</td>
<td>Displays <strong>Accept</strong> and <strong>Reject</strong> buttons. Users can select an option without opening the app.</td>
</tr>
<tr>
<td>Accept-Reject-fore</td>
<td>Displays <strong>Accept</strong> and <strong>Reject</strong> buttons. The app opens when the user selects an action.</td>
</tr>
<tr>
<td>accept-rejectwcomments-back</td>
<td>Displays <strong>Accept</strong> and <strong>Reject</strong> buttons. Users see a text field to enter comments when they select the <strong>Reject</strong> option. Users can select an option without opening the app.</td>
</tr>
<tr>
<td>Action</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ack-escalate-ignore-back</td>
<td>Displays <strong>Acknowledge</strong>, <strong>Escalate</strong> and <strong>Ignore</strong> buttons. Users can select an option without opening the app.</td>
</tr>
<tr>
<td>approve-reject-back</td>
<td>Displays <strong>Approve</strong> and <strong>Reject</strong> buttons. Users can select an option without opening the app.</td>
</tr>
<tr>
<td>approve-rejectwcomments-back</td>
<td>Displays <strong>Approve</strong> and <strong>Reject</strong> buttons. Users see a text field to enter comments when they select the <strong>Reject</strong> option. Users can select an option without opening the app.</td>
</tr>
<tr>
<td>confirm-reschedule-cancel-back</td>
<td>Displays <strong>Confirm</strong>, <strong>Reschedule</strong> and <strong>Cancel</strong> buttons. Users can select an option without opening the app.</td>
</tr>
<tr>
<td>joinbridge-fore</td>
<td>Displays a <strong>Join Meeting</strong> button. Tapping this button opens the app.</td>
</tr>
<tr>
<td>PromoteWComments-RejectWComments-fore</td>
<td>Displays <strong>Promote</strong> and <strong>Reject</strong> buttons. Users see a text field to enter comments when they select either option.</td>
</tr>
<tr>
<td>queue-rejoin-openincident-back</td>
<td>Displays <strong>Online Check-in</strong> and <strong>Open Incident</strong> buttons. Users can select an option without opening the app.</td>
</tr>
<tr>
<td>queue-stay-queueleave-back</td>
<td>Displays <strong>Stay in Queue</strong> and <strong>Leave Queue</strong> buttons. Users can select an option without opening the app.</td>
</tr>
<tr>
<td>yes-no-back</td>
<td>Displays <strong>Yes</strong> and <strong>No</strong> buttons. Users can select an option without opening the app.</td>
</tr>
</tbody>
</table>

If the base system push action categories do not meet your requirements, you can create new action categories for use in your push notifications. See the section below for details on this process.

Your push notification displays buttons to your user in addition to the message content. In the next steps, you create functions that determine what happens when a user taps these actions.

2. Right-click the form header and click **Save**.

**Create a push action**

Create push actions to suit your needs, when the base system actions do not meet your requirements.

**Before you begin**

Role required: admin
Procedure

1. In the web-based UI, enter `sys_sg_push_action.list` in the filter navigator to open the push action categories list.

   ![ServiceNow Service Management interface](image)

2. In the push action categories list, click **New**.

3. On the push action form, fill in the fields.

   **Push action fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Label for the push action. This is the text your users will see in the notification for this action.</td>
</tr>
<tr>
<td>Application</td>
<td>The scoped application for this record. This field automatically populates.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the push action</td>
</tr>
<tr>
<td>Foreground</td>
<td>When enabled, the app will open when your users select this action.</td>
</tr>
<tr>
<td>Response Type</td>
<td>Select <strong>Simple</strong> or <strong>Text response</strong>. Actions using <strong>Text response</strong> prompt your users to enter text before completing the action.</td>
</tr>
<tr>
<td>Placeholder Text</td>
<td>Text displays as a placeholder text to your users when they initiate the actionable push.</td>
</tr>
</tbody>
</table>

4. Click **Save**.

Create functions for each push action

Create mobile function for each function in the selected action category. These actions perform tasks on your instance based on what the user selects in the notification.

**Before you begin**

Role required: admin

**Procedure**

1. Navigate to **All > System Mobile > Functions**.

2. Click **New** to create a new function record.

3. Create a new function for one of your push actions. When selecting a function type in the **Type** field, you must select one of the type supported for actionable push notifications.
• Action item
• Navigation
• URL
• Chat launcher

For information on function types, and detailed steps for creating functions, see Mobile functions.

4. Create additional functions for each push action.
   For example, if you have selected ack-escalate-ignore-back as your push notification category, your notification displays Acknowledge, Escalate and Ignore buttons. You need to create a function to determine the behavior of each of these buttons.

Results
You have functions to perform actions for each of the buttons that display in your push notification. In the next steps, you will associate these functions to the notification actions so your functions trigger when users tap the buttons.

Map functions with push actions on the actionable push notification
Associate functions to the actions in your notification so the instance uses the correct function for each action.

Before you begin
Role required: admin

Procedure
1. System Mobile > Mobile Push Notifications > Push Notifications, and open the notification you created in previous steps.
2. In the Push Action Instances related list, click New.
3. In the Push action instance form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push Notification</td>
<td>Push notification associated with this push action instance. This is automatically filled in with your push notification.</td>
</tr>
<tr>
<td>Application</td>
<td>Scoped application associated with this record. This is automatically filled in with the current application.</td>
</tr>
<tr>
<td>Push Action</td>
<td>Push action from your push notification record. Select one of the available actions.</td>
</tr>
<tr>
<td>Button</td>
<td>Function to associate with the push action. Select an action you created in previous steps. This action will trigger when a user uses the action listed in the Push Action field.</td>
</tr>
</tbody>
</table>

4. Optional: Add a push parameter to your push action instance.
   If your push action receives a value from your user, you need to configure a push parameter to pass this value to your record. For example, if you create a reject action that
requires your user to enter a reason for the rejection, the action must have a parameter to pass that text value to your approval record.

**a.** Verify that your push action requires a parameter. In the push action instance form, click the preview icon (①) next to the **Push action** field. If the push action has a value of Text Response in the **Response type** field, you must create a push parameter.

**b.** Find the button parameter by clicking the preview icon (①) next to the **Button** field.

**c.** In the function pop up click **Open record** to open the function form.

**d.** In the function form, note the parameter that appears in the **UI Parameters** related list at the bottom on the form. You will need this value in later steps.

**e.** Click the back button to return to the push action instance form.

**f.** In the **Push Parameter Values** related list, click **New**.
g. In the push parameter value form, select your parameter in the Parameter field. This is the parameter you noted from the function form in step d.

h. Click Submit to save your push parameter value.

5. In the push action instance form, click Submit.

Example:
This example notification uses accept-rejectwcomments-back in the Category field. If you click the preview icon (①), you can see that this category uses two actions: Accept and Reject.

In the push action instance shown here, the Approve push action is selected in the Push Action field, and the Approve REQ /w comments function is selected in the Button field. The instance executes this function when a user taps the Approve button in their notification.

Create push message content
Create a record to determine what information the notification displays to your users.

Before you begin
Role required: admin
Procedure

1. In your push notification record, click **Create Push Message Content** in the **Related Links** section.
   
   If you are not already in this record, Navigate to **System Mobile > Mobile Push Notifications > Push Notifications**, and open the notification you created in previous steps.

2. In the **Push Notification Message Content** form, fill in the fields.

   **Push Notification Message Content form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of your push message content record. This name is not visible to your users.</td>
</tr>
<tr>
<td>Application</td>
<td>Scoped application associated with this record. This is automatically filled in with the current application.</td>
</tr>
<tr>
<td>Push app</td>
<td>Mobile app that uses your push notification.</td>
</tr>
<tr>
<td>Push Message Generation</td>
<td>Script that generates the push notification. When you create message content using these steps, this script is automatically generated.</td>
</tr>
</tbody>
</table>

3. Click **Submit**.

**Create a standard notification**

Create a standard notification on your instance using the platform notifications.

**Before you begin**

Role required: admin

**Procedure**

1. Navigate to **All > System Notification > Push > Create Push Notification**.

2. In the notification form, fill in the fields.

   **Push notification form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of your Notification. This name is not visible to your users.</td>
</tr>
<tr>
<td>Active</td>
<td>Whether the notification is active. The instance does not send inactive notifications to your users.</td>
</tr>
<tr>
<td>Table</td>
<td>The table containing the records relating to your notification. For example, notifications about new incidents would use the Incident [incident] table.</td>
</tr>
<tr>
<td>Push Message Only</td>
<td>Whether the notification is a push message only. This field is read-only by default.</td>
</tr>
<tr>
<td>Category</td>
<td>The category for your notification. This field value is <strong>Uncategorized</strong> by default.</td>
</tr>
</tbody>
</table>
3. Right-click the form header and select **Save**.

4. Configure the push message for your standard notification.

   a. Click to display the **What to Send** tab.

   b. Click the lock icon (🔒) next to the **Push Messages** field.

   c. Click the reference icon (🔍) to display the **Push Notification Messages** list.

   d. Click **New**.

   e. In the push notification message 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 of your push notification message record. This name is not visible to your users.</td>
</tr>
<tr>
<td>Application</td>
<td>Scoped application associated with this record. This field defaults to the current application.</td>
</tr>
<tr>
<td>Push App</td>
<td>Mobile app that uses your push notification.</td>
</tr>
<tr>
<td>Table</td>
<td>The table containing the records relating to your notification. For example, a notification about new incidents would use the Incident [incident] table.</td>
</tr>
<tr>
<td>Push Message Content</td>
<td>Select the record for the push message content you created in the previous section.</td>
</tr>
<tr>
<td>Message</td>
<td>Text the user sees in the push notification. For example, An incident has been assigned to you.</td>
</tr>
</tbody>
</table>

   f. Click **Submit**.

5. In the **Notification** form, click **Update**.

**Configure notifications for mobile activity stream mentions**

Configure your instance to send an email notification or a push notification when a user mentions a contact’s name within a mobile activity stream record.

**Configure push notifications for mobile activity stream mentions**

Enable users to alert their contacts, through the use push notifications, that they are mentioned in a thread. The user enters the @ character followed by the contact’s name in a record’s activity stream, for their contact to be alerted.
Before you begin
Role required: admin

Procedure

1. Create the push notification message content, which defines where you are pushing the notification to.

   a. Navigate to **System Notification > Push > Push Message Content** in the filter navigator.
   
   b. In the Push Notification Message Content form, select **New**.
   
   c. On the form, fill in the fields.

   **Push Notification Message Content form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of your push message content record. This name is not visible to your users.</td>
</tr>
<tr>
<td>Application</td>
<td>Scoped application associated with this record. This field defaults to the current application.</td>
</tr>
<tr>
<td>Push app</td>
<td>Mobile app that uses your push notification.</td>
</tr>
<tr>
<td></td>
<td>• Enter <strong>ServiceNow Mobile Application</strong> for the Mobile Agent app.</td>
</tr>
<tr>
<td></td>
<td>• Enter <strong>ServiceNow Request Application</strong> for the Now Mobile app.</td>
</tr>
</tbody>
</table>

   d. In the **Push Message Generation** field, copy the following sample JSON code and paste it in the field.

   ```javascript
   (function buildJSON(/*GlideRecord*/ current, /*String*/ message, /*Object*/ attributes) {
       var layoutFieldGenerator = new global.NotificationLayoutFieldGenerator();
       var json = {};
       var deepLinkGenerator = new global.MobileDeepLinkGenerator("<client type>");

       var link = deepLinkGenerator.getFormScreenLink('<form_screen_sys_id>',
           current.getValue("table"), current.getValue("document"));

       json = {
           "aps" : {
               "sound" : "default"
           },
           "Link": link
       };
   }
   ) (current, message, attributes);
   ```

   Customize the values in the code.
i. Locate the sys_id of the form screen by right-clicking on the screen name and selecting Copy sys_id.

ii. Replace the &lt;form_screen_sys_id&gt; portion of the code with the sys_id.

iii. Define a client type in the variable global.MobileDeepLinkGenerator Possible values are:
   • agent
   • request
   • onboarding

e. Select Submit.

2. Create a message for the user's contacts to receive when they are mentioned in a record's activity stream.

   a. Navigate to System Notification &gt; Push &gt; Push Messages.

   b. In the Push Notification Message form, select 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 of your push message content record. This name is not visible to your users.</td>
</tr>
<tr>
<td>Application</td>
<td>Scoped application associated with this record. This field defaults to the current application.</td>
</tr>
<tr>
<td>Push app</td>
<td>Mobile app that uses your push notification.</td>
</tr>
<tr>
<td>Note:</td>
<td>Make sure to enter the same mobile app entered in the Push app field in the Push Notification Message Content form.</td>
</tr>
<tr>
<td>Push Message Content</td>
<td>Select the push message content you created.</td>
</tr>
<tr>
<td>Table</td>
<td>Table containing the records relating to your notification. Select live_notification.</td>
</tr>
<tr>
<td>Message</td>
<td>Message that appears to the user's contacts, informing them they are mentioned.</td>
</tr>
</tbody>
</table>

d. Select Submit.

3. Create the push notification and define when to send it, who receives it, and what it contains.

b. In the Push Notification view form, select New.

c. 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 your push notification. This name is not visible to your users.</td>
</tr>
<tr>
<td>Active</td>
<td>Option for determining whether the push notification is active. Select this option.</td>
</tr>
<tr>
<td>Table</td>
<td>Table containing the records relating to your notification. Select the same table as the push notification view form, live_notification.</td>
</tr>
<tr>
<td>Push Message Only</td>
<td>Option that determines whether to send a push notification to a mobile device. This field is automatically selected and is read-only by default.</td>
</tr>
<tr>
<td>Category</td>
<td>Category to which this notification belongs. A category identifies and groups related notifications. This category defines which actions are available in the push notification.</td>
</tr>
</tbody>
</table>

d. Select the When to send tab and then select Inserted.

e. Select the Who will receive tab and then select the lock icon (padlock) in the Users/Groups in fields.

f. Move User to the selected list.

Note: If the notification is addressed to a user with an inactive record in the User [sys_user] table, the notification is not sent.

g. Select the What to send tab and select the lock icon (padlock) next to the Push Messages field.

h. Select the reference lookup icon (search) to open the Push Notification Messages page.

i. Select the push notification message you created. The push notification is listed as one of the available messages in the What to send tab.

j. Select Submit.

4. Enable your mobile app settings for notifications.
a. Select the Settings icon (⚙️) and select the **Notifications** tab.

b. Verify that **Allow Notifications** is enabled.

c. Verify that the mobile app you selected to receive notifications is enabled by noting the setting of either the **ServiceNow Mobile Application** or **ServiceNow Request application** field.

**Customize email notifications for mobile activity stream mentions**

Enable users to alert their contacts by email that they are mentioned in a thread. The user enters the @ character followed by the contact's name in a record's activity stream, for their contact to be alerted. This feature is enabled by default.

**Before you begin**

Role required: admin

**About this task**

Configure your instance to send an email notification when a user mentions a contact within the activity stream. You can either use the default notification record or create a new notification record. For information about creating a new notification record, see Create an email notification.

**Procedure**

1. Navigate to **All > System Notification > Email > Notifications** in the filter navigator.

2. Search for the notification record named **Activity Stream @Mention Email**, then select the record to open it.

3. Verify that **Active** is selected.

4. **Optional:** Customize the email notification fields.

5. Select **Update**.

6. Select the Settings icon (⚙️) and select the **Notifications** tab.

7. Verify that **Allow Notifications** is enabled.

**Mobile URLs**

Create mobile URLs to direct users to areas, screens, and features within specified mobile apps. Using mobile deep linking, these re-directional links can be placed within emails, texts, and push notifications.

**Deep linking for mobile**

Mobile deep links are linking formats that direct users to specific content and locations within a mobile app. These links can be placed in communication items such as email notifications, text messages, and push notifications.

You can activate mobile deep link URLs to navigate users into any screen type within a mobile app. The deep link consists of a specific structure, which points to a specific location within a defined mobile app. These links can then be placed wherever desired. Most commonly, deep links are used within push and email notifications.
• For push notifications, deep link URLs are used to navigate the end user to a defined screen after tapping on the notification itself. For information on configuring a push notification, see Mobile push notifications.

• For email notifications, deep link URLs can be placed within the email body. The end user taps on this link to launch the appropriate mobile app and automatically navigate to the desired screen. For information on configuring an email notification, see Create an email notification.

Mobile deep links are activated through the mobile deep link generator. The mobile deep link generator operates as a platform script include, within the global application scope. For additional information see, Script includes.

Accessing the mobile deep link generator
You can view the script include configuration by entering script includes in the web-based UI filter navigator and search for the name MobileDeepLinkGenerator.

ℹ️ Note: The mobile deep link generator script include must not be modified.

Mobile deep link structure
Familiarize yourself with how to construct and use mobile deep links.

Mobile deep links essentially consist of three components. The name of the variable, a reference to a mobile app, and the public method used to pull the mobile deep link. The following mobile deep link has a variable with the name deepLinkGen, where a specified screen opens in the Mobile Agent.

```javascript
var deepLinkGen = new global.MobileDeepLinkGenerator('Agent');
return deepLinkGen.getScreenLink('73f3d135532033002d96ddee77b1212');
```

Variable name
You can give any variable name for the mobile deep link, although it should be identifiable for future reference.

Mobile app call
Enter one of the mobile apps, either Request, Agent, or Onboarding, that the mobile deep link points to. For example, to generate a link using the mobile deep link generator that points to the Now Mobile app, use the following code.

```javascript
new global.MobileDeepLinkGenerator('Request');
```

Public method
Use one of the public methods that can be used to create a mobile deep link. Reference the table for an explanation of each method, and the reason for its use.

<table>
<thead>
<tr>
<th>Method</th>
<th>Reason for use</th>
<th>Method parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>getScreenLink(documentId, uiParams)</td>
<td>getScreenLink is used to link into any list screens, including both parameterized and non-parameterized types. Can also be used for any non-parameterized screen type. An optional UI parameter.</td>
<td>documentId (required): sys_id of desired screen, uiParams (optional): Parameter of a data item</td>
</tr>
<tr>
<td>Method</td>
<td>Reason for use</td>
<td>Method parameters</td>
</tr>
<tr>
<td>--------</td>
<td>----------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>getFormScreenLink(formScreenId, tableName, recordSysId)</td>
<td>can be passed into this method.</td>
<td>formScreenId (required): sys_id of required form screen&lt;br&gt;tableName (required): table name of required form screen&lt;br&gt;recordSysId (required): sys_id of record from tableName</td>
</tr>
<tr>
<td>getUniversalLink(url)</td>
<td>getUniversalLink generates a link into either a list, form, or browser. A screen is generated based on the content of this URL. See Universal linking for mobile.</td>
<td>url (required): a link to a resource on an instance</td>
</tr>
<tr>
<td>getPrefillLink(forceLocalLogin)</td>
<td>getPrefillLink is useful for first-time ServiceNow mobile users. This link format pre-fills the instance link for users who have not yet added an instance to their mobile app. This method is the default authentication experience. This method does not generate a link to a specific screen.</td>
<td>forceLocalLogin (required): Enter true for the system to pre-fill the instance login details of the user. Enter false, for the user to manually enter login details.</td>
</tr>
<tr>
<td>getSSOPrefillLink(sys_id)</td>
<td>getSSOPrefillLink generates a link that uses the identity provider (IdP) that is identified by the sys_id parameter. This link can be used to authenticate end users that use different IdPs. For example, pre-hires might use Microsoft Azure as their IdP, and on-boarded employees might use Okta as their IdP. A common use of these SSO (single sign-on) pre-filled links is to embed them in</td>
<td>sys_id (required): Enter the sys_id of the IdP you want to use from the Identity Providers [sso_properties] table. To get the sys_id of the IdP:</td>
</tr>
</tbody>
</table>
Universal linking for mobile

Mobile universal linking is a method to redirect your users from using the ServiceNow platform web experience on their mobile device. Instead it promotes using the faster, more responsive, and location-specific ServiceNow Mobile Platform experience via mobile apps. Mobile universal links enable you to drive users to specific screens within a mobile app, that they would not be able to access via the platform web experience.

Universal linking works by presenting the user with a web banner at the top of designated screens, while they work within a ServiceNow platform web experience. For the user to proceed, they must select View in the banner, where the following occurs:

- If the user has already installed the required ServiceNow mobile app and is logged in to the instance - then the user is directed to a defined location within the specific mobile app.

- If the user has already installed the required ServiceNow mobile app and is not logged in to the instance - then the user is required to log in to the instance and is then directed to a defined location within the specified mobile app.

- If the user has not installed the required ServiceNow mobile app - they are directed to either the Apple Store or Google Play Store, and must download the ServiceNow app. Once the app is downloaded, the user must log in to the instance and is then directed to a defined location within the newly installed app.

After the user taps View in the banner, direct them to any record screen, form screen, or browser screen within a specified mobile app. For form and record screens, you can direct the user to a specific location based on a defined table.

<table>
<thead>
<tr>
<th>Method</th>
<th>Reason for use</th>
<th>Method parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>emails sent to employees. These emails contain a link that can be used to log in securely to a ServiceNow instance using their mobile apps.</td>
<td>1. Navigate to All &gt; Multi-Provider SSO &gt; Identity Providers. 2. Right-click an identity provider record and select Copy sys_id.</td>
</tr>
</tbody>
</table>

Important: You must activate the Multi-Provider SSO plugin and have an active IdP to use this method. For more information, see Activate Multi-Provider SSO plugin.
Enable mobile universal linking

Enable the universal linking option if you want a web banner displayed on specified pages of the ServiceNow platform website. By default the mobile universal linking option is not available in the base system.

Before you begin
Role required: admin

Procedure
1. In the web-based UI, enter sys_properties.list in the filter navigator, to open a list of system properties.
2. In the Name field, search for the term *universal.
3. Select the property glide.sg.universal_links.enabled.
4. In the universal links system property table, set the Value field to true.
5. Right-click in the header and select Save.
6. In the web-based UI, enter `sys_sg_mobile_application.list` in the filter navigator to open a list of mobile applications in the `sys_sg_mobile_application` table.

7. Select the Request record, and open the Deep Link Configuration tab.

8. Select the Use firebase links check box to enable it.

9. Right-click in the header and select Save.

10. Repeat Steps 6 through 9 for the Agent record.

**What to do next**
After you have enabled the mobile universal linking feature, the following configuration can be performed:

- Define specific ServiceNow platform website pages to display the web banner, where the user does not have a specific mobile app installed. See, Mobile universal linking for supported URLs.

**Enable expansion of universal linking to all web pages**
Enable the expanded universal linking option to display the web banner on each page of the ServiceNow platform website, when the user does not have the appropriate mobile app installed. By default this mobile universal linking option is not available in the base system.

**Before you begin**
The mobile universal linking feature must be enabled, see Enable mobile universal linking.

Role required: admin

**Procedure**

1. In the web-based UI, enter `sys_properties.list` in the filter navigator, to open a list of system properties.

2. In the Name field search for the term *universal.*

3. Click the property `glide.sg.universal_links.open_unsupported_links`.

4. In the universal links system property table, set the Value field to True.

5. Right-click in the header and select Save.

6. Optional: Select the mobile app you want to redirect the user to, each time they tap the link in the banner. The default is the Mobile Agent.
   a. Return to the system properties table.
   b. Click the property `glide.sg.universal_links.unsupported_link_default_native_client`.
   c. In the universal links unsupported links system property table, set the Value field to either agent, requester, or onboarding.

7. Right-click in the header and select Save.

**Mobile universal linking for supported URLs**

Use mobile universal linking to display a web banner on specified platform web pages to encourage users to work within a ServiceNow mobile app. Configure the supported URLs that facilitate this functionality to ensure that the banners are displayed on the correct web pages. These URLs also direct users to a defined location within the appropriate mobile app.

Each time a user accesses a web page using their mobile web browser, the instance checks the URL against all the records in the path segment table and the supported URLs table. When there's a match, the instance displays a web banner. After the user taps on the
banner, the instance directs the user to the appropriate screen on a designated mobile app. This screen displays the selected data. When configuring a URL record for universal linking, the path, query keys, and query values of the URL must be an exact match of the web page.

**Structural layout of URLs used for mobile universal linking**

Supported URLs can comprise of elements like host, path, query keys, and query values. These URLs are created in the universal link path segment table and the universal link supported URL table, as described in Create path segment records and supported URL records for universal linking. You can make your URL as general or specific as you require. A more general the URL definition means that the banner is displayed over a larger number of web pages.

ℹ️ **Note:** When creating supported URLs, you must know what path segment, query keys, and query values you want the system to look for.

An example URL structure is `https://<baseURL>/sample_path.do/query_key_01=XYZ/query_key01=value_ABC`. The table describes the parts to create in a path segment record and a supported URL record.

<table>
<thead>
<tr>
<th>URL part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sample_path</td>
<td>Due to its generic nature, the path segment defines the category of the URL and is used to quickly filter out any unmatched records. The path segment record uses regular expressions to match URL texts.</td>
</tr>
<tr>
<td>query_key</td>
<td>Query keys are an extension of the URL that help define specific content.</td>
</tr>
<tr>
<td>value</td>
<td>Values can be added to the query keys, to further streamline the web pages you want to match.</td>
</tr>
</tbody>
</table>

Create path segment records and supported URL records for universal linking

Create path segment records and supported URLs records for universal linking to display a mobile banner. Users see this banner when they land on specific platform web pages. Within each supported URL, you define the page where the banner appears, the mobile app and the screen type the user views, and the data displayed within the screen.

**Before you begin**

Role required: admin

**Procedure**

1. Create a path segment record.

   a. In the web-based UI, navigate to **System Mobile > Mobile Universal Linking > Path Segments**.

   b. Select **New** to create a universal link path segment.

   c. On the form, fill in the fields.

   **Universal Link Path Segment form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Title for the universal link path segment.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path segment</td>
<td>Regular expression of the string that defines the category path of the URL. The system evaluates the regular expression string and then performs the matching process on the supported URLs.</td>
</tr>
</tbody>
</table>

**d.** Select **Submit**.

2. Create a universal link supported URL.

   a. In the web-based UI, navigate to **System Mobile > Mobile Universal Liking > Supported URLs**.

   b. Select **New** to create a universal link supported URL.

   c. On the form, fill in the fields.

**Universal Link Supported URL form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Title for the universal link supported URL.</td>
</tr>
<tr>
<td>Mobile app</td>
<td>ServiceNow mobile app the link opens in. Options are Agent, Onboarding, or Request (Now Mobile).</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The selection here overrides the selection made in the <strong>Mobile app</strong> field listed in the topic <strong>Mapping specific mobile screens for universal linking</strong>.</td>
</tr>
<tr>
<td>Active</td>
<td>Option for enabling the activation of this record.</td>
</tr>
<tr>
<td>Order</td>
<td>Option to determine in which order the instance matches a record. The lower the number, the higher the priority for matching the URL.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The system matches each record one by one. When a record matches a supported URL the instance generates a banner. The order is a way to put the more general supported URLs lower down on the list to prevent the usurping of specific links you want to match.</td>
</tr>
<tr>
<td>Path segment</td>
<td>Use the reference lookup icon to select the path segment you created earlier. The system begins to search for matches by first checking the referenced path segment. If there is no match, the instance does not display a banner.</td>
</tr>
<tr>
<td>Has query keys</td>
<td>Comma-separated string field for query keys. Each query key added into this string field must exist in the URL for there to be a match. An example entry is id,table,sys_id</td>
</tr>
<tr>
<td>Has query values</td>
<td>Option to add a name a value for each query key you added. An example is <strong>Name</strong> = id and <strong>Value</strong> = ticket</td>
</tr>
<tr>
<td>Mobile screen builder</td>
<td>Use the mobile screen builder script block to define the screen type to build and the data the screen uses. Using one of the provided methods in the script block it creates a list screen, record screen, or browser screen. Each method has its own parameter requirements. You can find specific parameters in the script block comments.</td>
</tr>
</tbody>
</table>
d. Select Submit.

What to do next
After defining when banners appear and the screen and content the user views, consider enhancing the experience by associating the screen to a preconfigured table. See, Mapping specific mobile screens for universal linking.

Mapping specific mobile screens for universal linking
Configure universal links for screen mapping so a user is redirected to a previously styled record screen or list screen. Use this option to provide a customized experience for the user.

Before you begin
Role required: admin

About this task
When creating universal links, you define the screen type and the data the user sees after they tap the mobile banner. See, Create path segment records and supported URL records for universal linking. Also there is an option to create a screen mapping for universal links. Users click these links to see a preconfigured screen based on a defined table.

Note: Universal linking mapping does not apply to browser screens, as there are no options for styling and configuring a web page.

Procedure
1. In the web-based UI, navigate to System Mobile > Mobile Universal Liking > Screen Mappings.
2. Select New to create a universal link screen mapping.
3. On the form, fill in the fields.

Universal Link Screen Mapping form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>From the menu, select a table associated with the mobile universal linking screen. The content can be a preconfigured list screen and/or a record screen.</td>
</tr>
<tr>
<td>Mobile app</td>
<td>Use the reference lookup icon to select which mobile app the links navigate to.</td>
</tr>
<tr>
<td>Note:</td>
<td>The selection here is overridden by the selection made in the Mobile app field listed in the topic Create path segment records and supported URL records for universal linking.</td>
</tr>
<tr>
<td>Active</td>
<td>Option for enabling the activation of this record.</td>
</tr>
<tr>
<td>List screen</td>
<td>Use the reference lookup icon to select a preconfigured list screen associated with the table, selected from the Table field.</td>
</tr>
<tr>
<td>Note:</td>
<td>Only use single item streams and single segment list screens. You cannot use multi-item streams and multi-segment list screens in this reference field.</td>
</tr>
</tbody>
</table>
### Field | Description
--- | ---
Record screen | Use the reference lookup icon to select a preconfigured record screen that is associated with the table selected from the **Table** field.

4. Select **Submit**.

## Roles and permissions for ServiceNow Mobile Apps

Limit user access to parts of the mobile app.

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

**Applications**

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 a mobile application using Guided Application Creator*.

**Applets**

Allow only users with specified roles to access applets within your mobile applications. For example, allowing only managers to view user records for their employees. For more information, see *Mobile screens*.

**Applet Launchers**

Allow only users with specified roles to access applet launcher within your mobile applications. For example, create applet launchers specifically for certain roles or regions, and permit only users with specified roles to view these launchers. For more information on applet launchers, see *Launcher screens*.

**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.

## Siri shortcuts

Configure Siri shortcuts to give your users quick access to common app functions app using their iOS devices.

By default, your iOS users can use the following pre-configured shortcuts in the Now Mobile app:

- Open a chat window
- Browser items and services
- Open my tasks
- Open my requests

In the Mobile Agent app, users can access Agent Chat if the following conditions are met:
• The Agent Chat[com.glide.interaction.awa] plugin must be installed on your instance. For details on configuration, see Setting up Agent Chat.

• The mobile app must have a chat launcher quick action enabled. For details on enabling chat launcher, see Enable Live Agent in the Now Mobile app.

Configure a Siri shortcut
Change the destination screen of your Siri shortcuts to direct your users to a different screen than the default.

Before you begin
Role required: admin
As an admin, you can change the destination screen of the three of default Siri shortcuts listed below. Creating new Siri shortcuts is currently not supported.

Procedure
1. In the web-based UI, enter sys_sg_screen_shortcut.list in the filter navigator.
2. Open the record for the shortcut you want to modify.

<table>
<thead>
<tr>
<th>Siri shortcuts</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Shortcut</strong></td>
</tr>
<tr>
<td>com.servicenow.requestor.sc</td>
</tr>
<tr>
<td>com.servicenow.requestor.tasks</td>
</tr>
<tr>
<td>com.servicenow.requestor.request</td>
</tr>
</tbody>
</table>

⚠️ Note: You can update the screen your user will navigate to when using a shortcut, however you cannot change the text of the shortcut.

⚠️ Note: Open the record by clicking on the **Shortcut** field. Clicking on the **Screen** field will open the screen record rather than the shortcut record.

3. In the screen shortcut record, update the **Screen** field by clicking the reference icon (🔗) and selecting an applet. This applet displays to your users when they access the associated Siri shortcut.

Mobile surveys
Mobile users can take surveys without navigating outside ServiceNow mobile apps. As an administrator, you can use the new parameter and scriptable screens to create a survey experience. Users can input parameters and use surveys even while offline.

Displaying campaigns on ServiceNow mobile
Use campaigns to deliver messages and important information to your users. You can set up and display a campaign on ServiceNow mobile.

Mobile campaigns enable you to share curated content using a scrolling list of images. You can mix any of three card types in the carousel: video, image, and text. At the bottom of the campaign cards, dots represent up to eight cards and numbers more than eight cards. A mobile campaign, with its carousel format, is most effective when displayed at the top of the mobile home page.
Note:
To set up a campaign to display on mobile devices, you must have the HR Service Content Delivery plugin [com.sn_content_delivery], installed. For more information, see Activate Content Publishing.

You create campaign content through Content Automation in HR Service Delivery. For more information, see Campaigns for Content Experiences and Configure mobile content.

Examples of a mobile campaign with an image, text, and video card

Displaying mobile campaign process
Content for campaigns include videos, images (banners), and text cards.

You specify how to display the cards within a mobile campaign by first creating templates that provide information about the appearance of cards. Once you have defined the card appearances, you build and link components to use the data provided by the item views and display the content as a campaign on your mobile device.

This process involves multiple components contained in a content UI section:

- An item view for each type of content card, which serves as a template for the appearance of the three types of cards: video, image, and text.
- An item configuration for each type of content card, which contains the item view.
- An item stream that contains the data item, which collects the relevant content for the campaign, and the item configuration.
- An item stream container, which references the item stream.

Create appearance templates for cards for a mobile campaign
Create item view templates that contain information about the appearance of each type of card displayed in a mobile campaign.

Before you begin
Role required: admin
**About this task**

When creating cards, you need to assign an item view. Item views provide information about a card’s appearance. Therefore, for each card style you want to use, you need a separate item view even if the card type is the same. For example, if you want some image cards to have light text on a dark background and other image cards to have dark text on a light background, you would need to create a separate item view for each of those styles.

**Procedure**

1. In the web-based UI, enter `sys_sg_item_view.list` in the filter navigator to open a list of item views.
2. Click **New**.
3. On the form, fill in the fields.

**Item view form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the item view. &lt;i&gt;Note: For easy identification, include the card type (image, text, or video) in the name.</td>
</tr>
<tr>
<td>Short Description</td>
<td>Optional additional details.</td>
</tr>
<tr>
<td>Item view JSON</td>
<td>The configuration code you created for the card type. For more information, see Configure the details for the appearance of cards for a mobile campaign.</td>
</tr>
<tr>
<td>Table</td>
<td><code>sn_cd_content_mobile</code> the HR Service Content Delivery table, which contains the content of the cards.</td>
</tr>
<tr>
<td>Dependency fields</td>
<td>Dependency fields taken from data entered in the <strong>Item view JSON</strong> field.</td>
</tr>
<tr>
<td>UI Styles</td>
<td>UI styles to apply to text cards. For information about how to configure the font color and background color for text cards, see .</td>
</tr>
</tbody>
</table>

4. Click **Update**.

**Results**

After creating item views for each card style, you need to build and link components to display your mobile campaign. For more information, see Configure mobile campaign components.

**Configure the details for the appearance of cards for a mobile campaign**

Configure item views to provide the details that define the appearance of cards within a mobile campaign to enhance display options and make your campaign more effective.

When creating item views for cards for a mobile campaign, you need to construct required JSON code to define the appearance of the cards. You then provide this code in the **Item view JSON** field of the Item view form, as described in Create appearance templates for cards for a mobile campaign.
Note: If any of the configurable values are empty or the line of code is removed, the default value is used.

Item view JSON for text cards
For a text card, copy this sample JSON code and paste it in the Item view JSON field within an item view form.

```json
{
    "Type": "ViewGroup",
    "Distribution": "Equal",
    "Orientation": "Vertical",
    "Alignment": "Center",
    "Margin": {
        "Left": 0,
        "Right": 0,
        "Top": 0,
        "Bottom": 0
    },
    "Children": [
        {
            "Type": "Text",
            "CellId": "title",
            "TextAlignment": "center",
            "Margin": {
                "Left": 16,
                "Right": 16
            },
            "MaxLines": 3,
            "Font": {
                "Weight": "bold",
                "Size": 36,
                "MinSize": 24
            }
        }
    ]
}
```

The parameters in the sample JSON code are as follows:

• The Type parameter value is not configurable.

• The CellId parameter value in the sample code is based on the value in the default base system version of the HR Service Content Delivery table [sn_cd_content_mobile]. If the value in that table has changed, add the changed value in the sample JSON code.

Note: To access the table to check the values, enter sn_cd_content_mobile.list in the navigation filter of your application navigator.

• The other parameter values in this sample code for text cards are standard setup values that provide a balanced appearance for the cards. You can customize these values.

Item view JSON for video cards
For a video card, copy this sample JSON code and paste it in the Item view JSON field within the item view form.

```json
{
    "Type": "Media-Video",
    "CellId": "video_link.url",
}
```
The parameters in the sample JSON code are as follows:

- The Type parameter value is not configurable.
- The CellId, TitleCellId, and SubtitleCellId parameter values in the sample code are based on the values from the default base system version of the HR Service Content Delivery table [sn_cd_content_mobile]. If any of the values in that table have changed, add the changed value in the sample JSON code.

⚠️ Note: To access the table to check the values, enter sn_cd_content_mobile.list in the navigation filter of your application navigator.

### Item view JSON code for image cards

For an image card, copy this sample JSON code and set any values that differ from the default values. If you want to use the default values, you can delete that line of code. Once you have set your desired values, paste the JSON code in the Item view JSON field within the item view form.

```json
{
"Type": "Media-Image",
"CellId": "image",
"TitleCellId": "title",
"SubtitleCellId": "text",
"Parallax": <true|false>,
"DisplayType": "<over|under>",
"DisplayTheme": "<light|dark>",
"FocusOnFaces": <true|false>
}
```

The standard parameters in the sample JSON code are as follows:

- The Type parameter value is not configurable.
- The CellId, TitleCellId, and SubtitleCellId parameter values in the sample code are based on the values from the default base system version of the HR Service Content Delivery table [sn_cd_content_mobile]. If any of the values in that table have changed, add the changed value in the sample JSON code.

⚠️ Note: To access the table to check the values, enter sn_cd_content_mobile.list in the navigation filter of your application navigator.

The custom parameters for the item view for image cards are as follows:

- **Parallax**
  - Determines whether to create an illusion of depth and perspective on the image.
  - true (the default): The text at the bottom of the card moves a slower speed than the background image, making the two objects appear as though they are on a different three-dimensional plane.
  - false: The text at the bottom of the card and the image move at the same speed.

- **DisplayType**
Determines whether text is displayed under the image (under) or as an overlay on the image (over). The text includes the title and the subtitle. For a smooth visual experience, use the same format for all the card types. The default value is over.

<table>
<thead>
<tr>
<th>Text displayed under image</th>
<th>Text displayed as overlay</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image with text" /></td>
<td><img src="image2.png" alt="Image with text" /></td>
</tr>
</tbody>
</table>

**DisplayTheme**

Determines the color of the text overlay. This option is available only if the display type is over. Use light for text that is to be displayed over a dark background. The default value is light.

<table>
<thead>
<tr>
<th>DisplayTheme set to light</th>
<th>DisplayTheme set to dark</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image3.png" alt="Image with text" /></td>
<td><img src="image4.png" alt="Image with text" /></td>
</tr>
</tbody>
</table>

**FocusOnFaces**

When set to true (the default), images on an image card are cropped to where the faces are captured in the frame.

**Configure mobile campaign components**

Configure components to display the image, video, and text cards in a campaign for your users to view and interact with on mobile devices.

**Before you begin**

To configure components to display a campaign on mobile devices, ensure that you have established the following items:
• Developed content for your campaign. For more information, see Campaigns for Content Experiences and Configure mobile content.

• Defined the appearance criteria for at least one card type. For more information, see Create appearance templates for cards for a mobile campaign and Configure the details for the appearance of cards for a mobile campaign.

• Have the HR Service Content Delivery plugin [com.sn_content_delivery] installed. For more information, see Activate Content Publishing.

Role required: admin

**Procedure**

1. Create a data item to collect the data for the campaign content.

   a. In the web-based UI, enter `sys_sg_data_item.list` in the filter navigator to open a list of data items.

   b. Click **New** to create a data item.

   c. On the form, fill in the 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.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional additional details.</td>
</tr>
<tr>
<td>Condition Type</td>
<td>Select the method to use when building conditions to retrieve data from your data item. The available options are:</td>
</tr>
<tr>
<td></td>
<td><strong>Declarative</strong> Create conditions for the data item using the condition builder.</td>
</tr>
<tr>
<td></td>
<td><strong>Scripted</strong> Use a script to determine the conditions of your data item. A text window to enter a script appears below this field when you select this option.</td>
</tr>
<tr>
<td>Table</td>
<td><code>sn_cd_content_mobile</code> the HR Service Content Delivery table, which contains the content of the cards.</td>
</tr>
<tr>
<td>Query Condition</td>
<td>Condition that filters content data retrieved from the HR Service Content Delivery table. For example, if a field called month exists in the table, you could set the condition <code>[month][is][September]</code></td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group by</td>
<td>Select the field used to group the records in your list.</td>
</tr>
</tbody>
</table>

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

**2.** Create an item configuration for each item view you use.

**a.** In the web-based UI, enter `sys_sg_master_item.list` in the filter navigator.

**b.** Click **New** to create an item configuration.

**c.** On the form, fill in the fields.

### item configuration form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A title for the item configuration.  📘 <strong>Note:</strong> Include the card type in the name.</td>
</tr>
<tr>
<td>Item view</td>
<td>The item view to use for the card type.</td>
</tr>
<tr>
<td>Table</td>
<td><code>sn_cd_content_mobile</code> the HR Service Content Delivery table, which contains the content of the cards.</td>
</tr>
</tbody>
</table>
| Condition   | Condition that filters display-related data retrieved from the HR Service Content Delivery table. Use a declarative condition to create conditions for the item configuration using the condition builder.

For example, you could specify to display all image records related to the condition September, which you defined in the data item condition. You then set the condition `[Content type][is][Image]` and as a result all images tagged as September are displayed in the mobile campaign.

| Embedded screen | Not applicable for mobile campaigns. |

**3.** Create an item stream, which connects the data item with the item configuration.

**a.** In the web-based UI, enter `sys_sg_item_stream.list` in the filter navigator.

**b.** Click **New** to create an item stream.

**c.** On the form, fill in the fields.
4. Create an item stream container.
   a. In the web-based UI, enter `sys_sg_item_stream_container.list` in the filter navigator.
   b. Click **New**.
   c. In the **Name** field, enter a title for the item stream container.
   d. Right-click on the form header (≡) and select **Save**.
   e. In the Item Stream Container M2M Item Streams section, click **New**.
   f. On the form, fill in the fields.

   **Item Stream Container M2M Item Stream form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item stream</td>
<td>The item stream that you want to add to the item stream container.</td>
</tr>
<tr>
<td>Item stream container</td>
<td>The item stream container that contains the item stream.</td>
</tr>
<tr>
<td>Order</td>
<td>A number that indicates where to place the item stream container. Lower-numbered items display before higher-numbered items.</td>
</tr>
</tbody>
</table>

   g. Click **Submit**.
   h. Click **Update**.

5. Create an applet launcher content UI section to contain the campaign.
   a. In the web-based UI, enter `sys_sg_content_section.list` in the filter navigator.
   b. Click **New**.
c. On the form, fill in the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Title for the content UI section. This title is not displayed in the mobile UI.</td>
</tr>
<tr>
<td>Active</td>
<td>Option for enabling the display of the campaign in your mobile instance.</td>
</tr>
</tbody>
</table>

d. Click Submit.

What to do next
Add the content UI section to an applet launcher. For more information, see Configure content UI sections.

Mobile external users
Define external users to have availability to access your instance, but without access to all the capabilities allocated to internal users. External users are usually third parties, contractors, or external agents. External users are not employees of your organization.

External users can perform tasks like view knowledge base articles, create, edit and manage incidents, and use Virtual Agent chat, without affecting the security of the instance. The snc_external role is assigned to external users and specifies that such users have fewer privileges. For more information on the snc_external role and how it is defined, see External roles in self-registration and Explicit roles in CSM.

Functionality not available to mobile external users
The following is a list of different functionalities and capabilities not available to external users on mobile.

- **Zing search** - The search bar is not displayed for screens that use Zing search. For AI Search, the search bar is displayed.
- **Charts** - Both Performance Analytics and reporting charts are not displayed.
- **Attachments** - Users are not able to view, edit, or upload attachments.
- **Offline** - Mobile offline mode is not available. External users without an internet connection cannot continue to work from a mobile device.
- **Debug drawer** - The ability to display performance-related data via the debug drawer, is not available.
- **Impersonate user** - External users cannot use the mobile impersonation functionality.

Mobile system property configurations
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.
Require an app PIN for the mobile app

Require users to enter a PIN when the application has been inactive for five minutes. To require the mobile user to set and enter a local application PIN, add the system property glide.sg.require_mobile_application_pin.

Before you begin
Role required: admin

About this task
Users generate a six-digit code for the app PIN. The PIN must be entered when they log in to an instance from their mobile device, or after the application has been inactive for more than five minutes. If your users have faceID, touchID, or similar biometric security configured on their phone, they can use biometric authentication in place of the PIN.

Procedure
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>glide.sg.require_mobile_application_pin</td>
</tr>
<tr>
<td>Type</td>
<td>true</td>
</tr>
</tbody>
</table>
Note: The mobile app is automatically locked after five minutes of inactivity. Users accessing the mobile app after a period of inactivity must enter their PIN code.

Configure mobile re-authentication system properties

Set mobile re-authentication system properties so users must re-authenticate their login credentials when performing specific actions.

To activate the re-authentication option, select Re-authentication in the Preconditions field within the Action functions table. See, Configure an action function.

Configure mobile re-authentication login method

Define the re-authentication method to be either single sign-on (SSO) or local login, depending on your security requirements.

Before you begin

Make sure to select Global as the application scope.

Role required: admin

Procedure

1. Type sys_properties.list in the filter navigator.
2. Select New in the System Property table.
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>glide.authenticate.reauth.login.method</td>
</tr>
<tr>
<td>Type</td>
<td>string</td>
</tr>
<tr>
<td>Value</td>
<td>sso - To use single sign-on</td>
</tr>
<tr>
<td></td>
<td>db - To use local login</td>
</tr>
</tbody>
</table>

Configure mobile re-authentication SSO login

Define which identity provider to use for the SSO login. If this property is not defined, the system reuses the default identity provider used for a regular login.

Before you begin

Only configure this property if glide.authenticate.reauth.login.method is set to sso.

Make sure to select Global as the application scope.

Role required: admin

Procedure

1. Type sys_properties.list in the Filter Navigator.
2. Select New in the System Property table.
3. In the form, match the following values:
Configure mobile re-authentication logout option

Configure this parameter to force your users to log out each time before they re-authenticate.

**Before you begin**

Only configure this property if `glide.authenticate.reauth.login.method` is set to `sso`.

Make sure to select `Global` as the application scope.

**Note:** Note: Some SSO identity providers may require users to be logged out before successfully enabling a re-authentication flow. Use the property `glide.authenticate.reauth.logout.first`, to force users to log out, before re-authentication.

Role required: admin

**Procedure**

1. Type `sys_properties.list` in the Filter Navigator.
2. Select **New** in the System Property table.
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.authenticate.reauth.login.sso_id</code></td>
</tr>
<tr>
<td>Type</td>
<td>string</td>
</tr>
<tr>
<td>Value</td>
<td>Enter the <code>sys_id</code> of the SSO provider as listed in the <code>[sso_properties]</code> table.</td>
</tr>
</tbody>
</table>

Configure mobile re-authentication for single or multiple use

Define whether each mobile action requires re-authentication. Alternatively, define whether the user can perform multiple actions without the need to re-authenticate each time.

**Before you begin**

Role required: admin

**Procedure**

1. Type `sys_properties.list` in the filter navigator.
2. Open the record for `glide.sg.reauthentication.single_use_token`.
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.authenticate.reauth.logout.first</code></td>
</tr>
<tr>
<td>Type</td>
<td>`true</td>
</tr>
<tr>
<td>Value</td>
<td><code>true</code> - Users are forced to log out each time before re-authenticating.</td>
</tr>
<tr>
<td></td>
<td><code>false</code> - Users are not forced to log out before re-authenticating.</td>
</tr>
</tbody>
</table>
### Configure mobile re-authentication login timespan

Define a time period in seconds that a user is not required to re-authenticate themselves. This parameter only applies when a user has permission to re-authenticate multiple times.

**Before you begin**

If glide.sg.reauthentication.single_use_token is set to false, you can configure this property.

Make sure to select Global as the application scope.

Role required: admin

**Procedure**

1. Type `sys_properties.list` in the Filter Navigator.
2. Select **New** in the System Property table.
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>glide.authenticate.reauth.token.lifespan</td>
</tr>
<tr>
<td>Type</td>
<td>integer</td>
</tr>
<tr>
<td>Value</td>
<td>Integer value in seconds.</td>
</tr>
</tbody>
</table>

### Configure the blur app option to improve security

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.

**Procedure**

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>glide.sg.blur_ui_when_backgrounded</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Type</td>
<td>true</td>
</tr>
<tr>
<td>Value</td>
<td>true</td>
</tr>
</tbody>
</table>

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

**Results**
With the property in place, the app is blurred when not in focus.

---

**Configure the status for rooted and jailbroken devices**
Define whether jailbroken (iOS) and rooted (Android) devices are permitted on your mobile device. The default value is set to false to increase security and to minimize possible disruption to your system.

**Before you begin**
Role required: admin

**About this task**
Configure settings to decide whether to allow jailbroken and rooted devices.
Note: If you select true, jailbroken and rooted devices can launch the ServiceNow application, however your device may be compromised and you are at risk of losing your ServiceNow data.

Procedure

1. Type sys_properties.list in the filter navigator.
2. Open the record for glide.sg.allow_rooted_jailbroken_devices system.
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>glide.sg.allow_rooted_jailbroken_devices system</td>
</tr>
<tr>
<td>Type</td>
<td>true</td>
</tr>
</tbody>
</table>
| Value                      | true - Rooted and jailbroken devices can be used when using ServiceNow® mobile apps.  
                             | false - Rooted and jailbroken devices are not permitted when using ServiceNow® mobile apps. (Default value) |

Configure the mobile app to clear the copy/paste clipboard and block ability to share content

To have the mobile app clear the pasteboard when the app enters the background, as well as block the ability to share content, add the system property glide.sg.clear_pasteboard_when_background.

Before you begin
Role required: admin

About this task
By default, content that you copy in the mobile app remains on your device’s clipboard even after you close the mobile app. In addition, you can share copied content to an external app and to a non-native virtual agent. Use the system property glide.sg.clear_pasteboard_when_background to change this default behavior.

Note:

- Text edit menus on Android devices may display a share option, even though its functionality is disabled.
- Some third-party keyboards on Android devices are not disabled from the ability to paste and share content to the keyboard’s clipboard.
- The ability to copy/paste and share from some web views in the app may result in unexpected behavior.

Procedure

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

**Before you begin**
Role required: admin

**About this task**
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 1000.

**Procedure**
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>glide.sg.data_item.row_count</td>
</tr>
<tr>
<td>Type</td>
<td>integer</td>
</tr>
<tr>
<td>Value</td>
<td>&lt;maximum-number-of-records-retrieved&gt;</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.choice_list.window_size`.

**Before you begin**
Role required: admin

**About this task**
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.

**Procedure**
1. Type `sys_properties.list` in the Application Navigator.
2. Open the record for `glide.sg.choice_list.window_size` or create a new property if it does not exist.
3. In the form, match the following values:
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.

**Before you begin**
Role required: admin

**About this task**
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.

**Procedure**
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>glide.sg.list.max_items_number</td>
</tr>
<tr>
<td>Type</td>
<td>string</td>
</tr>
<tr>
<td>Value</td>
<td>&lt;maximum-number-of-parameters-items&gt;</td>
</tr>
</tbody>
</table>

Configure a placeholder image for missing images in mobile apps

You can specify an image on your instance as a placeholder for missing images. This image appears in your mobile apps when a record has an image field with an empty value, such as a user avatar or catalog item. You can select a different image to use for each table on your instance.

**Before you begin**
Role required: admin

**About this task**
You configure placeholder images by creating properties on the System Properties [sys_properties] table. The property is table-specific. If you want to define placeholder images for many tables, you must create multiple properties. Multiple tables can use the same image as a placeholder.
Procedure

1. Upload an image to your instance to use as a placeholder. For details on uploading images, see Storing images in the database.

2. To open the system properties list, type `sys_properties.list` in the Application Navigator. Some tables already have a placeholder image defined. You can search the system properties table for properties that start with `glide.sg.image.default` to see any existing properties. Creating multiple properties for the same table can cause inconsistent results.

3. Click New.

4. Use the following information to complete the fields on the system property form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
</table>
| Name      | Enter `glide.sg.image.default.[tablename]`. Replace `[tablename]` with the name of the table you want this property to apply to. For example to use the Catalog Item table, enter `glide.sg.image.default.sc_cat_item`.
| Type      | Select String |
| Value     | Enter the name of your image. This value is the same as the Name field on the image [db_image] record. |

5. Click Update.

Example:
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`.

<table>
<thead>
<tr>
<th>glide.sg.image.default.sys_user property form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Type</td>
</tr>
<tr>
<td>Value</td>
</tr>
</tbody>
</table>

**Note:** Match `<file-name>` to the file name of the image that you have uploaded.

Configure which external apps are available for deep linking

Administrators can define which external apps can be used in deep links with the `glide.sg.allowed_external_deeplinks` property.

Before you begin
Role required: admin

About this task
ServiceNow mobile supports deep linking to third party applications using the URL smart buttons. Use the `glide.sg.allowed_external_deeplinks` system property to define which third party applications can be used as the targets of deep links in your ServiceNow mobile apps.
Procedure

1. Type sys_properties.list in the Application Navigator.
2. Open the record for glide.sg.allowed_external deeplinks.
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>glide.sg.allowed_external_deeplinks</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Value</td>
<td>Comma separated list of the names of the third party apps that can be targets of deep links within ServiceNow mobile apps.</td>
</tr>
</tbody>
</table>

4. Click Save.

Results

Only third party apps listed in value of the property can be linked to using deep linking.

⚠️ Note: If the value of this property is left empty, or the property is deleted, there are no restrictions on deep linking to third party apps.

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.

Before you begin

Role required: none

Procedure

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>InstanceUrl</td>
<td>The URL for your instance. For example, <a href="https://example.servicenow.com">https://example.servicenow.com</a></td>
</tr>
<tr>
<td>Name</td>
<td>The name of your instance. For example, Example.</td>
</tr>
</tbody>
</table>

Example

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 these examples. The image shows how the instance appears in instance selection screen for the app.

```
{
 "InstanceUrl":"https://example.servicenow.com",
 "Name":"Example"
}
```

Example

This second example includes only the `InstanceUrl` value, which is also valid.

```
{
 "InstanceUrl":"https://example.servicenow.com"
}
```

**Note:** ServiceNow instances do not provide a method to create JSON files. You can create these files using your text editor of choice.

**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 as text rather than JSON. For example, `https://example.servicenow.com`.

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

**Note:** ServiceNow instances do not provide a method to create QR codes, however there are many online resources you can use to create QR codes using the JSON created in the previous steps.

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 a mobile app.

**Configure mobile app token lifespan**

Configure the length of time it takes for the app to time out.

**Before you begin**

Role required: admin

**About this task**

Mobile apps timeout when their associated OAuth token expires. Token lifespans are considered active if the app is in the foreground or if the app is processing a long running task in the background.

**Procedure**

Configure the OAuth entity for a mobile app.
a. Navigate to System OAuth > Application Registry and open the OAuth entity for the application you want to modify. For example, open the ServiceNow Agent record to modify the timeout for your Mobile Agent app app.

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.

Results
For example, if you want the mobile app's token to expire after 30 minutes, use the following configuration.

<table>
<thead>
<tr>
<th>Configuration point</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh Token Lifespan for the Mobile Agent app mobile app</td>
<td>1800 seconds</td>
</tr>
<tr>
<td>Access Token Lifespan for the Mobile Agent app mobile app</td>
<td>1800 seconds</td>
</tr>
</tbody>
</table>

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

Force local login in mobile apps
Configure the force local login option to provide local login experience on mobile apps even when the instance is configured with Single Sign On (SSO) configuration. You can configure this feature independently on any available ServiceNow® app.

Before you begin
Role required: admin

Procedure
1. Navigate to All > System Mobile > Mobile Publishing > Native Clients.
2. Open the record for the mobile app where you want to force local logins.
3. Enable the Force Local Login field.
4. Click Update.

Results
The selected mobile app routes your users to the local login authentication page. The app will default to the authentication method defined on your instance if this field is disabled.

User Experience Analytics for mobile applications

User Experience Analytics helps you understand usage and adoption of your mobile applications. User Experience Analytics was formally called Mobile Analytics.

The ServiceNow® User Experience Analytics application provides dashboard views for monitoring key performance indicators of users of your web and mobile applications built with the ServiceNow mobile platform and web assets in Service Portal instances. Visualize metrics and interactions to better understand the user experience, and create more intuitive journeys for your users.
Mobile location tracking configuration

As an administrator, you can activate geolocation tracking for your ServiceNow® Mobile Agent app app users by installing the Geolocation plugin (com.snc.geolocation). Complete other geolocation tracking tasks to finalize the configuration.

Activate the Geolocation plugin

Enable geolocation features on your instance by activating the Geolocation plugin (com.snc.geolocation). For details on plugin activation, see Activate a plugin.

Enable location tracking on your user records

After activating the geolocation plugin, you must enable location tracking for users who need to use the geolocation feature. Enable geolocation records for a user by opening a User [sys_user] record, and enabling the Geolocation Tracked field.
Users must enable location tracking on their mobile devices

After enabling geolocation tracking on user records, your user will see geolocation tracking settings in the Settings tab of their mobile devices. Your users can enable or disable location tracking on their device, and set a duration to enable location tracking. For details on these settings, see Location tracking for mobile.

Offline mode

Configure offline mode to enable your users who have no internet connection to continue working from a mobile device.

Configure specific application, applets, or functions for users to use offline in the Mobile Agent app app.

Watch this three-minute video to learn how offline mode works, how to download data, enable and disable offline mode, synchronize your outbox, and resolve synchronization errors. Demonstrates how offline mode works in the ServiceNow mobile app.

Offline mode works by creating a cache of record on your mobile device that your users can view and update while offline. This cache is a limited set of data based on the applications your users can access. By default, this cache expires 48 hours after it is downloaded to a mobile device. This means that any changes your users make to the cache not synchronized to your instance within 48 hours are lost.
While in offline mode, only parts of the app with cached data are available. For example, if you have only enabled offline mode for tasks and inventory, all other parts of your mobile app are unavailable while offline.

When your users have network access again, they can disable offline mode and synchronize their cached data with the data on your instance. Updates between the instance and the mobile device are automatic, unless there is a conflict. Users can resolve conflicts in their outbox.

**Incremental Offline**

Incremental offline mode periodically compares the data on your instance and data cached on your mobile device. When the record watcher detects a difference, triggers an update containing only the changed files. Using this option, your users are not required to update the entire cache manually or according to a fixed schedule. These smaller updates are ideal in situations where users frequently or unexpectedly lose network connectivity.
Request Offline Mode

To activate this feature, request activation of the SG Offline support plugin (com.glide.sg.offline).

**Before you begin**

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

Role required: admin

**Procedure**

1. Navigate to **All > System Applications > All Available Applications > All.**
2. On the All Applications page, select **Request Plugin** to open the Request form on HI.

3. On Now Support, select the redirect link to access the Now Support Service Portal Service Catalog.

4. Select your instance.

5. Select **Actions > Activate Plugin.**

6. On the form, fill in the fields.

**Activate Plugin request 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>The date and time must be at least two business days from the current time.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> Plugins are activated in two batches, once in the morning and once in the evening, on every business day in the US Pacific time zone. If the plugin must be activated at a specific time, enter the request in the Reason/Comments field.</td>
</tr>
<tr>
<td>Reason/Comments</td>
<td>Information that would be helpful for the ServiceNow personnel who are activating the plugin. For example, if you need the plugin activated at a specific time instead of during one of the default activation windows, specify that in the comments.</td>
</tr>
</tbody>
</table>

7. Select **Submit**.

**Related reference**

List of plugins (San Diego)

**Configure Offline Mode behavior**

Create system properties to customize the Offline Mode behavior of the mobile application.

**Before you begin**

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

Role required: admin

**Procedure**

1. In the application navigator, enter: **sys_properties.list**.
2. Verify that the property does not exist by searching for the property name.
3. Click **New**.
4. Complete the System Property form using the property names listed in this table. Use the information in the description to determine a value for the property.

**Note:** For more detail on creating system properties, see Add a system property

**Mobile offline system properties**

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.sg.offline.attachment.allowed_content_types</td>
<td><strong>Note:</strong> A 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, offline mode does not download any files.</td>
</tr>
<tr>
<td>glide.sg.offline.attachment.max_size</td>
<td>The maximum size, in bytes, for attachments that are cached while in offline mode. If no download, a placeholder displays. The default size is 50 MB.</td>
</tr>
<tr>
<td>Property Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.sg.offline.attachment.max_total_bytes</td>
<td>Maximum space allocated for all attachments in cache. Default value is 400 MB, max is 2 GB. The property value is in MB.</td>
</tr>
<tr>
<td>glide.sg.offline.enabled</td>
<td>Enable offline capabilities 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>
<tr>
<td>glide.sg.offline.expiration</td>
<td>The length of time before cached data is expired. The default length is 48 hours. After 48 hours, the system deletes the data due to security protocol.</td>
</tr>
<tr>
<td>glide.sg.offline.incremental.client_polling_interval</td>
<td>Determines minimum polling frequency. The property value is in minutes. The default value is 1, the minimum value is 0.</td>
</tr>
<tr>
<td>glide.sg.offline.incremental.enabled</td>
<td>Enable the incremental offline for all users. Incremental offline is disabled by default.</td>
</tr>
<tr>
<td>glide.sg.offline.incremental.record_watcherExpiration</td>
<td>Determines how long a record watcher is active but offline payload is not retrieved before it is removed. The property value is in minutes. The default value is 30.</td>
</tr>
<tr>
<td>glide.sg.offline.incremental.silent_push.max_pushes_per_hour</td>
<td>Limits the number of silent pushes to send per 60 minutes per user. The property value is in minutes. The default value is 3. The maximum value is 10.</td>
</tr>
<tr>
<td>glide.sg.offline.incremental.silent_push.min_wait_time</td>
<td>Determines minimum time between silent pushes per user. The property value is in minutes. The default value is 1. The maximum value is 1440.</td>
</tr>
<tr>
<td>glide.sg.offline.roles</td>
<td>A comma-separated list of role names that are allowed to work in offline mode. If empty, all users may use offline mode.</td>
</tr>
<tr>
<td>glide.sg.offline.scheduled_download.cachesync_offset</td>
<td>Number of minute before a user's schedule to begin payload generation. The property value is in minutes.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Be sure not to set the value of this property to a value that was less than the interval that the Mobile Offline Scheduling job is running. Doing so will prevent offline payloads from being generated.</td>
</tr>
<tr>
<td>glide.sg.offline.scheduled_download.enabled</td>
<td>Enable scheduled downloading for all Agent app users</td>
</tr>
<tr>
<td>glide.sg.offline.scheduled_download.reminder_offset</td>
<td>Number of minutes before a user's schedule that they receive a manual download reminder. If no there is no cache or the cache is expired, offline mode sends a reminder for the user to manually download</td>
</tr>
<tr>
<td>Property Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.sg.offline_payload.refresh_frequency</td>
<td>The number of minutes before a cached record is flagged for refresh. The default is 240 minutes. The property value is in minutes.</td>
</tr>
</tbody>
</table>

5. Click Submit.

Results
For a view of the offline mode end-user experience, see: Working offline in mobile.

Configure Offline Mode for applications or applets

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

About this task
Because applications contain applets, if you set an applet to work offline, you must also set the application to work offline.
Role required: admin

Procedure
1. In Studio, navigate to the application you want to configure Offline Mode for, then click the pop-out icon ( ) that appears when you point to the application name.
2. Optional: Open an applet you want to allow users to 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.

What to do next
Determine which functions are available to users in Offline Mode.

**Define the number of displayed records in Offline Mode**

Define the number of records to display to users in Offline Mode. Choose between 0 through 1000 records. This range gives you the flexibility to display different amounts to the user in online and Offline Modes.

**Before you begin**
Role required: admin

**Procedure**
1. Navigate to **All** and in the filter enter `sys_sg_item_stream_segment.list`.
2. Select an existing screen segment.
3. In the **Max Offline Rows** field, enter a number from 0 through 1000 to be the number of displayed rows displayed in Offline Mode.

**Note:** When this field is empty, the default value of 1000 rows is used.
4. Right-click in the header and select **Save**.

**Configure data items in Offline Mode**

Define a separate data item for Offline Mode, giving you the flexibility to define the amount of data to display when a user is offline.

**Before you begin**
Role required: admin

**About this task**
Use the Mobile Offline properties area to specify the type and amount of information to display to a user when they are in Offline Mode. For example, you can display a month’s worth of tasks in online mode and only three days' worth in Offline Mode. In both online and offline modes, the data item type is always the same and the information is extracted from the same table. For more information, see **Data items**.
Note: If a data item is not created for Offline Mode, then the standard data item is used for both offline and online modes.

Procedure
1. Navigate to All and in the filter enter sys_sg_data_item.list.
2. Select an existing data item.
3. In the Offline Properties area, define offline mode query conditions for the data item to conform to. Create conditions using the Condition builder.

Note: Declarative and Scripts are supported conditional types.
4. Right-click in the header and select Save.

Configure Offline Mode properties for action functions

Determine which fields and functions are available to users accessing the app in Offline Mode

Before you begin
Role required: admin

Procedure
1. In Studio, navigate to Mobile Studio > Functions > Actions.
2. Click the pop out icon (              ) to open the Actions list in a tab.
3. Open an action you want to configure for Offline Mode.
4. Select the Offline toggle.

5. Use the Hide field and Show field lists 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 lists to determine which fields 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 display instead.
7. From the Mark as zombie on screens field, search for screens to gray out when a user performs an action.

In Offline Mode, an indicator appears next to a record after a user makes a change. Setting a screen to be marked as a zombie allows the list to mimic the online behavior where a record is grayed out in the list after a user makes a change.

**Online behavior for "zombie mode"**

<table>
<thead>
<tr>
<th>INC00000057</th>
<th>5 - Planning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Trouble getting to Oregon mail server</strong></td>
<td></td>
</tr>
<tr>
<td>Assigned to</td>
<td>Beth Anglin</td>
</tr>
<tr>
<td>State</td>
<td>In Progress</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INC00000039</th>
<th>1 - Critical</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Need access to sales DB for the West</strong></td>
<td></td>
</tr>
<tr>
<td>Assigned to</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>New</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INC0010003</th>
<th>3 - Moderate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unable to access team file share</strong></td>
<td></td>
</tr>
<tr>
<td>Assigned to</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>New</td>
</tr>
</tbody>
</table>

8. Click **Update** to save the offline properties configuration.

**What to do next**

You can also configure Offline Mode for **navigation** and **smart button** functions.
Configure Offline Mode properties for function instances

Determine if the function instance available to users in online mode is also available in Offline Mode. This option gives you more control over how users manage their offline tasks.

**Before you begin**
Role required: admin

**Procedure**
1. Navigate to All and in the filter enter sys_sg_button_instance.list.
2. Select a function instance.
3. In the Offline Property area, select the Disable Offline field if you do not want this function instance to display in Offline Mode.

> Note: The default value of the Disable Offline field is empty.
4. Right-click in the header and select Save.

**Scheduled offline caching**
Configure offline caching so your field technicians can receive scheduled updates to their offline data cache. Scheduled downloads are based on the user's work schedule.

**Activate the Agent Schedule plugin**
To enable and configure scheduled offline caching, the Agent Schedule plugin [com.snc.agent_schedule] must be activated. For details on plugin activation, see [Activate a plugin](#).

**Create work schedules for offline caching**
After activating the plugin, you must create work schedules for the agents or technicians to enable users to automatically receive scheduled offline caches. This can be done directly through the Agent Work Schedules [agent_work_schedule] table. For more information on how to create schedules through this table, see [Create a work schedule for an agent or technician](#).

Based on the records from the Agent Work Schedules table, your instance runs background scheduled jobs that create schedules in the Agent Daily Schedules [agent_daily_schedule] table. Within these schedules, agents will receive a silent push sometime throughout their scheduled day.

The offline payloads that your instance generates are based on the times recorded in the Agent Daily Schedules. These can be found in the Events [sysevent] table. Use the records in the Event table to track these payloads, and info about when they are created and when they are sent to the agents.

**Scheduled Jobs associated with offline caching**

These scheduled jobs are automatically scheduled for only users who enable background downloading on their app. For information on how users can enable this feature, see [Offline mode for mobile](#).

- **Populate Agents Daily Schedule Table**
  
  This job runs once daily for all users with background downloading set to `true`.

- **Scheduled Download of Offline Payload**
This job creates an event for the first payload of the day in the [mobile_offline_payload_gen_queue] table.

Offline scheduling system properties
Use the following properties on the System Properties [sys_properties] table to configure scheduled offline caching.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.sg.offline.scheduled_download.enabled</td>
<td>Enables scheduled downloading for all Mobile Agent app users.</td>
</tr>
<tr>
<td>glide.sg.offline.scheduled_download.cachesync_offset</td>
<td>Determines when your instance starts payload generation prior to schedule. Value determines how many minutes before the user schedule payload generation begins.</td>
</tr>
<tr>
<td>Note:</td>
<td>Be sure not to set the value of this property earlier than your Mobile Offline Scheduling job is runs. Doing so will prevent offline payloads from being generated.</td>
</tr>
<tr>
<td>glide.sg.offline.scheduled_download.reminder_offset</td>
<td>Determines when your instance sends your users a daily reminder to manually download a cache. The instance sends this reminder only when there is no cache or the cache is expired. Value determines how many minutes before a user schedule your instance sends this reminder.</td>
</tr>
<tr>
<td>glide.sg.offline_payload.refresh_frequency</td>
<td>Determines refreshTimestamp frequency of payloads. Value represents this frequency in minutes. The default value is 480.</td>
</tr>
<tr>
<td>glide.sg.offline.attachment.max_total_bytes</td>
<td>Determines the maximum size allocated for all attachments in a cache. Value is represented in Megabytes (MB). The maximum value is 2048 MB.</td>
</tr>
</tbody>
</table>

For information on more offline related system properties, see, Configure Offline Mode behavior.

Mobile bar-code scanning
Use the camera on your mobile device to scan bar-codes into your instance using mobile applications.
Multi-scan bar-code scanning

Use multi-scan to scan multiple bar-codes sequentially without leaving the scanning interface. Input information for your assets faster than manual entry or individual scans. After scanning in multiple bar-codes, users can review a list of their scanned items.
Grouped bar-code scanning

Use grouped bar-code scanning to scan multiple bar-codes and group them into a set. For example, a single item that has an asset number, serial number, and model number. You can configure a button to use grouped inputs to accept multiple bar-code scans.

Supported barcode types

The ServiceNow mobile apps for iOS and Android support scanning the following barcode types:

- 2D barcodes: QR Code, Data Matrix, PDF-417, AZTEC

Zebra scanning devices

The Android version of ServiceNow mobile apps can be used with most Zebra scanning devices. The scanner works only when the Zebra device camera is off. In the app, when the cursor is in a QR/Barcode input field, the scanner outputs the results. Users can modify the barcode output to include a return/enter string at the end of each scan with other apps, such as DataWedge.

Multi-scan for mobile applications

Use multi-scan to quickly scan multiple barcodes or groups of codes. With multiscan, you can input information for your assets faster than manual entry or individual scans.

Use the scan screen to scan items into your instance by using your mobile device.
Use the scan screen to scan in your barcode items. You see this screen when you access a function that your administrator configured for barcode scanning.

### Mobile scan screen features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close (1)</td>
<td>Tap <strong>Close</strong> to leave the scan screen.</td>
</tr>
<tr>
<td>Disable camera (2)</td>
<td>Tap <strong>Disable camera</strong> to disable the camera on your mobile device. This button is available only on Android. Disable your camera to scan using a non-phone scanner.</td>
</tr>
<tr>
<td>Enable flashlight (3)</td>
<td>Tap <strong>Enable flashlight</strong> to enable the flashlight feature on your mobile device.</td>
</tr>
<tr>
<td>Scan crosshair (4)</td>
<td>Center a barcode to be scanned using the cross-hair icon in the center of the screen.</td>
</tr>
<tr>
<td>Last Scan (5)</td>
<td>See the last three entries that you scanned. Press the <strong>Rescan</strong> button to discard the scanned values.</td>
</tr>
<tr>
<td>Inputs (6)</td>
<td>See the UI parameters that are used in the scanned input. Scanning a barcode adds a value to the top parameter. Further scans add values to the next parameters in the list.</td>
</tr>
</tbody>
</table>

Last Scan: 636046316197, 636046316197
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>display order. You can tap any entry to manually enter a value.</td>
<td></td>
</tr>
<tr>
<td>Scan next item (7)</td>
<td>Tap <strong>Scan next item</strong> to accept the current scanned values and continue to a new item. If all fields in the Inputs (6) section have values, the scan screen automatically continues to the next item. If there are any mandatory fields without values, you can't tap <strong>Scan next item</strong>.</td>
</tr>
<tr>
<td>Review (8)</td>
<td>Tap the review button to review your scanned items. See the next section for details on the review screen.</td>
</tr>
</tbody>
</table>
## Review screen

After scanning your items, use the review screen to review, delete, and submit scanned items.

### Mobile review screen features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submit (1)</td>
<td>Tap <strong>Submit</strong> to save your scanned items.</td>
</tr>
<tr>
<td>Scanned items (2)</td>
<td>See your scanned items in this section. Each item displays the UI parameter name and its scanned value.</td>
</tr>
<tr>
<td>Delete item (3)</td>
<td>Tap the <strong>delete</strong> icon (🗑️) next to a scanned item to remove that item.</td>
</tr>
</tbody>
</table>

### Configure a grouped input for multiple scans

Enable your users to scan several barcodes sequentially. Users can scan barcodes without leaving the scanning interface when you configure a grouped input. You can configure grouped inputs on buttons to accept multiple barcode scans.

**Before you begin**

**Role required:** admin
Procedure
1. In the application navigator, enter `sys_sg_button` to open a list of mobile functions.
2. Open the record for the button where you want to include multi-scan functionality.
3. Click the **Grouped Inputs** tab on the form to display the grouped inputs list.
4. Click **New**.
5. On the form, fill in the fields.

### Grouped input form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of your grouped input. Because this name is used in scripts, do not include spaces in names.</td>
</tr>
<tr>
<td>Label</td>
<td>Label to describe your grouped input. You may use spaces in your label.</td>
</tr>
<tr>
<td>Source</td>
<td>Source for the grouped input. Select <strong>Button</strong>.</td>
</tr>
<tr>
<td>Button</td>
<td>Button used for the grouped input. The <strong>Button</strong> field appears after you select <strong>Button</strong> in the <strong>Source</strong> field.</td>
</tr>
<tr>
<td>Order</td>
<td>Order value for your grouped input.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>Field that you enable to require that at least one value is entered for this input.</td>
</tr>
<tr>
<td>Multiple entries</td>
<td>Ability to scan multiple entries.</td>
</tr>
<tr>
<td>Max entries</td>
<td>Maximum number of entries for this input. Enter 0 for an unlimited number of entries. This field is visible only when the <strong>Multiple entries</strong> field is enabled.</td>
</tr>
</tbody>
</table>

Create UI parameters for your grouped input
Create UI parameters to contain the values that your users scan in with your grouped input.

**Before you begin**
Role required: admin

**Procedure**
1. In the **UI Parameters** field, click the lookup icon (🔍) to display the UI parameters list.
2. Click **New**.
3. On the form, fill in the fields.

### UI parameter form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Descriptive name of your UI parameter</td>
</tr>
<tr>
<td>Parameter type</td>
<td>Type of parameter. Select <strong>None</strong></td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Input type</td>
<td>Type of entry for the parameter value. Select <strong>QR/Barcode</strong></td>
</tr>
</tbody>
</table>

4. Click **Submit**.

5. **Optional:** Repeat steps 1 through 4 to add additional parameters.
   For example, if you are creating a record to track hardware, you may include UI parameters for a serial number and a model number.

6. Click **Submit**.

**Configure your action item to use your grouped input UI parameters**

Configure the action item on your button so that you can use the values that your users scan in.

**Before you begin**

Role required: admin

**Procedure**

1. From your grouped input record, click the reference icon (🛂) next to the **Button** field, and then click **Open record** to open your function record.

2. From your function record, click the reference icon (🛂) next to the **Action item** field, and then click **Open record** to open your action item record.

3. Modify the **Execution Script** field to use your grouped input UI parameters.
   These parameters are contained within the input object. You can access the parameters by using this format: `input.grouped_input_name[input index]["UI Parameter Name"]`. For example, if you have a grouped input called `add_asset` that has a UI parameter named `Serial number`, you would use `input.add_asset[0]["Serial number"]` to access the first serial number asset.

   In this example, the script checks the length of `input.add_asset` to see how many items were scanned. It then adds an asset for each scanned item using the scanned-in serial number and asset tag.

   ```
   function WriteBackAction(input) {
       var assetString = input.add_asset[0]["Asset tag"];
       var serialNumber = input.add_asset[0]["Serial number"];
       var SMAssetUsage = new global.SMAssetUsage;
       SMAssetUsage.addAssetToStockroom(assetString, serialNumber);
   }
   `(input);
   ```

   **Note:** The action item type must be set to **Script** to display this **Execution Script** field.

**What to do next**

Your button is configured for multi-scan, and your users can begin creating records. For information on using the mobile scanning interface, see **Multi-scan for mobile applications**.
Localization on mobile devices

ServiceNow mobile apps are localized in 20 languages.

Native localization

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 the following languages:

- I18N: Brazilian Portuguese Translations
- I18N: Czech Translations
- I18N: Dutch Translations
- I18N: Finnish Translations
- I18N: French Canada Translations
- I18N: French Translations
- I18N: German Translations
- I18N: Hungarian Translations
- I18N: Italian Translations
- I18N: Japanese Translations
These translations can't be customized since they ship with the application binary and translated natively on the device according to the language settings for the device.

**Server-side localization**

Server-side localization: Controlled the same way as desktop web localization (server system language / user preference on server). Localized components on the server include things like field labels, web content, and other data 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. Translated these elements using the translated name/field table on your instance. For more details on this table, see Translated Name / Field table

⚠️ **Note:** Right-to-left languages are not supported in the native mobile app.

This series of screenshots shows a mobile app on a mobile device with Spanish set as its native language. The device is connected to a ServiceNow instance where the instance language is set to English. The user's mobile device translates the Spanish portions the text according to the device's user preferences. These elements cannot be changed with your instance localization settings.

The English portions are not translated by the mobile device. The English text can be translated using your instances localization settings. As a admin you have control over how these elements are translated.
Create translation records for mobile

Create records on the Translated Name / Field table to translate elements of your mobile app that are not natively translated.

**Before you begin**
Role required: admin
Use these steps to add translations for your mobile device that are not already translated by your mobile device’s native translation settings. To see which elements need to be manually translated, switch your mobile device’s native language to the language you are working on, and view your mobile apps to see which values remain in English. Use the steps below to translate the remaining values.

**Procedure**
1. Navigate to All > System Localization > Translated Name / Fields > .
2. Click New to create a new Translated Name / Fields record.
3. On the form, fill in the fields.
UI elements form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label (translate)</td>
<td>Translated text that users see on the translated item.</td>
</tr>
<tr>
<td>Language</td>
<td>Two-character ISO language code for this translated text.</td>
</tr>
<tr>
<td>Value</td>
<td>Original English value that causes this translated text to be displayed.</td>
</tr>
<tr>
<td>Table</td>
<td>Select a table depending on the element to be translated:</td>
</tr>
<tr>
<td></td>
<td>• For Applet names, use <code>sys_sg_screen</code></td>
</tr>
<tr>
<td></td>
<td>• For Applet launcher names, use <code>sys_sg_applet_launcher</code></td>
</tr>
<tr>
<td></td>
<td>• For Function names, use <code>sys_sg_button_instance</code></td>
</tr>
<tr>
<td></td>
<td>• For global search placeholder text, use <code>sys_sg_global_search</code></td>
</tr>
<tr>
<td></td>
<td>• For search source names, use <code>sys_search_source</code></td>
</tr>
<tr>
<td>Element</td>
<td>The element on the table to be translated. See the table below for element names for each translatable element.</td>
</tr>
</tbody>
</table>

4. Click Submit.
5. Repeat the steps to create a record for each language you want to support on your instance.

Translation values for mobile element

Use this reference to find which values to use to translate your mobile elements

Translation values

<table>
<thead>
<tr>
<th>Mobile Element</th>
<th>Table name</th>
<th>Element name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applet</td>
<td><code>sys_sg_screen</code></td>
<td>name</td>
</tr>
<tr>
<td>Applet launcher</td>
<td><code>sys_sg_applet_launcher</code></td>
<td>title</td>
</tr>
<tr>
<td>Function instance</td>
<td><code>sys_sg_button_instance</code></td>
<td>label</td>
</tr>
<tr>
<td>Function cancel button</td>
<td><code>sys_sg_button</code></td>
<td>cancel_label</td>
</tr>
<tr>
<td>Function confirm button</td>
<td><code>sys_sg_button</code></td>
<td>confirm_label</td>
</tr>
<tr>
<td>Global search placeholder text</td>
<td><code>sys_sg_global_search</code></td>
<td></td>
</tr>
<tr>
<td>Media section headline</td>
<td><code>sys_sg_section</code></td>
<td>headline</td>
</tr>
<tr>
<td>Media section text</td>
<td><code>sys_sg_section</code></td>
<td>text</td>
</tr>
<tr>
<td>Mobile section</td>
<td><code>sys_sg_section</code></td>
<td>title</td>
</tr>
</tbody>
</table>
This example shows a translation record for an applet called **My Tasks** translated into **Spanish**.

### Domain separation and mobile

Domain separation is supported in mobile. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

**Support level: Basic**

- Business logic: Ensure that data goes into the proper domain for the application’s service provider use cases.
- The application supports domain separation at run time. The domain separation includes separation from the user interface, cache keys, reporting, rollups, and aggregations.
- The owner of the instance must set up the application to function across multiple tenants.

Sample use case: When a service provider (SP) uses chat to respond to a tenant-customer’s message, the client must be able to see the SP’s response.

For more information on support levels, see [Application support for domain separation](#).

### Overview

The mobile platform supports domain separation for all native clients. The mobile UI design clearly indicates the domain that a record is associated with.

Before extending the domain separation functionality to mobile, the feature must be enabled on the platform web-based interface. For further information on configuration, see [Domain separation setup and administration](#).

---

**Translation values (continued)**

<table>
<thead>
<tr>
<th>Mobile Element</th>
<th>Table name</th>
<th>Element name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigation tab</td>
<td>sys_sg_navigation_tab</td>
<td>label</td>
</tr>
<tr>
<td>UI parameter name</td>
<td>sys_sg_ui_parameter</td>
<td>name</td>
</tr>
<tr>
<td>UI parameter placeholder</td>
<td>sys_sg_ui_placeholder</td>
<td>placeholder</td>
</tr>
</tbody>
</table>

---

© 2022 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.
How domain separation works in mobile

You can use the company or account fields to display appropriate records by domain. These fields are available in tables when the domain separation plugin is enabled. Because each company or account is linked to a single domain, when you create a record, you can use these fields to specify the domain in which you want to create the record.

After the feature has been set up through the platform web-based interface, a two-part process is required to further extend the domain separation functionality on mobile. For additional information about configuring domain separation on mobile devices, contact Customer Service and Support.

⚠️ Note: ITSM Mobile Agent contains an automatically configured domain separation feature. For more information, refer to the ITSM mobile agent documentation.

Related reference

Domain separation for service providers

ServiceNow Classic mobile app

Use the ServiceNow Classic mobile app to access records, update information, and collaborate with other users.

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.
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
- 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.

**Migrate to ServiceNow mobile**

The ServiceNow Classic app no longer receives enhancements or non-priority bug fixes. Migrate to ServiceNow® mobile to take advantage of features such as rapid development, offline capability, and integration with native mobile device features. For more detail on the migration process, see [Migrate from the ServiceNow Classic mobile app to the ServiceNow Mobile Platform](#).

**Device security for ServiceNow Classic**

This document applies to the ServiceNow Classic app for iOS and Android. This document may be subject to change for future mobile releases and re-platforming efforts.

**Components and architecture**

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 Mobile single sign on.

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 256 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 Classic 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
- BlackBerry
- Citrix
- Intune
- Jamf Pro
- IBM
- MobileIron

Data security for ServiceNow Classic
The ServiceNow Classic 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 256.

Data in motion
Data in motion is over a secure SSL/TLS channel and encrypted with HTTPS.

Offline 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.

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
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 Customer Service and 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 (MFA).

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 in mobile browsers and the ServiceNow Classic mobile app.

Before you begin
Role required: admin

Note: This feature is only available in mobile browsers and the ServiceNow Classic mobile app.

About this task
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.

Procedure

   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.

What to do next
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.

About this task
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.

Procedure

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>
</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 menu. For example, an entry of 100 would place this module before one with an <strong>Order</strong> entry of 200.</td>
</tr>
<tr>
<td>Application menu</td>
<td>Displays the application menu from which you accessed this screen. Select a different application menu, if appropriate.</td>
</tr>
<tr>
<td>Table</td>
<td>Select the table for this module.</td>
</tr>
<tr>
<td>Updated</td>
<td>Displays the date and time when the module record is updated.</td>
</tr>
<tr>
<td>Roles</td>
<td>Click the lock icon and select the roles for this module. Only users with the designated roles can access this module.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box to activate this module. Only active modules appear in the application menu.</td>
</tr>
<tr>
<td>Filter</td>
<td>Create a filter for identifying which fields this module uses from the selected <strong>Table</strong>.</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>• <strong>type_of_link</strong>: determines what kind of page opens, for example, form, list, or view.</td>
</tr>
<tr>
<td></td>
<td>• <strong>table</strong>: Refers to the table being referenced. For example, <strong>incident</strong>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>parameters</strong>: Any additional information you want to use to direct the user to 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>$type.do</code>.</td>
</tr>
<tr>
<td>Path Relative To Root</td>
<td>Allows mobile navigator modules that link to desktop pages. Select this check box to use mobile URL in the Path field, such as <code>$sp.do</code>, <code>$chat.do</code>, or <code>$vtb.do</code>.</td>
</tr>
<tr>
<td></td>
<td>When the Path Relative To Root is cleared, the path is considered a mobile specific path and is relative to <code>$m.do</code>.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

**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.
**Mobile list view**

<table>
<thead>
<tr>
<th>PROBLEMS</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority in (1 - Critical, 2 - High, 3 - Moderate)</td>
<td></td>
</tr>
</tbody>
</table>

Please remove this hotfix
2 - High
High Priority!

**Missing file dependency in SuccessFactors Workforce Analytics**

**Display title**

**Extras**

Performance issue in the custom extension written for Workday Payroll
3 - Moderate
Carry on

Bad data causing page failure in HireRight Reports

**Procedure**

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*. 
Example:
This example shows a list of 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.

Before you begin
Role required: admin

About this task
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.

Procedure
1. Navigate to All > 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.

**Before you begin**

Role required: admin

**Procedure**

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.

**Results**

<table>
<thead>
<tr>
<th>Display title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance issue in the custom extension written for Workday Payroll</td>
</tr>
<tr>
<td><strong>3 - Moderate</strong></td>
</tr>
<tr>
<td>Carry on</td>
</tr>
</tbody>
</table>

### Configure field status indicators for the ServiceNow Classic mobile app

Differentiate items in a list by configuring the field status indicators.

**Before you begin**

Role required: admin
Procedure
1. Navigate to All > 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.

Before you begin
Role required: admin

Procedure
1. Navigate to All > 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></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 2 years
I got this

Don Goodliffe 2 years
Task reassigned due to employee termination
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form</td>
<td>Opens the list item in the form view.</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.

#### Procedure

1. Navigate to **All > 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.

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.

Procedure

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.

### Results
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.

© 2022 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.

**Before you begin**
Role required: admin

**Procedure**
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 <strong>Configure &gt; List Control</strong>.</td>
</tr>
<tr>
<td>List v3</td>
<td>Open the list title menu and select <strong>List Control</strong>.</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.

Before you begin
Role required: admin

About this task
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.

Procedure
1. Navigate to All > 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.
## Home Page Collections fields

<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 collect. 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.

## Mobile home page fields

<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>
| Header image  | Optional. Select an image to appear at the top of the mobile home page. A header image lets you visually communicate the purpose of the home page. Note the following guidelines:
  - Use a high-resolution image that looks good at different sizes and aspect ratios. The display size and aspect ratio of a header image vary by the device screen size. Since it is not possible to specify a... |
Mobile home page fields (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fixed display size for a header image, use a high-quality image that works for different screen sizes. • Avoid text on a header image. Text can be cropped or appear blurry when the image is scaled for different screen sizes. Also, text is not localized because the header image is a static file. Use the home page title and subtitle properties to include any text on a home page. • Avoid using a company logo as a header image. Ideally a header image artistically represents the purpose of the home page. If a logo is needed, create a custom mobile theme (sys_ui_mobile_theme) with a title image that includes the company logo in the navigation bar of the home screen. • For domain separation, header images uploaded in the parent domain do not appear for users in the child domain. Either use a global domain or the child domain that the users are in.</td>
</tr>
</tbody>
</table>

Roles | Configure a home page by role. |

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 &quot;app-like&quot; 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.

#### Before you begin
Role required: admin

#### Procedure
1. Navigate to All > System Classic Mobile UI > Themes.
2. Click an existing theme to edit, or click New.
3. Complete the mobile theme form fields.
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Title for the theme. This does not appear anywhere for end users.</td>
</tr>
<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 Classic 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.

About this task
Connect Chat has only been optimized for the ServiceNow Classic mobile app, not the mobile web.

Procedure
1. Navigate to All > 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.

What to do next
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)

**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.

⚠️ **Note:** Setting a redirect on the UI action through `action.setRedirectURL();` takes precedence over the `navigate_back` flag.

### 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.

**Procedure**

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**.

---

<table>
<thead>
<tr>
<th>Name</th>
<th>Active</th>
<th>Table</th>
<th>UI Type</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Selection Required&quot; label for Checkboxes</td>
<td>true</td>
<td>Variable [item_option_new]</td>
<td>Desktop</td>
<td>Global</td>
</tr>
<tr>
<td>&quot;Selection Required&quot; label for Checkboxes</td>
<td>true</td>
<td>Variable [item_option_new]</td>
<td>Desktop</td>
<td>Global</td>
</tr>
<tr>
<td>BPPI Close Mandatory on Close or Resolve</td>
<td>true</td>
<td>Incident [incident]</td>
<td>Desktop</td>
<td>Global</td>
</tr>
<tr>
<td>BPPI Hide Attachment Link when Closed</td>
<td>true</td>
<td>Incident [incident]</td>
<td>Desktop</td>
<td>Global</td>
</tr>
<tr>
<td>BPPI Hide Choice - Closed</td>
<td>true</td>
<td>Incident [incident]</td>
<td>Desktop</td>
<td>Global</td>
</tr>
<tr>
<td>BPPI Hide Close Notes and Code</td>
<td>true</td>
<td>Incident [incident]</td>
<td>Desktop</td>
<td>Global</td>
</tr>
<tr>
<td>BPPI Set Location to User</td>
<td>true</td>
<td>Incident [incident]</td>
<td>All</td>
<td>Global</td>
</tr>
</tbody>
</table>

### 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>Use the new mobile methods</th>
<th>Several new methods are available for modifying form fields instead of directly manipulating the HTML. These methods replace previous usages of <code>g_form.getControl()</code>, which is deprecated for the mobile platform. In your existing scripts, ensure that the new methods are used in place of methods that are not valid on the mobile platform. For information on these new methods, refer to Mobile GlideForm() API.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not use deprecated methods</td>
<td>The following methods have been deprecated for the mobile platform because direct access to HTML elements is not allowed:</td>
</tr>
<tr>
<td></td>
<td><code>g_form.getControl()</code></td>
</tr>
<tr>
<td></td>
<td><code>g_form.getFormElement()</code></td>
</tr>
<tr>
<td></td>
<td><code>g_form.getElement()</code></td>
</tr>
</tbody>
</table>
**Requirements (continued)**

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.

<table>
<thead>
<tr>
<th>Do not reference unsupported browser objects</th>
<th>The following browser objects are not supported in mobile scripts:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Window</td>
<td></td>
</tr>
<tr>
<td>• jQuery or Prototype ($, $j, or $$)</td>
<td></td>
</tr>
<tr>
<td>• Document</td>
<td></td>
</tr>
</tbody>
</table>

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>
</tbody>
</table>

Be sure that all `g_form.getReference()` calls include the callback parameter. For example, the following script does not include a callback and is incompatible with the mobile platform:

```javascript
var userName = g_form.getReference('assigned_to').user_name;
g_form.setValue('u_assigned_user_name', userName);
```

The following script has been updated to include the callback and is compatible with the mobile platform:

```javascript
var userName = g_form.getReference('assigned_to').user_name;
g_form.setValue('u_assigned_user_name', userName, function(now_GR) {  
g_form.setValue('u_assigned_user_name', now_GR.user_name);
});
```

<table>
<thead>
<tr>
<th>Do not make synchronous Ajax calls</th>
<th>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 AJAX. For information on the available GlideAjax methods, refer to the GlideAjax API.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Do not make synchronous GlideRecord calls</th>
<th>The mobile platform does not allow synchronous GlideRecord calls. Make sure that any existing GlideRecord calls include a callback. For example, the following script does not include a callback and is incompatible with the mobile platform:</th>
</tr>
</thead>
</table>
|                                          | `var now_GR = new GlideRecord('incident');
gr.addQuery('number', g_form.getValue('related_incident'));
gr.query();
gr.next();
g_form.setValue('u_related_incident_description', now_GR.short_description);` |
Requirements (continued)

The following script has been updated to include the callback, and is compatible with the mobile platform:

```javascript
var now_GR = new GlideRecord('incident');
gr.addQuery('number', g_form.getValue('related_incident'));
gr.query(function(now_GR) {
    gr.next();
    g_form.setValue('u_related_incident_description',
    gr.short_description);
});
```

<table>
<thead>
<tr>
<th>Do not use methods unavailable on the mobile platform</th>
<th>Due to the limitations and reduced functionality that is imposed by the mobile platform, the following methods are not deprecated but are not available on the mobile platform. If these run on the mobile platform, no action occurs:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• showRelatedList ()</td>
<td>• showRelatedList ()</td>
</tr>
<tr>
<td>• hideRelatedList ()</td>
<td>• hideRelatedList()</td>
</tr>
<tr>
<td>• showRelatedLists ()</td>
<td>• flash()</td>
</tr>
<tr>
<td>• hideRelatedLists()</td>
<td>• getSections()</td>
</tr>
<tr>
<td>• flash()</td>
<td>• enableAttachments()</td>
</tr>
<tr>
<td>• getSections()</td>
<td>• disableAttachments()</td>
</tr>
<tr>
<td>• enableAttachments()</td>
<td>• setReadonly() (Note that setReadonly() is available)</td>
</tr>
<tr>
<td>• disableAttachments()</td>
<td>• getParameter()</td>
</tr>
</tbody>
</table>

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.

**Procedure**

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.

**Example**

![UI Policy Configuration](image)

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.

**Before you begin**
Role required: admin

**About this task**
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.
Procedure

1. Navigate to **All > 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 <em>configured to appear in a mobile device</em>. 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.

**Before you begin**
Role required: admin

**Procedure**
1. On a desktop, navigate to the form where you want to add the location or barcode scanner.
2. Use the form context menu on the left side of the form header (≡) 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**.

**Results**

The added fields appear on the form in the native mobile app. This feature is not available on the mobile web. Users must 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.

**About this task**

Navigate to **System Properties > Mobile UI Properties** to enable or disable any of the properties below.

**UI Agent Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable mobile web UI</td>
<td>glide.ui.m_enabled</td>
<td>Turns on/off the mobile web UI. Even if this property is disabled, users still have access to the mobile web.</td>
</tr>
<tr>
<td>Enable native mobile applications (Android, iOS)</td>
<td>glide.ui.m.native_apps_enabled</td>
<td>Enables/disables user access to the ServiceNow Classic mobile app. This property only applies to the classic mobile app.</td>
</tr>
<tr>
<td>Use the mobile web UI if one of these strings (comma-separated) appears in the browser's User Agent</td>
<td>glide.ui.m_agents</td>
<td>Define which web UI to use depending on what name appears in a browser 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. The mobile UI varies depending on your device, the way your administrator has configured the mobile UI for your company, and whether you access your instance from the ServiceNow Classic mobile app or mobile browser.

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 Apple App 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.

Related information
Mobile activity streams in the ServiceNow Classic
Mobile filters in the ServiceNow Classic app
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 Apple App 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.

**Procedure**

1. After you download the app from the Apple App Store or the Google Play store, tap the icon on the homepage.
   
   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.

**What to do next**

After logging in, use the mobile app to perform tasks on your instance. For more information on using ServiceNow Classic mobile app, see *Using the ServiceNow Classic mobile app*.

**Access an instance from a mobile browser**

Access an instance using a supported mobile browser.

**About this task**

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.
Procedure

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.
Procedure
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.

About this task
To replace the default icon:

Procedure
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.
Procedure

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.

**What to do next**
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.

**About this task**
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.

**Related information**

Enable an application menu for the ServiceNow Classic mobile app

**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.
Procedure

1. To open the application navigator, tap the menu icon ( ).

Tablet application navigator

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.

Related information

Enable an application menu for the ServiceNow Classic mobile app

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.
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>

**Related information**

Customize the list title for the ServiceNow Classic mobile app
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**
Activity stream updates consist of the following elements.

- Back button
- User
- Activity details
- Record details
**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.

**Related information**

Activity streams in list view

**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.

**iPhone UI filter**

1. **Cancel**
2. **11 Total Results**
3. **Save**
4. **Add New Condition**
5. **Active = true**
6. **Assigned to is not empty**
7. **Category is Inquiry / Help**
8. **Category is Network**
9. **Add New Condition**

**Android UI filter**

1. **← 42 total results**
2. **1 SET 1 RESULTS**
3. **NEW FILTER**
4. **ADD CONDITION**
5. **Active = true**
6. **Assigned to is not empty**
7. **Category is Inquiry / Help**
8. **Category is Network**

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

<table>
<thead>
<tr>
<th>Condition set</th>
<th>Rule type</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All</td>
<td>Active = true</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assigned to = ITIL User</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Add New Condition</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Condition set</th>
<th>Rule type</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Any</td>
<td>Category = Inquiry / Help</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Category = Request</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Category = Network</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Add New Condition</td>
</tr>
</tbody>
</table>
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.

About this task
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:

Procedure
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.

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
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.

**Procedure**

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, access the search field from the List options menu (三).
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>

**Related information**

Customize the list title for the ServiceNow Classic mobile app

**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

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.

Related information

Activity streams in list view
Filters for ServiceNow Classic on a tablet

Construct complex filters with the ServiceNow Classic mobile app UI.

Mobile filters consist of the following elements.

<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>
</tbody>
</table>
Mobile UI filters (continued)

<table>
<thead>
<tr>
<th>Number</th>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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.

**Related information**

- View or modify a mobile list in the ServiceNow Classic app
- Mobile condition sets

**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.

Elements of the mobile UI home

<table>
<thead>
<tr>
<th>Number</th>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User profile button</td>
<td>Displays the current profile record.</td>
</tr>
<tr>
<td>2</td>
<td>Favorites</td>
<td>Links to records in the instance, sometimes with visualizations.</td>
</tr>
<tr>
<td>3</td>
<td>Application navigator button</td>
<td>Displays the list of menus and modules available to the current user.</td>
</tr>
</tbody>
</table>
• 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.

Procedure
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.

Related information
Edit a mobile favorite

Edit a mobile favorite
Make changes to the appearance of a favorite on a mobile favorites page.
**Procedure**

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.

**Related information**

   Add a mobile favorite

**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.

**About this task**

Visualizations allow users on the mobile app UI to see relevant table information directly from the home page favorite icon.

**Procedure**

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.

**Visualization options**

Use 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>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</td>
<td>Bar chart</td>
</tr>
<tr>
<td>when you set up the visualization.</td>
<td></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 Apple App 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.

**Related information**

**Push notifications**

**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.

**Before you begin**

Role required: none

**About this task**

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 UI16.

**Procedure**

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>
<tr>
<td></td>
<td>List of notification categories. Each category identifies and groups related notifications. To select a notification:</td>
</tr>
<tr>
<td></td>
<td>a. Tap the notification category.</td>
</tr>
<tr>
<td></td>
<td>b. Tap the notification to be updated.</td>
</tr>
<tr>
<td></td>
<td>c. Enable or disable the channels for that notification.</td>
</tr>
</tbody>
</table>

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>

Related information
Connect

Mobile Connect Chat conversations
Have conversations with groups of people in Connect in the ServiceNow Classic mobile app.
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>
</tbody>
</table>
Elements of the mobile UI: Connect Chat conversation (continued)

<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message field</td>
<td>Enables you to enter and send messages.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> For record conversations, all messages are sent as comments.</td>
</tr>
<tr>
<td></td>
<td>Sending a message as a worknote is not supported.</td>
</tr>
</tbody>
</table>

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.

**Related information**

Dictionary attributes

**Enable mobile location and barcode scanning for the ServiceNow Classic mobile app**
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

![Barcode scanning field diagram]

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

Related information

Dictionary attributes

Enable mobile location and barcode scanning for the ServiceNow Classic mobile app

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>Android™ mobile devices</td>
<td>All models of Android phone running KitKat (4.4) and above.</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 Now Community or other Service Portal-based pages.

**Disable the ServiceNow Classic mobile app**

Disable ServiceNow Classic on your instance to prevent users from using the legacy app and guide users toward the new ServiceNow mobile experience.

**Disable ServiceNow Classic on your instance**

Access to your instance using the ServiceNow Classic mobile app is controlled by the `glide.ui.m.allow_classic_mobile_app` system property.

This system property defaults to a value of `false` (blocked) on new instances, beginning with the Paris release.

For upgraded instances with existing ServiceNow Classic users, the property defaults to `true` (allowed). Your instance makes this determination based on the presence of records on the Mobile Devices [sys_mobile_devices] table. If there are records on this table, the instance sets the property value to `true` (allowed). With the property set to true, there will be no impact to your ServiceNow Classic users.

The `glide.ui.m.allow_classic_mobile_app` system property cannot be manually configured by an administrator. Contact Customer Service and Support to change this property from its default.
Configure the error message text for disabled ServiceNow Classic apps

Customize the message that appears on the ServiceNow Classic app when the Classic UI is disabled on your instance.

Before you begin
Role required: admin

Users accessing your instance with the ServiceNow Classic app will see a message if the classic UI is disabled. This message includes a title, a text area for a brief message, and a link.

Customize this information using records on the Messages [sys_ui_message] and System Properties [sys_properties] tables.

Procedure
1. Navigate to All > System UI > Messages.
2. In the Messages list, the message record for the portion of the error message you want to change.

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update required</td>
<td>The title that appears at the top of the message</td>
</tr>
<tr>
<td>A new version of the ServiceNow app is now available, visit the App Store to update.</td>
<td>The text that appears below the message title for iOS users.</td>
</tr>
<tr>
<td>A new version of the ServiceNow app is now available, visit the Play Store to update.</td>
<td>The text that appears below the message title for Android users.</td>
</tr>
<tr>
<td>Go to App Store</td>
<td>The text of the link that appears at the bottom of the message for iOS users</td>
</tr>
<tr>
<td>Go to Play Store</td>
<td>The text of the link that appears at the bottom of the message for Android users</td>
</tr>
</tbody>
</table>

3. Open the record.
4. Update the text in the Message field. The value of this field is the text that will be displayed in the error message in place of the text that appears in the Key field.
5. Click Save.
6. Repeat steps 2 through 5 as needed, to modify your message.

Configure the error message link for disabled ServiceNow Classic apps

Customize the link that appears at the bottom of the ServiceNow Classic app, when the classic UI is disabled on your instance.
Before you begin
Role required: admin
By default, the link at the bottom of the message directs your users to the Mobile Agent app on the Apple App store or Google Play store, depending on their devices operating system. You can change the value of these system properties to direct users to an alternate URL.

Procedure

1. Type sys_properties.list in the Application Navigator.
2. In the System Properties list, find and open the glide.sg.agent_app_apple_store_url system property.
3. In the Value field, replace the default value with the URL of your choice. This URL is used for users using iOS mobile devices.
4. Click Save.
5. In the System Properties list, find and open the glide.sg.agent_app_play_store_url system property.
6. In the Value field, replace the default value with the URL of your choice. This URL is used for users using Android mobile devices.
7. Click Save.
Index

A
action function 219
action item 219
AirWatch EMM 89
AirWatch MDM 89
AirWatch mobile device management 89
applet 219

B
BlackBerry MDM 93
BlackBerry mobile device management 93
BlackBerry UEM MDM 93
BlackBerry Unified Endpoint Management mobile device management 93

C
components and architecture 70, 639
configure a mobile theme 661
configure mobile classic location and barcode scanning 670
customize mobile homepages 658

E
e-signature for mobile approvals 646
enterprise mobility management (EMM) 80

F
functions 219

I
IBM Maas360 EMM 100
IBM Maas360 enterprise mobility management 100
IBM MaasS360 iOS distribution 100
IBM Maas360 MDM 100
IBM Maas360 mobile device management 100
Intune mobile device management 102

M
mobile action 196
action item parameters 522
action tutorial 196
button actions 511
configure employee directory applet 439
data item parameters 324, 327
home page collections 658
list applet 358
list search 686
list title 649
map applet 442
navigation functions 532
offline mode 621
Offline Mode 619
parameterized action functions 523
parametrized navigation function 208
roles and permissions 590
URL applet 455
mobile 3D touch 668
mobile advanced list customization 651
mobile app configure Connect 663
controls and practices  
642
data flow retrieval  
640
data security  
642
identity and access  
management  
70, 639
information storage  
72, 640
internal mobile app distribution  
641
security controls  
76, 642
security patching  
643
session length  
640
session length and timeout  
640
shared data  
643
timeout  
640
user authentication  
70, 639
user data collection  
643
user termination  
640
mobile SSO  
645
mobile studio  
applet  
351
create action item  
519
smart button function  
545
mobile Studio  
action functions  
514
mobile tested devices  
113
mobile theme configuration  
661
mobile tutorial  
applet parameter configuration  
191
data item with parameters  
191
parameterized navigation  
203
mobile UI  
169
activity streams  
680, 688
application navigator  
677
barcode scanning  
700
condition sets  
684
Connect Chat  
conversations  
698
device location  
700
favorites  
691
filters  
683, 689
get started for end users  
671
home page  
favorite  
693
list  
685
list configuration  
655
lists  
679, 687
mobile app  
696, 696
visualizations  
694, 695
mobile UI styles  
486, 496
MobileIron MDM  
110
MobileIron mobile device management  
110
N
navigation function  
219
notification settings  
696
O
offline mode  
mobile  
621
Offline Mode  
mobile  
619
S
ServiceNow Agent  
187
set up  
mobile experience admin  
187, 643
smart button function  
219
smartphone interface  
blackberry
tablet app
  application navigator
tablet support
170