MobileIron Core 10.8.0.0 Apps@Work Guide

August 31, 2020

For complete product documentation see MobileIron Core Product Documentation Home Page

Contents

New features and enhancements

This guide documents the following new features and enhancements:

- **Mobile@Work client no longer supports in-house apps for Managed device with Work profile mode on Android 11 devices:** Upon upgrade to Android 11, the MobileIron Mobile@Work client no longer supports in-house apps for devices that migrate from Work Profile mode to Work Profile on Company Owned Devices mode. This also applies to new Android 11 devices provisioned as Work Profile on Company Owned Devices. For more information, see the following:
  - Features specific to Android enterprise apps
  - Adding in-house apps for Android
  - Public and private Android enterprise app deployment
  - *MobileIron Core Device Management Guide for Android and Android enterprise Devices*
- **AppConfig XML Upload:** For an iOS app in the App Catalog, administrators can add a managed app configuration from one of the following:
  - AppConfig Community - This option is available for the apps that have an app config specification in the community repository
  - Upload .xml spec - Use this option to upload an XML schema to push a particular version of app configuration for the app.:

  For more information, see Adding a new managed app setting for an app
- **Support for freeze period in system update:** Administrators can now freeze firmware updates for up to 90 days. This is helpful if your company needs time to figure out the migration plan for changing from Managed device with Work Profile mode to Work Profile for Company Owned Device mode. Applicable to Android 11 devices in Device Owner mode and Android 9+ devices in Managed Device with Work Profile mode.
  - For more information, see Adding in-house apps for Android.
  - Also see "Setting the system update policy for Android devices" in the *MobileIron Core Device Management Guide for Android and Android enterprise Devices.*

Managing mobile apps with Apps@Work

This section addresses Apps@Work and the tools provided for distribution and management of mobile apps.

About managing mobile apps

Apps@Work provides the tools for distributing and managing mobile apps. You can use Apps@Work tools to facilitate installation of standard corporate apps, as well as to help regulate the apps that your users are bringing into the enterprise. Apps@Work tools consist of:

- App Catalog (previously called “app distribution library”)
- App Control
- Installed Apps (previously called “device app inventory”)

What is the App Catalog?

The **App Catalog** is a centralized location for the business apps you want to manage for your users. App distribution is customized for each supported platform, and allows you to set granular policies per app. By uploading apps to the App Catalog, you can make private apps available for users to download on their devices. You can also add external apps and distribute them to users, making it clear to employees that the apps are approved for download and supported.

With the App Catalog, you can:

- Include apps from the Apple Store, Google Play Store, or Windows Store.
- Upload in-house apps to the App Catalog.
- If your Core is enabled for Android enterprise, include private apps for Android enterprise devices that are hosted on Google Play Store for your domain.

You can then make these apps available for users to download with Apps@Work on their devices.

The Apps@Work home page on the device consists of three rows. Each row is separate section made up of:

- New Releases
- Featured Apps
- Categories of Apps

You can easily identify apps selected by the administrator as a featured app. These apps are displayed in a banner at the top of the screen. Swipe the banner to scroll through the featured apps. Featured apps are also displayed in a section listing the apps in a row on the Apps@Work home screen. Tap **More** to see all the apps in the that section. you can also tap the search icon at the top to search for an app.

To list an app in the Featured app banner in the Apps@Work home page:

1. Go to **Apps > App Catalog**, on the Admin portal and select an app and click on 'Edit', 'Add' or views the app details.
2. Choose to feature the app in the Featured Banner
   - Add a description. The description is blank by default.
   - Choose a background color or style for the banner
3. Click **Finish**.

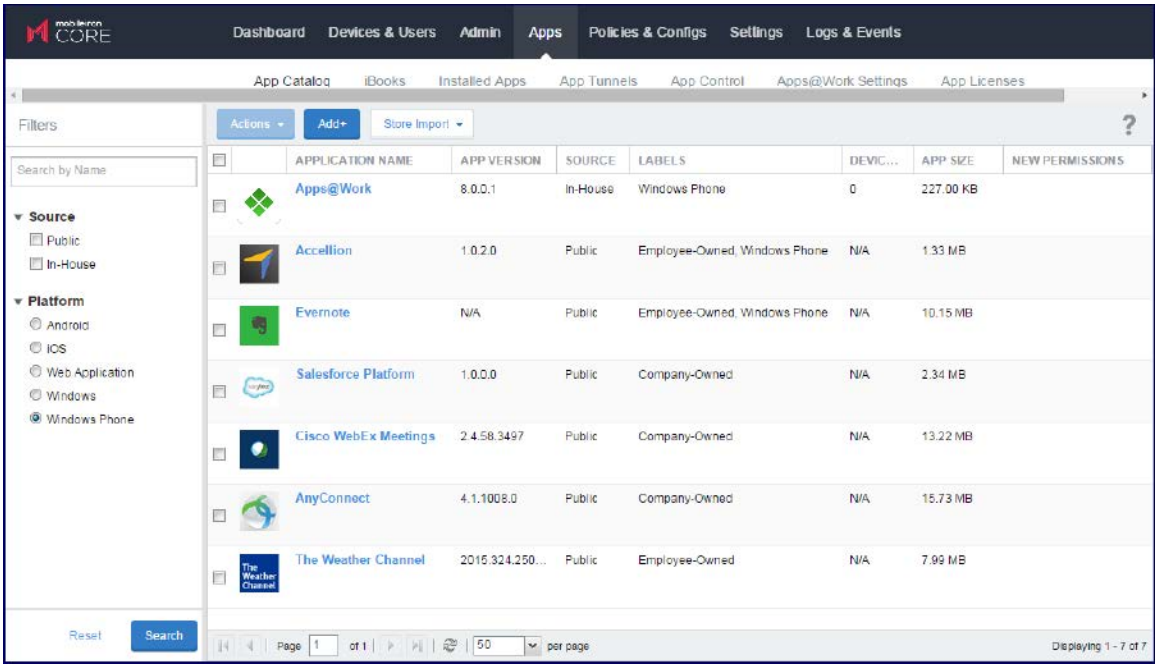
Apps deployed by the Administrator within the last 30 days are displayed in a New Releases section on the Apps@Work home page.

Select an app to view the Detail screen to see the app's ratings, size, developer, install status and more. In this view, click Install to install the app on your device. A Pending install message displays when installation is in progress.

All apps that users download from Apps@Work are considered managed apps.

FIGURE 1. ALL APPS DOWNLOADED FROM APPS@WORK ARE MANAGED APPS

FIGURE 2. APP CATALOG PAGE IN CORE



Use the App Catalog to:

- add, configure, update, and remove managed apps
- edit app configurations

- install and upgrade managed apps to devices using labels
- set the prerequisite app for a dependent app
- indicate mandatory installation of prerequisite apps in Apps@Work
- group apps into categories to be displayed in Apps@Work on the device
- view app details at a glance, such as the:
  - app name, size, and version number
  - label(s) to which the app is applied
  - origins of the app (public or in-house)
  - number of devices, and list of devices, to which the app is deployed
  - new permission status: an icon appears if the app requires new permission

For detailed instructions on working with apps for each platform, see:

- Managing apps for iOS and macOS
- Managing mobile apps for Android
- Managing apps on Windows devices.

Also, see Managing apps on MAM-only devices, if you are working with MAM-only Android or iOS devices, which are devices for which Core does not support device management (MDM).

App Catalog Device Details page

The **Device Details** page for the **App Catalog** tab displays information about devices, but also allows administrators to take actions.

Procedure

1. Log in to Core.
2. Select **Apps > App Catalog**.
3. Select the **Source** and the **Platform**.
4. Locate the app.
5. Use the search box or sort columns to quickly find the app you want.
6. Click the number link in the **Devices Installed** column.

In addition to viewing the device details, you can take the following actions from this page:

- Send a message to a device
- Force a device to check-in
- Indicate if an app must be installed (mandatory)

- Retire a device
- Export to device data (from the table) to an Excel .csv file

NOTE: MobileIron Core does not support viewing device information for apps installed on MAM-only iOS devices.

Exporting App Catalog data

Manage data more easily by exporting app data from the App Catalog to an Excel spreadsheet.

Procedure

1. Log in to Core.
2. Select **Apps > App Catalog**.
3. Select the **Source** and the **Platform**.
4. Locate the app.
   Use the search box or sort columns to quickly find the app you want.
5. Click the number link in the **Devices Installed** column to open the **Device Details** page.
6. Click **Export to CSV** to create an Excel spreadsheet containing the details of the selected app.
7. Locate the .csv file, open, modify, and save, as necessary.
   The exported spreadsheet contains the following information:
   - Device UUID
   - User Name
   - User ID
   - Platform
   - Model
   - Mobile Number
   - Device Space
   - App Version
   - Managed
   - App Name
   - App Identifier

NOTE: MobileIron Core does not support exporting App Catalog data for apps installed on MAM-only iOS devices.

What is App Control?

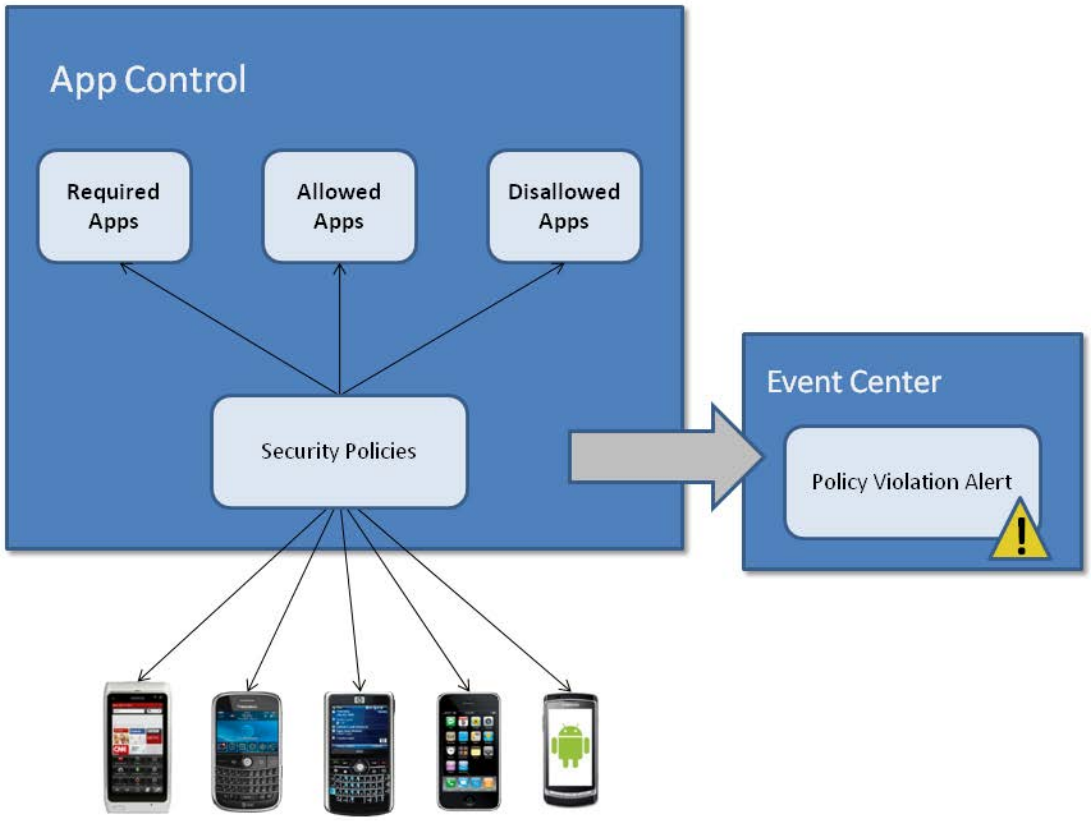
The **App Control** feature enables you to exert control over which apps are installed on managed devices.

Using app control rules, you can define which apps are allowed, disallowed, or required (for iOS, macOS, and Android only). You can then associate these rules with a security policy that specifies the consequences of being out of policy. Note that MobileIron Core does not support app control rules for MAM-only iOS and Android devices.

App control is achieved through a collaboration between the app control rules, Security policy, and alerts:

- The app control rules define which apps you want to control.
- The security policy specifies which devices the rules are applied to and the actions to associate with a rule violation.
- The alert determines the information that is sent as the result of rule violation, and the recipients of the information.

FIGURE 3. APP CONTROL

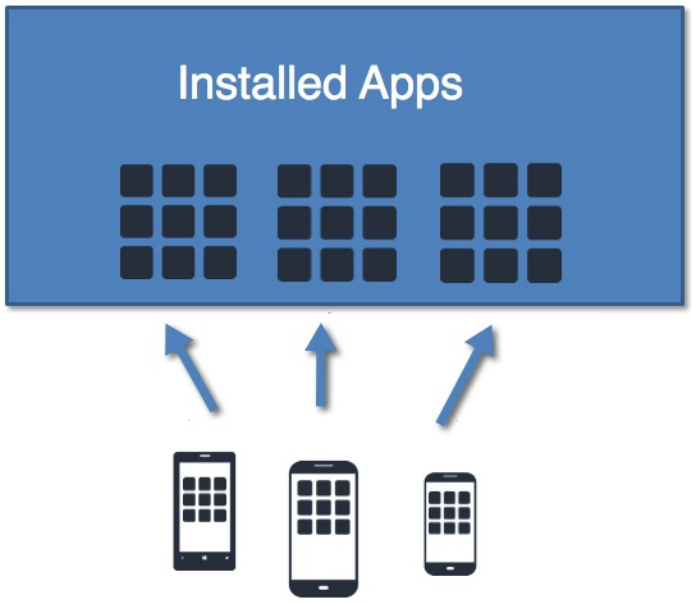


What are Installed Apps?

The **Installed Apps** feature presents a snapshot of the apps installed across your managed devices. The **Apps > Installed Apps** page displays the apps that have been reported as installed on each device. You can use this list to track new apps coming into the enterprise, determine the popularity of apps, and identify possibly rogue apps.

Privacy policy settings determine how devices report their installed apps to Core.

FIGURE 4. INSTALLED APPS



NOTE: MobileIron Core does not support viewing installed apps on MAM-only iOS devices.

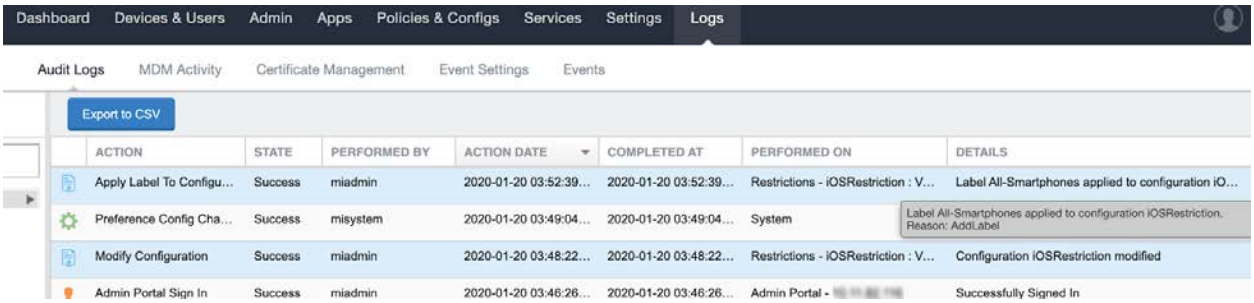
Best Practice: label management

If Notes for Audit Logs is enabled, whenever a change is made to a label, a text box displays for the administrator to provide a reason for the change.

This affects the following label-related activities:

- Add/Edit/Delete/Save Label (Both filter and manual)
- In **Devices & Users > Devices > Advanced Search > Save to Label**
- Add/Edit/Remove Label to devices
- Add/Edit/Remove Label to configurations
- Add/Edit/Remove Label to policies
- Add/Edit/Remove Label to apps
- Add/Edit/Remove Label to iBooks

Example text to enter would be a change ticket order number. This information then displays in the Audit logs, in the Details column as "Reason."



The Notes for Audit Logs feature is also applicable to any administrator-made changes to iOS and macOS restrictions.

To enable this feature, see "Setup tasks" in *Getting Started with MobileIron Core.*

Setting up app control

You can set up app control to enhance visibility into the apps being installed on managed devices and enforce corporate app policy.

App control is achieved through a collaboration between the app control rules, Security policy, and alerts:

- The app control rules define which apps you want to control.
- The security policy specifies which devices the rules are applied to and the actions to associate with a rule violation.
- The alert determines the information that is sent as the result of rule violation, and the recipients of the information.

Setting up app control involves completing the following tasks, in this order:

1. Configure alerts for when a device violates the app control rules in its security policy.
2. Define app control rules.
3. Apply the app control rules to a security policy that is applied to the target devices.

This order of tasks is strongly recommended to ensure that alerts are generated if devices are already in violation when they receive the corresponding policy from MobileIron Core. Otherwise, these devices will not generate an alert until one of the following actions occurs:

- administrator changes the security policy
- administrator edits the app control rule
- device updates app inventory
- device updates device details.

NOTE: MobileIron Core does not support app control rules for MAM-only iOS and Android devices.

About app control alerts

To create an alert, you configure a **Policy Violation Event** in **Logs > Event Settings.**

The security policy specifies whether violating devices should just trigger an alert or also be blocked from ActiveSync access and AppConnect apps. However, if the associated **Policy Violation Event** is not yet defined, no alert is generated.

> IMPORTANT: To ensure that the alert is generated in a timely fashion for devices that are already in violation when the policy is created, you should create the event first.

App control rule types

By creating app control rules, you define lists of apps that are Required, Allowed, or Disallowed on designated devices. These types are defined as follows:

TABLE 1. APP CONTROL RULE TYPES

Rule Type	Purpose	When Policy Violation Occurs
Required	(For iOS, macOS, and Android only) Specify apps that must be installed. NOTE: **Required** rules take precedence over **Disallowed** rules in case of a conflict.	The absence of a required app is a policy violation.
Allowed	Specify a small set of apps that are allowed to be installed.	The presence of an app not on the **Allowed** list is a policy violation.
Disallowed	Specify a set of apps that are forbidden.	The presence of a disallowed app is a policy violation.

You may want to use the rules as described in these examples:

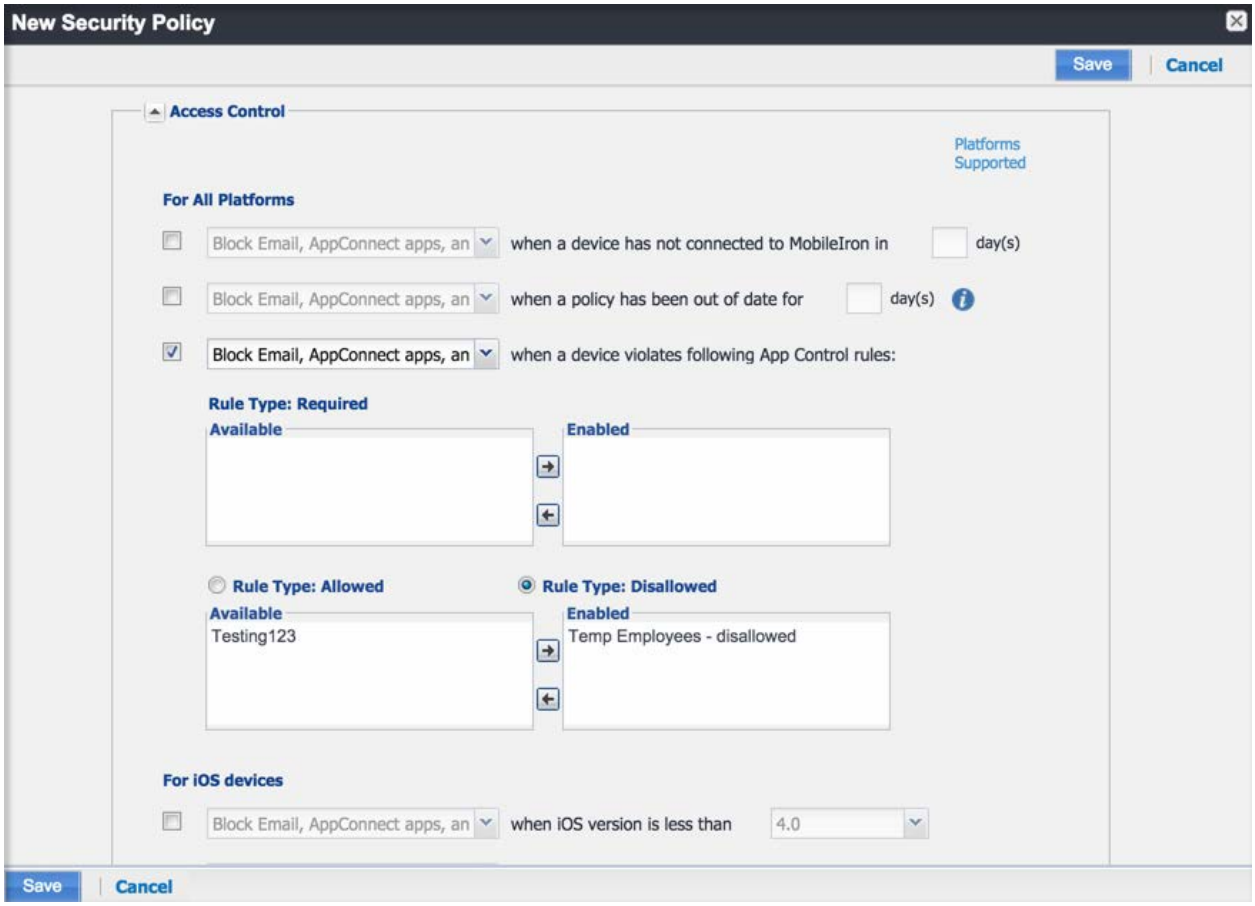
- **Required rules** (iOS, macOS, and Android only) example: since MDM-enabled iOS devices report inventory even if the Mobile@Work has been uninstalled, you can create a **Required** rule to ensure that if the user removes Mobile@Work, the appropriate response is triggered.
- **Allowed rules** example: create a set of **Allowed** rules for use by temporary employees to ensure that they are not installing any personal apps on a corporate device.
- **Disallowed rules** example: create a set of **Disallowed** rules to help lower exposure to apps with known security issues. Note that **Required** rules take precedence over **Disallowed** rules in the case of a conflict.

App control rules applied in security policies

The following figure shows app control rules applied in the **Access Control** section of a security policy. In this case, the selected compliance actions are applied if the disallowed apps are detected on a device to which the

security policy is applied.

FIGURE 5. ACCESS CONTROL SECTION IN A NEW SECURITY POLICY



Setting up App Control

Complete the App Control set up using the following steps in the following order:

1. Configure App Control alerts
2. Define App Control rules
3. Apply the App Control rule to a security policy

Each part of the setup is detailed next.

Step 1: Configure App Control alerts

To enable app control alerts:

1. In the Admin Portal, go to **Logs > Event Settings**.
2. Select **Add New > Policy Violations Event**.

3. Enter a name for the event.
4. In the **Security Policy Triggers** section, look for the **App Control - All Platforms** heading.
5. Confirm that the app control alerts you want to generate have been selected. The following table summarizes these alerts:

Item	Description
Disallowed app found	Generate an alert if a disallowed app is found on a designated device.
App found that is not in Allowed Apps list	Generate an alert if an app is found that is not on the Allowed Apps list for the designated device.
Required app not found	Generate an alert if a required app is not found on a designated device.

6. Disable any other triggers that you do not want to enable.
7. Click **Save**.

Step 2: Define App Control rules

To add an app control rule:

1. In the Admin Portal, go to **Apps > App Control**.
2. Click **Add**.
3. Enter a name for this rule.

NOTE: The name cannot be changed once the app control rule is saved.

4. For the **Type** option, select the type of rule you want to define:
   - Required: (iOS, macOS, and Android only) This rule specifies criteria for apps that MUST be installed. WP8.1 devices ignore this option.
   - Allowed: This rule specifies criteria for apps that MAY be installed, exclusive of all other apps.
   - Disallowed: This rule specifies criteria for apps that MUST NOT be installed.
5. Under **Rule Entries**, provide one or more entries to identify the apps you want to control. Fill out each entry using the guideline that follow:
6. For **App**, select one of the values listed below to indicate if you are providing a partial or exact match with the app name or identifier, or if you're providing a MS Store GUID.

NOTE: if you selected **Required**, then you must select **Identifier Equals** or **Name Equals. Required** is not supported for Windows.

TABLE 2. APP CONTROL RULES

Operator Value	Use for:	App Identifier/ Name field must have:
Identifier Contains	iOS, macOS, and Android	At least a partial match with the app identifier
Identifier Equals	iOS, macOS, and Android	An exact match with the app identifier.
MS Store GUID Equals	Windows Phone 8.1 and Windows 10 Desktop	An exact match with the application's MS Store GUID
Publisher/PFN Equals	Windows 10 Desktop	Dynamic lookup of the Publisher Product Family Name (PFN) from the **Windows Store Search** window. App Control for Windows 10 Desktop devices
EXE/Win32 Equals	Windows 10 Desktop	See Identifying the GUID for a Windows Phone app for details.
Name Contains	iOS, macOS, and Android	At least a partial match with the app name
Name Equals	iOS, macOS, and Android	An exact match with the app name.

7. In the **App Identifier / Name** field, you can use the application name, unique application identifier, or MS Store GUID as follows:
   - **App name**: For iOS, macOS, or Android, type in the official app name you want to match. Do not enter wildcards. If you don't know the official name, enter text that you will be able to identify with this app. Once a managed device has installed the app once, the **Installed Apps**page will display the app's official name. You can then change this field to match.
   - **App identifier**: For iOS and macOS you can enter the app's unique bundle ID, or for Android its package name. Using the unique app identifier instead of the app name helps to ensure that a security policy doesn't unexpectedly block access to important apps when or if an app developer changes the name of an app.
8. For WP8.1 enter the MS Store GUID of the app. (See also: Identifying the GUID for a Windows Phone app.)
9. In the **Device Platform** list, select the platform to which you want to apply this entry.
10. In the optional **Comment** field, you can enter a note about the purpose of the entry.
11. To add another rule entry, click the **+** icon.
12. Click **Save**.

    NOTE: When editing the App Control Rules dialog box, upon clicking **Save**, you will be asked to confirm your changes.

13. This app control rule is now defined.
14. To put this app control rule into use, select it in the **Access Control** section of the appropriate **Security Policy** dialog, as described next.

Step 3: Apply the app control rule to a security policy

To apply an app control rule to a security policy:

1. In the Admin Portal, go to **Policies & Configs > Policies**.
2. Select the security policy you want to work with.
3. In the Policy Details pane, click **Edit**. The Modify Security Policy dialog box opens.
4. Scroll down to the **Access Control** section.
5. Under **For All Platforms**, select the check box **when a device violates following App Control rules:**". The field activates.
6. In the drop-down list, select the action you want to perform if the app control rule is violated. You can select from:
   - **Block Email, AppConnect apps, and Send Alert**: This option prevents the device from accessing email via ActiveSync and generates a policy violation alert, if configured. This option also unauthorizes AppConnect apps, and blocks app tunnels.
   - **Send Alert**: This option generates a policy violation alert if you have configured the alert in **Logs > Event Settings** page.
   - Any custom compliance actions you have created, which will appear in this list.
7. Under **Rule Type: Required**, select the rules you want to apply, if any, and click the arrow button to move them from the **Available** list to the **Enabled** list.

NOTE: The list of items that appear in the **Available** column are the App Control Rules you defined in the previous setup step.

8. To apply allowed-type or disallowed-type rules, select either **Rule Type: Allowed** or **Rule Type: Disallowed**. You may not select both in the same security policy.
9. Select the allowed-type or disallowed-type rules you want to apply and click the arrow button to move them from the **Available** list to the **Enabled** list.
10. Click **Save**.
11. Apply the security policy to a label that is also applied to the target devices. Click **Actions > Apply to Label**.

The app control rules are now defined and applied to the devices through the security policy.

Editing app control rules

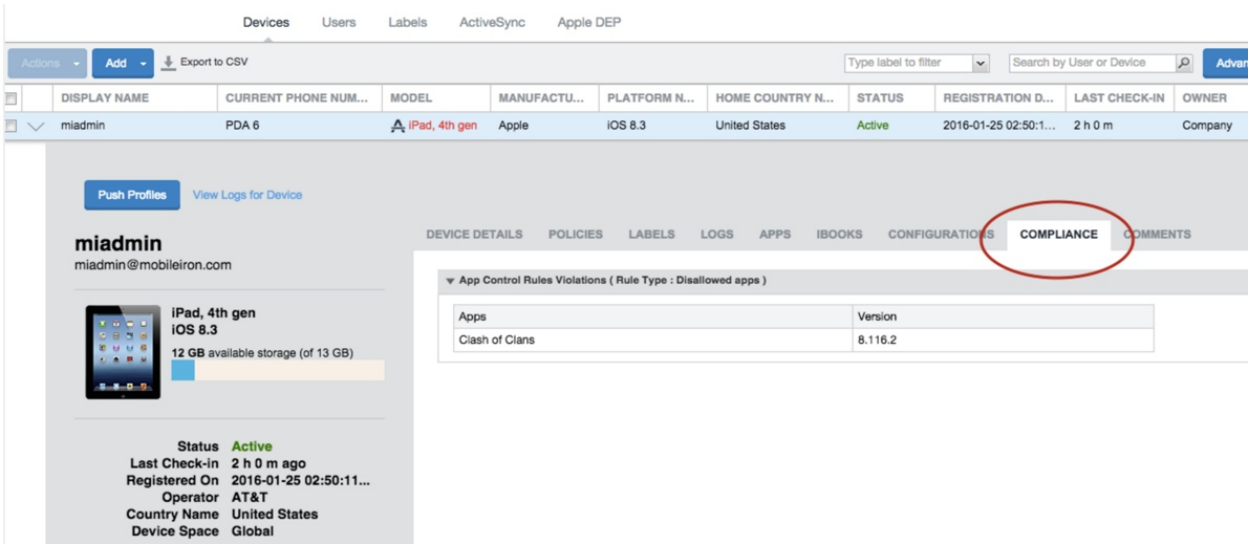
To edit an app control rule, click the edit icon next to the rule in the **Apps > App Control** page. Note that you cannot change the type of an app control rule if that rule has been applied to a security policy. To delete it, remove it from the security policy first.

Viewing app control status

In addition to the alerts you can configure, Core displays app control status for devices in the **Devices & Users > Devices** page. Select the entry for a device in violation to see details in the **Device Details** pane.

Click the caret next to the device entry to open the device details pane. Click **Compliance** to see the app control status information.

FIGURE 6. VIEWING APP CONTROL COMPLIANCE STATUS



The following table shows the icons that indicate app control violations:

Icon	Description
	App control violation
	Required app violation
	Allowed app violation
	Disallowed app violation

App Control for Windows 10 Desktop devices

NOTE: This feature is for Windows 10 Desktop only.

AppLocker allows administrators to block specific apps from being downloaded or executed. You can block apps by using one of the following two approaches:

- Excluding apps (blacklist) - specifying apps to block.
- Including apps (whitelist) - Identifying allowed apps and excluding all other apps not on the list and all systems not defined by the administrator.

Use the dynamic lookup feature to include Publisher/PFN (Product Family Name) from the Microsoft store to include or exclude apps to security policies.

Creating a rule to block apps from Windows 10 desktop devices

This procedure describes:

- using dynamic lookup to create a rule (called *Blacklist*) that excludes specified apps.
- applying the *Blacklist* rule to a security policy.

To create an app control rule excluding specified apps using dynamic lookup:

1. In the Admin Portal, go to **Apps > App Control > Add**. The Add App Control Rule dialog box opens.
2. Enter *Blacklist* in the **Name** field as the name of the rule.
3. Select **Disallowed** for the **Type** option.
4. Select **Publisher/PFN Equals** from the **App** drop-down.
   Leave the **App Identifier/Name** blank.
5. Select **Windows** from the **Device Platform** drop-down.

NOTE: The Windows icon appears next to the **Comment** field when you select **Windows** as the platform.

6. Click the Windows icon to open the **Windows Store Search** dialog box.
7. Click the **Windows** option.
8. Locate the app and click the **Select** button to automatically insert the PFN into the **App Identifier/Name** field.
9. (Optional) Click the green plus (+) sign to add more apps to the rule, as necessary.
10. Click **Save**.

    NOTE: When editing the App Control Rules dialog box, upon clicking **Save**, you will be asked to confirm your changes.

Applying a rule to block apps from Windows 10 desktop devices

When you block an app that is in use and installed from the Microsoft Store, the app will continue to run until users close it. When users open a blocked app, Windows displays a message informing users that the app has been blocked by their system administrator. MobileIron sends instructions to the OS to block the specified app(s).

When users try to install a blocked app, they will see a message that the app has been blocked due to company policy.

Procedure

1. In the Admin Portal, go to **Policies & Configs > Policies**.
2. Select **Default Security Policy** and click **Edit**.

3. Scroll to the **For Windows Devices** section in the **Access Control** group.
4. Click the box next to **Application Restrictions** and select *Blacklist* from the drop-down.
5. Click **Save**.

Identifying the GUID for a Windows Phone app

The GUID is a unique number that identifies the app in the Microsoft ecosystem. In the Windows Phone Store, select the app. The URL for the app includes the GUID. The GUID is the alpha numerical section at the end of the URL.

Example:

```
http://www.windowsphone.com/en-us/store/app/netflix/c3a509cd-61d6-df11-a844-00237de2db9e
```

In the example, the GUID is c3a509cd-61d6-df11-a844-00237de2db9e.

Viewing installed apps

The **Apps > Installed Apps** page (previously called Device App Inventory) displays the apps that MobileIron Core has detected on managed devices. Only the apps that were installed on devices after the manufacturer's image was loaded are listed. The privacy policy assigned to a device can determine whether or not the device reports its installed apps to Core.

This section includes the following sub-sections:

- What's in an app name?
- Synchronizing app inventory
- Determining which apps devices will report
- Filtering the App Catalog inventory view
- Displaying the devices on which an app is installed
- Installed Apps Device Details page
- Export Installed Apps data

NOTE: MobileIron Core does not support viewing installed apps for MAM-only iOS devices.

What's in an app name?

The app names displayed on the **Installed Apps** page are the names reported by the apps installed on managed devices, not the name you assigned when you added an app to the **App Catalog**. Therefore, if you are looking for an app you know is installed, but you cannot find it, make sure you are looking for the correct name. Note that any control characters found in the reported app name are converted to spaces in Core, and app names are stored without regard to case.

Synchronizing app inventory

App inventory data is updated based on the **Sync Interval** specified in the Sync policy. Therefore, inventory changes on the device are not reflected in real time on the **Installed Apps** page. During testing, you can use one of the following methods to decrease the amount of time it will take to update the inventory:

- decrease the **Sync Interval** in the Sync policy
- use the **Force Device Check-in** feature in the Admin Portal (for supported platforms). Go to **Devices & Users > Devices**; select the device and click **Actions > Force Device Check-in**.
- use the **Connect Now/Check for Updates/Refresh** feature in the MobileIron client (for supported platforms)
- check for updated configurations (for iOS)

Determining which apps devices will report

The Privacy policy assigned to a device determines whether the device reports its installed apps. If the **Apps** option in the privacy policy is set to **None**, then installed apps data for the device do not appear in the **App Catalog**.

Also note that changing the setting **Apps** to **None** in the Sync policy drops the current inventory data. Setting **Apps** back to **Sync Inventory** re-enables inventory reporting for iOS (with timing governed by the **Sync Interval** specified in the sync policy). For all other platforms, you must make a change in the app distribution or reboot the device in order to restart the inventory process.

App filters for iOS installed apps

The **App Filters** feature in the Privacy policy allows you to control which iOS apps are reported on the **Installed Apps** page. Select a choice in the **iOS Installed App Inventory** drop-down to set the device to report only iOS managed apps or a list of apps that the administrator specifies. All other apps on the user's device are not reported to Core, providing additional privacy to the device user.

Filtering the App Catalog inventory view

In the Apps Catalog, you can filter the inventory display by:

- Platform
- Label
- App name

For example, to display iOS apps that are on company-owned devices and contain the letter "A", you would select **iOS** from the **Platform** list, select **Company-Owned** from the **Labels** list, and enter **A** in the **Search by Name** field. Clicking the **Search** button begins the search.

Click **Reset** to clear the search results.

Displaying the devices on which an app is installed

Each app entry on the **Installed Apps** page includes the number of devices on which the app has been installed in the **Devices Installed** column. The displayed number is a link. Click the link to display a list of the devices on which the app is installed.

Installed Apps Device Details page

The **Device Details** section of the **Installed Apps** page displays information about devices, but also allows administrators to take actions.

Procedure

1. In the Admin Portal, select **Apps > Installed Apps**.
2. Select the **Source** and the **Platform**.
3. Locate the app
4. Use the search box or sort columns to quickly find the app you want.
5. Click the number link in the **Devices Installed** column.

In addition to viewing the device details, you can take the following actions from this page:

- Send a message to a device
- Force a device to check-in
- Retire a device
- Export to device data (from the table) to an Excel .csv file

Export Installed Apps data

You can manage data easier by exporting app data installed on devices to an Excel spreadsheet.

Procedure

1. In the Admin Portal, select **Apps > App Catalog**.
2. Select the **Source** and the **Platform**.
3. Locate the app.
4. Use the search box or sort columns to quickly find the app you want.
5. Click the number link in the **Devices Installed** column to open the **Device Details** page.
6. Click **Export to CSV** to create an Excel spreadsheet containing the details of the selected app.
7. Locate the .csv file, open, modify, and save, as necessary.
   The exported spreadsheet contains the following information:
   - Device UUID
   - User Name

- User ID
- Platform
- Model
- Mobile Number
- Device Space
- App Version
- Managed
- App Name
- App Identifier

Managing app inventory

You can use the **Apps > Installed Apps** page to help manage the apps that are appearing in your enterprise.

- The **Summary View** shows one entry per unique app identifier, and displays fewer columns for simplicity.
- The **Detail View** shows one entry for every version of every app that is installed on users' devices. You can also view detailed information about each app by clicking on the app's link.

The detail view also shows the app rating and app score for each app if you have set up an app rating service in **Settings > System Settings > Additional Products > App Reputation**.

This section includes the following sub-sections:

NOTE: These views do not display apps on MAM-only iOS devices.

Determining when an app was first reported

The date an app was first reported by a managed device can be an important piece of information when investigating possible issues with the app. MobileIron Core tracks this information for each app displayed in the **Installed Apps** page in Summary View, in the **First Found** column. Click the column header to sort the rows by this field.

Displaying permissions for Android apps

Android's unique approach to app permissions can pose a challenge to administrators, as each app may have dozens of permissions associated with it. To provide easier access to this information, Core displays the permissions granted to each Android app on the **Installed Apps** page in Detail View, in the **Permissions** column.

Click the number in the **Permissions** column to display the permissions.

What happens when an app is removed?

Once an app is removed from all managed devices, the entry for that app no longer appears in the **Installed Apps** page. If you want to be able to track which apps you have determined to be “bad”, consider adding the information in the **Comment** field for an app control rule.

App management action workflows

This section addresses dependent and prerequisite apps and how to manage them.

- A "dependent" app is an app / in-house app that has dependencies on one or more app in order to function correctly. In application associations, a dependent app can have only one level of pre-requisite app support. This means a prerequisite app cannot be a dependent app for another app.
- A "prerequisite" app is an app that is required to be installed so that the dependent app can fully function.

Overview of the device users' workflow

For Apps@Work users, when a device user taps a (dependent) app to install, the user is informed that prerequisite app(s) are required to be downloaded first. Once the prerequisite app(s) are downloaded, the user can then download the main / dependent app. The user is prompted by tapping Install or Install All prerequisite apps, and then prompted to install the dependent app.

This applies to managed apps, unmanaged apps and in-house apps. For Android devices, the user will need to manually install each of the prerequisite apps before installing the dependent app.

All apps that users download from Apps@Work are considered managed apps. If a prerequisite app, is subsequently removed from a label, in Apps@Work the device user will see a "Not Available" text after the prerequisite app name listed.

Overview of the administrator's workflow

1. As a convenience to the end user, in MobileIron Core, the administrator can associate a prerequisite app to a dependent app. See To associate a prerequisite app to a dependent app: on page 36.
2. The administrator then assigns the dependent app to a label (the prerequisite apps are automatically assigned to the same label.) See To assign a label to a dependent / prerequisite app: on page 37.

As long as the administrator has the **Enforce this version for Mandatory Apps.** check box selected, the device user will download the latest version of that app.

Manage prerequisite app

The Manage Prerequisite App action item is used by administrators and is applicable for iOS, macOS, Android, and Android enterprise platforms. A prerequisite app could be set to mandatory for the specific label. This ensures that if a user inadvertently removes the app, upon the device's next check in, the app will be pushed to the device.

To associate a prerequisite app to a dependent app:

Once a prerequisite app has been associated, it cannot be defined as a dependent app and vice versa. Unless the association is removed, apps deployed by the Administrator within the last 30 days are displayed in a New Releases section on the Apps@Work home page.

Procedure

1. Log in to Core.
2. Select **Apps > App Catalog**.
3. Select the **Source** and the **Platform**.
4. Locate and select the dependent app; only one app of the same platform can be selected.
5. Select **Actions > Manage Prerequisite App**.
6. In the Manage Prerequisite App dialog box, use the search box to quickly find the prerequisite app you want to assign to the dependent app. You can select one or more prerequisite apps. Whatever label that is associated to the dependent app will be applied to the prerequisite app, for example, iOS.
7. Select the app and then select **Apply**.
8. The dependent app now has the prerequisite app associated to it and vice versa. In the **App Catalog** , this information displays in the App Dependencies column. Hovering over the item in the **App Dependencies** column displays the application name, source and version number the prerequisite / dependent app is associated to.

The audit logs captures the following information: admin name, date, action, app names, and app dependencies created.

NOTE: To make the prerequisite apps a mandatory installation in Apps@Work, see Managing installed iOS and macOS apps. To send installation requests to users of Apps@Work, see Notifying users of new iOS and macOS apps or app updates.

To remove the association of a dependent / prerequisite app:

You can remove the association of prerequisite apps to dependent apps.

Procedure

1. Log in to Core.
2. Select **Apps > App Catalog**.
3. Select the **Source** and the **Platform**.

4. Locate and select the dependent app.
5. Select **Actions > Manage Prerequisite App**.
6. In the Manage Prerequisite App dialog box, use the search box to quickly find the prerequisite app you want to remove.
7. Clear the prerequisite app check box and then select **Apply**.
8. In the **App Catalog** > **App Dependencies** column, both the prerequisite app and its associated dependent app are not displayed.

The audit logs capture the following information: admin name, date, action, app names, and app dependencies deleted.

To assign a label to a dependent / prerequisite app:

After associating the prerequisite app to a dependent app, you need to apply a label to the dependent app. Once a dependent app is assigned to a label, prerequisite apps are automatically associated to the same label.

If a prerequisite app is removed from a label, in Apps@Work, device users will see a "Not Available" text after the listed prerequisite app name.

When a master Apple license app is assigned to non-Apple license labels, the prerequisite apps are auto-assigned to those labels. However, if a primary Apple license app is assigned to Apple license labels, auto-assigning to Apple license labels will not occur. Administrators will need to manually apply Apple license prerequisite apps to Apple license labels.

Procedure

1. Log in to Core.
2. Select **Apps > App Catalog**.
3. Select the **Source** and the **Platform**.
4. Locate and select the dependent app.
5. Select **Actions > Apply to Label**.
6. In the **Apply to Label** dialog box, select the prerequisite app and then select **Apply**.
7. In the **App Catalog** > **App Dependencies** column, both the prerequisite app and its associated dependent app are displayed.

To remove a label from a dependent / prerequisite app:

Procedure

1. Log in to Core.
2. Select **Apps > App Catalog**.
3. Select the **Source** and the **Platform**.

4. Locate and select the prerequisite app.
5. Select **Actions > Manage Prerequisite App**.
6. In the Manage Prerequisite App dialog box, use the search box to quickly find the prerequisite app you want to associate.
7. Clear the app's check box and then select **Apply**.
8. In the **App Catalog > App Dependencies** column, both the prerequisite app and its associated dependent app are not displayed.

NOTE: If the prerequisite app is removed from the label without removing the association, then device users will see a "Not Available" text after the listed prerequisite app name.

The audit logs capture the following information: admin name, date, action, app names, and app dependencies deleted.

NOTE: To make the prerequisite apps a mandatory installation in Apps@Work, see Managing installed iOS and macOS apps. To send installation requests to users of Apps@Work, see Notifying users of new iOS and macOS apps or app updates.

For detailed instructions on working with apps for each platform, see:

- Managing apps for iOS and macOS
- Managing mobile apps for Android
- Managing apps on Windows devices.

NOTE: MobileIron Core does not support viewing device information for apps installed on MAM-only iOS devices.

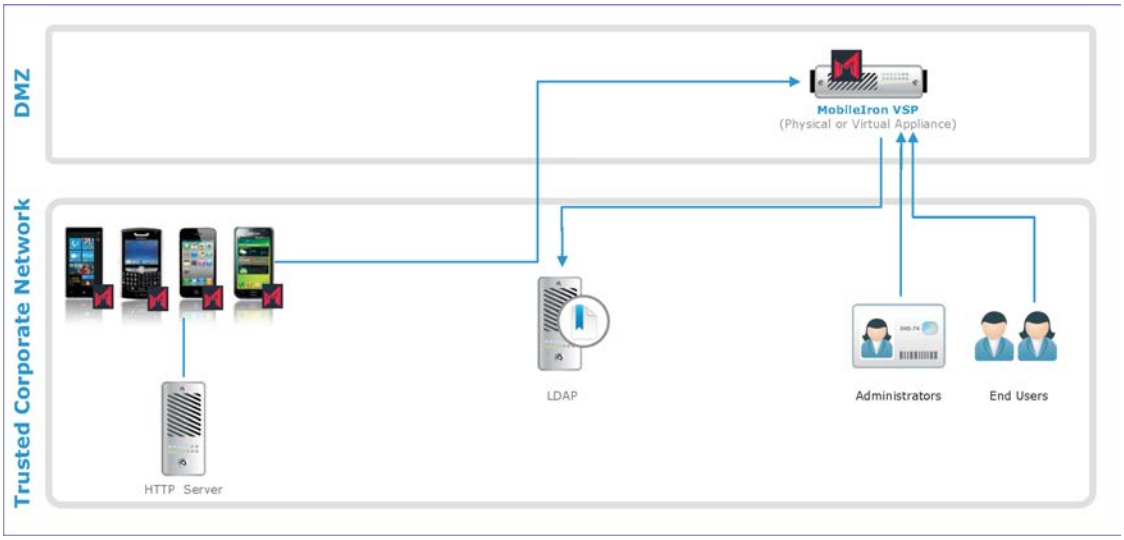
Override for in-house app URLs

MobileIron supports an alternative for off-loading distribution of in-house apps to alternate HTTP servers. This option is intended only for those customers who meet all of the following criteria:

- numerous internally-developed apps for distribution to thousands of devices
- a trusted and secure internal network
- available HTTP servers
- concerns about performance impact on MobileIron Core
- ability to manually synchronize apps between Core and an alternate location

This alternative enables you to specify an override URL, per app, to be used for in-house app distribution. MobileIron Core routes download requests to this alternate location. The following diagram illustrates a typical deployment.

FIGURE 7. OVERRIDE FOR IN-HOUSE APP URLS



This feature uses unauthenticated URLs. Therefore, this feature is intended for use behind the firewall, using a trusted and secure internal network. The URL should use the HTTPS, not HTTP, URL scheme. However, the feature allows you to use the HTTP URL scheme. Before you use an HTTP URL, make sure you understand the risks of using an insecure connection.

This section includes the following sub-sections:

Implementing app source override on MobileIron Core

If you have the supporting infrastructure in place, complete the following steps to implement app source override:

1. In Admin Portal, go to **Apps** > **App Catalog**.
2. Select the appropriate OS from the **Platforms** list.
3. As you complete the forms in app wizard, include an appropriate URL in the **Override URL** field.
   The URL must point to the in-house app in its alternate location.
   If you are using the HTTP URL scheme, select **Allow app downloads over insecure networks**. Make sure you understand the risks of using insecure networks.
4. Finish adding the app and assign an appropriate label to the app.

Manual synchronization of apps for override URLs

MobileIron Core does not synchronize the apps configured in Apps@Work with those stored on the HTTP server in this override URL configuration. The administrator must perform this maintenance manually and develop a process for ensuring proper synchronization.

Malware prevention: app reputation

Integration with Appthority provides app reputation data for apps detected on managed devices. This information helps you protect your organization from malware.

This section includes the following sub-sections:

Enabling app reputation

Before using an app reputation service:

- Find out whether or not the service supports the MobileIron APIs and can be used with MobileIron Core
- Get a URL for their service
- Determine the service's rating range (for example, 0 to 50)
- Determine what the low and high numbers in the service's rating range indicate (do low numbers indicate a high or low threat?)

Procedure

1. Consider configuring debug mode for MIFS logs (in System Manager).
   Debug logs will capture successful configuration. Otherwise, you will have no indication if you mistype the license key for the reputation service.
2. Go to **Settings > Additional Products**.
3. Click **App Reputation**.
4. Select the **Enable App Reputation** option.
5. Use the following guidelines to complete the displayed fields:

Item	Description
Reputation Service URL	Enter the URL your app reputation service provided.
Authentication Type	Select **Basic** or **Token Authentication**.
Name/Password	Sepcify a username and password when you select **Basic Authentication**.
Authentication Key	Provide an authentication key when you select **Token Authentication**.
Rating Range Low Value	Enter the low number of the service's range.

Item	Description
Rating Range High Value	Enter the high number of the service's range.
Rating Scale	Click **Low** to indicate that apps with ratings lower than the Rating Threshold have the highest threat level (for example, if the range is 0 to 100, and the Rating Threshold is 60, apps with a rating of 60 or below have a high threat rating) Click **High** to indicate that apps with ratings higher than the Rating Threshold have the highest threat level (for example, if the range is 0 to 100, and the Rating Threshold is 65, apps with a rating of 65 or more have a high threat rating)
Rating Threshold	Specify the rating you select as the limit for determining whether an app has a high or low threat rating. It is used in combination with Rating Scale to determine the app threat risk.
Check Interval	Select an interval for contacting the reputation service to retrieve updated reputation data: • **Daily**: Update occurs at midnight each day. • **Weekly**: Update occurs at midnight between Saturday and Sunday. • **Monthly**: Update occurs at midnight before the first of the month. The reputation data is stored on MobileIron Core. NOTE: The day of the week and time of the update are not configurable.

6. Click **Save**.

An initial sync begins shortly after initial configuration. Thereafter, the **Check Interval** setting determines when Core contacts the reputation service.

Confirming configuration of the app reputation service

You can use the following keywords to check the logs for successful configuration of the reputation service:

```
appReputationEnabled=true

Enabling Appthority-Sync-Job with schedule: 0 30 22 * * ?

appReputationServiceOption=Appthority

appRatingThreshold

appReputationIntervalOption

Rescheduling Appthority-Sync-Job with schedule

AppthoritySyncJob.execute
```

```
Done with sync job

scores.length
```

Viewing app reputation data

The **Apps** > **Installed Apps** page displays the information about apps detected on managed devices. Select **Detail View** to see the app rating and app score columns. Those columns appear if you have enabled app reputation in **Settings > Additional Products > App Reputation**.

The values that may appear in the **App Rating** field are listed in the table below.

TABLE 3. APP REPUTATION RATINGS

Rating	Description
Not Rated	With a score of 0 indicates that MobileIron Core has not processed the app yet. With a blank score indicates that the app is not currently in the designated service's database. The app might be new or the service might provide app data only for specific operating systems.
OK	Indicates that the app's score exceeds the threshold specified in the **App Reputation** settings.
Risky	Indicates that the app's score does not exceed the threshold specified in the **App Reputation** settings.

Apps@Work branding

You can brand Apps@Work on iOS, macOS, and Android devices with your own enterprise app store branding. To brand Apps@Work, you specify the branding assets in the Admin Portal in **Apps > Apps@Work Settings**. The assets that you specify are:

- App Icon
- App Name
- Text color

On Android devices, Apps@Work is part of Mobile@Work. Branding requires Mobile@Work 9.5.0 for Android through the most recently released version as supported by MobileIron.

On iOS and macOS devices, Apps@Work is either:

- a web clip provided by MobileIron Core
- The assets that you specify are applied to the web clip.

- the Apps@Work container app, a MobileIron app which you rebrand and sign (iOS MDM devices only).
- For the Apps@Work container app, only the app name and app color that you specify in the Admin Portal are applied. The app icon you specify in the Admin Portal is not used. Instead, the app uses the app icons you provide in the Apps@Work container app package.

This section includes the following sub-sections:

Related topics

The tech note *Apps@Work Container for iOS*

Apps@Work custom branding assets

The following table describes the Apps@Work branding assets that you can provide and how they are used:

TABLE 4. APPS@WORK BRANDING ASSETS USAGE

Asset	Android use	iOS use
App Icon	• As the home page icon • In the splash screen	• As the home page icon • In the splash screen NOTE: If you use the Apps@Work container app for iOS, rather than the provided web clip, this icon is not used. See the tech note *Apps@Work Container for iOS*.
App Name	• Below the home page icon • Below the splash screen icon • In the main menu of Mobile@Work • In the navigation bar at the top of the display	• Below the home page icon • Below the splash screen icon
Text color	• splash screen • navigation bar	• splash screen • various button text, such as install,

TABLE 4. APPS@WORK BRANDING ASSETS USAGE (CONT.)

Asset	Android use	iOS use
	• **Home**, **Categories**, and **Updates** tabs (lighter shade of selected color) • buttons • selectable text • Search and review text field borders • carousel pagination dots	download, and update • top and bottom borders (lighter shade of selected color)

Apps@Work custom icon requirements

The following table gives the requirements for the custom-branding of Apps@Work app icons.

TABLE 5. REQUIREMENTS FOR APP ICONS

Icon	Dimensions	Format	iOS resolution scale factor	Android densities
App Icon	1024 x 1024	PNG	@1x, @2x, or @3x	mdpi, hdpi, xhdpi, xxhdpi, or xxxhdpi

Apps@Work custom app color requirements

For Apps@Work custom branding, you can provide the app color. You can either click on a color box to select a color, or enter the color directly as a # symbol following by either three or six of the following characters:

- 0 through 9
- A through F

Six characters specify a hex color code. A hex color code contains three pairs of hexadecimal numbers, in which each pair represents the intensity of red, green, or blue (RGB). The characters 00 represent the lowest intensity of a color, and the characters FF represent the highest intensity.

For example:

- #FF0000 is red
- #00FFFF is aqua
- #FF00FF is fuchsia

Three-character color codes are shorthand for six-character codes. For example, #84D is the shorthand for both #8042D1 and #8040D0, although the six-character codes represent different shades.

IMPORTANT: Try your app color code on iOS and Android devices. Make sure you do not choose a color that is hard to see, or easily confused with typical color usage, such as gray for disabled buttons.

Relationship of Apps@Work branding with iOS or macOS web clip configuration

On the Admin Portal, in **Policies & Configs > Configurations**, MobileIron Core provides a default web clip configuration for Apps@Work named **System - iOS Enterprise AppStore**. Because you provide the custom app name and app icon in **Apps > Apps@Work Settings**, you cannot edit the **Name** or **Icon** fields in the web clip configuration. You also cannot edit the **Address/URL** field.

Core upgrade impact to Apps@Work branding

The following table shows how the more limited Apps@Work branding support in MobileIron Core versions prior to 9.5.0.0 are impacted after upgrading to this version of MobileIron Core:

TABLE 6. CORE UPGRADE IMPACT TO BRANDING

Feature prior to Core 9.5.0.0	Impact after upgrade
For iOS devices, you could upload a banner icon.	The icon displays as an App Icon in the Admin Portal in **Apps > Apps@Work Settings** page.
You could modify the Apps@Work name and app icon in the web clip configuration for Apps@Work, which is named **System - iOS Enterprise AppStore**	Your modifications to the name and icon display in the web clip configuration, but cannot be modified. Enter modifications in **Apps > Apps@Work Settings.**

NOTE: After upgrade, iOS devices continue to use existing custom branding settings, if any, until you save custom branding settings in **Apps > Apps@Work Settings**. Any new customization will result in updating the splash screen to white, however, the App Icon and App Text will be preserved.

Configuring Apps@Work branding

You can brand Apps@Work on macOS, iOS, and Android devices with your own enterprise app store branding. The changes made affects the client Home screen, Splash screen, and App Home Screen.

Procedure

1. In the Admin Portal, go to **Apps > Apps@Work Settings**.
2. In the **App Storefront Branding** section, select **Custom Branding**.
3. In the Customize App Storefront > App Icon section, click **Replace Icon**.
4. Navigate to and select your custom image for the app icon and then click **Upload**.
5. In the **App Name** section, enter your custom app name.

   NOTE: The app catalog name you enter applies to Android, iOS, and macOS.
6. In the **Text Color** section, click on the color box to select a color, or enter the three or six character color code for your custom app color.
7. Click **Save**.

Related topics

- Apps@Work custom icon requirements
- Apps@Work custom app color requirements

Android OS limitations to updating the Apps@Work home page icon

When you change the Apps@Work app icon or app name, whether Mobile@Work for Android can automatically update the icon and name on the home page depends on the version of Android running on the device. The following table summarizes for which Android versions the home page update is automatic.

TABLE 7. IS HOME PAGE ICON AND NAME AUTOMATICALLY UPDATED?

	Prior to Android 5.0	Android 5.0 - 6.0	Android 7.0 through the most recently released version as supported by MobileIron
non-Samsung devices	Yes, update is automatic.	No	No
Samsung devices	Yes, update is automatic.	Yes, update is automatic.	No

NOTE: Updates to the app name, app color, and app icon *inside* the Apps@Work app are automatic on all versions of the Android OS.

When the update is not automatic, the device user can manually update the new home page icon and name by doing the following steps.

Procedure

1. Manually remove the existing home page icon (shortcut) for Apps@Work.
2. Launch Mobile@Work.
3. Tap the menu icon.
4. Tap **Settings > Check for Updates.**

Managing app reviews in Apps@Work (Android, iOS, macOS)

NOTE: This feature is available for Apps@Work on Android, iOS, and macOS devices only. You can manage app reviews in the global device space only.

As long as an app is available for installation from Apps@Work, device users can review the app after they have installed it. You can manage app reviews so that only the most current reviews for the latest app version are visible

to device users, for example. For any given app in the App Catalog, you can delete individual reviews, or all reviews.

Managing app reviews involves the following main steps:

- Enabling app review management
- Deleting app reviews for a managed app (Android, iOS, macOS)

Enabling app review management

You enable the administrative management of app reviews by adding the Managing reviews option to the App Management admin role.

NOTE: This feature is available for Apps@Work on Android, iOS, and macOS devices only. You can manage app reviews in the global device space only.

Procedure

1. In the Admin Portal, select **Admin > Admins**.
2. Select the check box next to the name of the administrators for whom you wish to enable app review management.
3. Select **Actions > Edit Roles**.
4. In the **Edit Roles** window, select **App Management > Manage reviews**.
5. Click **Save**.

Related topics

Deleting app reviews for a managed app (Android, iOS, macOS)

Deleting app reviews for a managed app (Android, iOS, macOS)

You can delete selected reviews of a managed app available on Apps@Work. For example, you can delete one individual review, several reviews, or all reviews of a managed app.

NOTE: This feature is available for Apps@Work on Android, iOS, and macOS devices only. You can manage app reviews in the global device space only.

Before you begin

Enable app review management for the administrator users in the global space, as described in Enabling app review management.

Procedure

1. Go to **Apps > App Catalog**.
2. Select an app whose reviews you want to delete.
3. Select **Actions > Manage Reviews**.
   A list of reviews for the app is shown in the **Manage Reviews** window. You can optionally sort the reviews by date.
4. Select the review you want to delete.
5. Alternatively, select all the reviews for this app.
6. Click **Delete**.

Related topics

- Enabling app review management

Enabling device users to submit app ideas through Apps@Work

You can enable device users to submit app ideas through Apps@Work by enabling a feedback icon in Apps@Work. Tapping the feedback icon in Apps@Work opens a web page where users can enter their thoughts and ideas about the apps they install through Apps@Work, or new apps they would like to have available through Apps@Work.

When you enable this feature, you add the URL of the web page you want to display to users on tapping the feedback icon.

Procedure

1. Select **Apps > Apps@Work Settings**.
2. Go to the **App Storefront Feedback** section.
3. Select **Enable feedback**.
4. In the text field that displays, enter the full URL of the page device users will use to provide feedback. The URL must include the protocol, such as http:// or https://, for example:
   `https://www.example.com/appfeedback`.
5. Click **Save**.

Setting the default landing page for Apps@Work

When opening Apps@Work, Apps@Work defaults to showing the last page visited. Instead, you can configure Apps@Work to display the home page by default upon launching.

Procedure

1. Select **Apps > Apps@Work Settings**.
2. Go to the **Apps@Work Launch Setting** section.
3. Select **Default to home screen**.
4. Click **Save**.

Configuring popular apps for display in Apps@Work (Android, iOS, macOS)

The Popular Apps section in Apps@Work shows up to 25 App Catalog apps with the greatest number of installations in descending order over the last 30, 60, or 90 days, or all time. Device users only see those popular apps applied to labels to which they belong, regardless of whether they have installed these apps.

Popular apps are updated in Apps@Work every 60 minutes. Popular apps not available for download to a given device will not be shown. Uninstalled apps are not counted or shown.

Procedure

1. Select **Apps > Apps@Work Settings**.
2. Go to the **App Storefront Popular Apps** section.
3. Select **Enable Popular Apps**.
4. From the Duration drop-down list, select one of the following options:
   - All Time
   - 30 days
   - 60 days
   - 90 days
5. Click **Save**.

Managing app categories (Android, iOS, macOS)

The Categories page allows you to create and manage app categories that are displayed in Apps@Work. You can organize App Catalog apps into categories that are displayed in Apps@Work. For example, you can assign all sales-related apps to the Sales app category, making it easy for salespeople to find the apps they need to use in Apps@Work.

For each app category, Core shows the number of apps assigned to that category and the order in which the categories are displayed in Apps@Work.

Managing app categories involves the following tasks:

- Adding an app category for Apps@Work (Android, iOS, macOS)
- Editing or deleting an app category for Apps@Work (Android, iOS, macOS)
- Changing the display order of app categories in Apps@Work (Android, iOS, macOS)

Before you begin

Make sure you have the correct permissions for managing app categories.

Procedure

1. Select **Admin** > **Admins**, then select the admin user whose permissions you want to change.
2. Select **Actions** > **Edit Roles**.
3. Select **App Management** > **Import and edit app**.
4. Click **Save**.

Adding an app category for Apps@Work (Android, iOS, macOS)

You can create categories to help organize apps for display in Apps@Work. For each category, you can add a name, description, and image.

Procedure

1. Select **Apps** > **Categories**.
2. Click **Add+**.
3. Configure the following:
   - **Name**: Enter a meaningful name for the app category.
   - **Description**: Enter a meaningful description for the app category.
   - **Category Icon**: Click **Replace Icon** to choose an image for the app category (JPEG, GIF, or PNG files only).
4. Click **Save**.

Related topics

- Managing app categories (Android, iOS, macOS)
- Editing or deleting an app category for Apps@Work (Android, iOS, macOS)
- Changing the display order of app categories in Apps@Work (Android, iOS, macOS)

Editing or deleting an app category for Apps@Work (Android, iOS, macOS)

You can edit or delete an existing app category from Apps@Work.

NOTE: You can only delete an app category if there are no apps assigned to that category.

Procedure

1. Select **Apps** > **Categories**.
2. Select the category you want to edit or delete.
3. To edit a category, select **Actions** > **Edit**. Make the changes you desire and click **Save**.
4. To delete a category, select **Actions** > **Delete**.

Related topics

- Managing app categories (Android, iOS, macOS)
- Adding an app category for Apps@Work (Android, iOS, macOS)
- Changing the display order of app categories in Apps@Work (Android, iOS, macOS)

Changing the display order of app categories in Apps@Work (Android, iOS, macOS)

Given at least two app categories, you can change the order in which app categories are displayed in Apps@Work.

Procedure

- Select **Apps** > **Categories**.
- Drag and drop a given category in the order you want it to appear.

Related topics

- Managing app categories (Android, iOS, macOS)
- Adding an app category for Apps@Work (Android, iOS, macOS)
- Editing or deleting an app category for Apps@Work (Android, iOS, macOS)

Managing apps for iOS and macOS

This section addresses the management of apps for iOS and macOS devices.

Overview of working with apps for iOS devices

If MobileIron Core has Apps@Work configured, then Core installs an Apps@Work web clip on the user's device after registration is complete. The user will see the default Apps@Work web clip icon, or your custom icon if you have customized the app store.

The device user taps this icon to access Apps@Work. Apps@Work shows lists of apps that you have configured for download from the Apple App Store or MobileIron Core. These are called managed apps, as they are managed by MobileIron Core.

The apps appear in these tabbed sections:

- Featured: The featured page lists all apps that the administrator designates as featured. These apps can include in-house, recommended, web, and prepaid apps.
- Categories: An app can be featured and listed under multiple categories. Uncategorized apps are displayed under the **Uncategorized** category. Only categories that have at least one app will appear on the user's device.
- Updates: The updates page shows all in-house apps that have an available update. The **Update All** button allows the device user to update all apps at the same time.

NOTE: Device users must use an iTunes account to download apps from the Apple App Store.

iOS managed apps

When iOS apps are managed apps on a device for which MobileIron Core is the Mobile Device Management (MDM) server, the apps are called *iOS managed apps*. As the Core administrator, you can control whether an iOS managed app is backed up and whether the app is deleted when the MDM profile is removed or the device is quarantined. Existing apps installed on a device can be converted to iOS managed apps on devices running iOS 9 through the most recently released version as supported by MobileIron. Device users running iOS 8.4 and earlier must delete existing unmanaged apps on their devices and reinstall them as iOS managed apps.

You can also:

- restrict document interaction between iOS managed apps and unmanaged apps.
  See "Restriction settings" in the MobileIron Core Device Management Guide.
- provide app-specific configurations to iOS managed apps.
  See iOS managed app configuration.

Also, per Apple guidelines, MobileIron Core periodically checks the validity of iOS managed apps on iOS devices running iOS 9.2.1 through the most recently released version as supported by MobileIron.

NOTE: iOS managed apps are not supported on MAM-only iOS devices.

Prerequisites for iOS managed apps

Complete app functionality, including updates to badges resulting from inventory data, requires:

- iOS MDM certificate (See "Enabling iOS MDM support" in the On-Premise Installation Guide for MobileIron Core and Enterprise Connector)
- iOS MDM profile enabled (Settings > System Settings > iOS > MDM)

If you intend to develop and manage in-house apps, an enterprise-level Apple Developer account is required. For more information, see the Apple Developer site: https://developer.apple.com/.

AppConnect apps

You upload iOS AppConnect apps created with the AppConnect wrapping technology to the App Catalog as in-house apps. AppConnect apps created with the SDK can be distributed as either in-house apps or recommended

public apps from the Apple App Store. The process for adding an AppConnect app to the App Catalog is the same as for any iOS app.

When you upload an iOS AppConnect app as an in-house app to the App Catalog, in some cases MobileIron Core automatically creates an AppConnect container policy and AppConnect app configuration. Core takes this action when the app has specified its desired default values for the policy and configuration in its IPA file. You can override these values by editing the app's AppConnect container policy or AppConnect app configuration. Core keeps in sync the labels that you apply to the app and the labels that you apply to the AppConnect container policy and AppConnect app configuration.

For information about AppConnect apps, see the *AppConnect and AppTunnel Guide*.

Apps@Work container app for iOS that displays badges for app updates

An unsigned Apps@Work container app is available for iOS. You can download, re-brand, and sign this app if you want device users to see badges for app updates. The total number of updates available is shown in a badge that diesplays on the Apps@Work icon.

This number includes updates, new installations, and unmanaged to managed app conversions for iOS managed apps, featured apps, and in-house apps. Individual apps with new installations available will display their own badges.

Apps@Work will only be badged if it is being pushed as a container app.The package is available as a separate file in the Apps@Work Container App article in the Customer Support knowledge base in community.mobileiron.com. You will need to click through a separate license agreement before being able to download the file.

See the *Apps@Work Container for iOS* tech note for information on implementing and distributing this app.

NOTE: The AppConnect container app is not supported on MAM-only iOS devices.

Authentication options and iOS versions

The authentication options supported and the resulting user experience depend on the iOS version being used:

- Certificate-based app authentication
  - app downloads proceed without routing end-users to the app page in iTunes (assuming an iTunes account has been properly configured on the device)

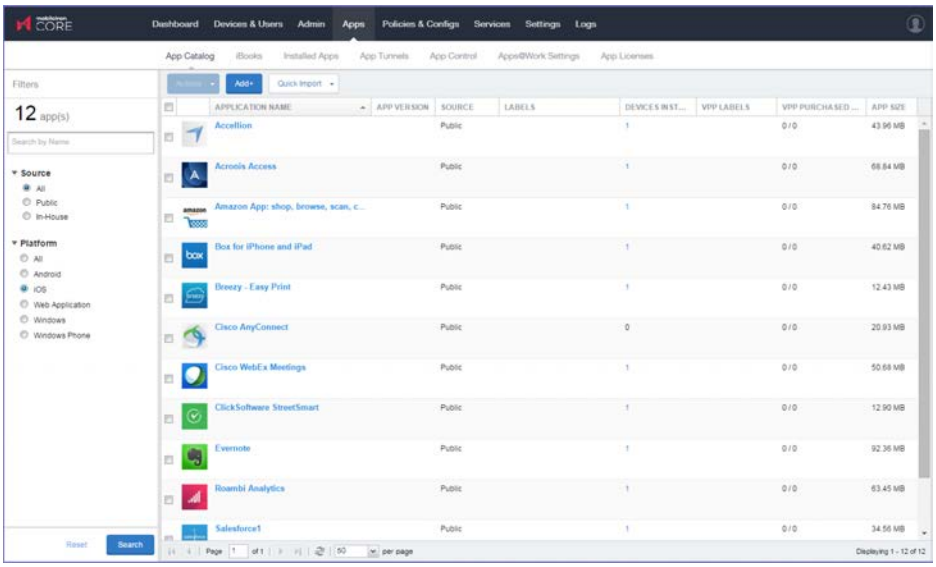
- HTTP basic authentication
  - app downloads proceed without routing end-users to the app page in iTunes (assuming an iTunes account has been properly configured on the device)
  - requires end-users to enter their MobileIron username and password to download apps

The App Catalog

The **App Catalog** is a centralized location for the apps you want to manage for your users. By importing apps to the App Catalog, you can make the apps available for users to download to their devices.

You can provide device users with links to recommended iOS apps on the Apple App Store, or links to internally developed apps they can download from MobileIron Core using Apps@Work on their device.

FIGURE 8. APP CATALOG



You use the App Catalog to:

- add, configure, and remove managed apps
- install and uninstall managed apps to devices using labels
- group apps into categories to be displayed in Apps@Work on the device
- set the prerequisite app for a dependent app
- indicate mandatory installation of prerequisite apps in Apps@Work
- use Apple licenses

The App Catalog also allows you to view app details at a glance, such as the app name, size, the version number of in-house apps, the labels to which the app is applied, the origins of the app (public or in-house), and the number of devices to which the app is installed.

Note The Following:

- Core shows the version number of an app if the app developer assigned a version number to the app.
- Some App Catalog features are not available for MAM-only iOS devices, as described in MAM-only iOS devices.

The iBooks screen for iOS

The iBooks feature allows you to distribute iBooks, Kindle books (ePub), and PDF files to iOS devices managed by MobileIron Core. You can also edit and delete managed books, and search for particular managed books.

For more information about managing books on iOS devices, see the "Managed iBooks on iOS devices" section in the MobileIron Core Device Management Guide for iOS devices.

NOTE: iBooks are not supported on MAM-only iOS devices.

iOS managed app configuration

An iOS managed app can automatically get its app-specific configuration from MobileIron Core, rather than requiring the device user to enter the values in the app. Some examples of app-specific configuration are:

- user information
- server information
- whether particular features should be enabled

This feature results in easier app deployment and fewer support calls for you, and a better user experience for the device user.

MobileIron Core supports iOS managed app configuration with two different mechanisms:

- The Managed App Config setting that use plists
- Managed App Configuration settings for iOS apps in the App Catalog

IMPORTANT: Both mechanisms use native iOS capabilities. iOS stores the configuration settings **unencrypted** on the device. Therefore, do not provide sensitive information such as passwords or private keys in managed app configuration values.

NOTE: iOS managed app configuration is not supported on MAM-only iOS devices.

The Managed App Config setting that use plists

The Managed App Config setting is one mechanism that MobileIron Core can use to provide configuration settings to iOS managed apps. You create a Managed App Config setting in **Policies & Configs > Configurations > Add New > iOS and macOS > Managed App Config**.

Using a Managed App Config setting requires a MobileIron license. For more information on this feature, see "Managed App Config settings that use plists" in the *MobileIron Core Device Management Guide for iOS and macOS Devices*.

NOTE: By default, a legacy Managed App Config setting is ignored if a Managed App Configuration setting is available for the app in its App Catalog entry.

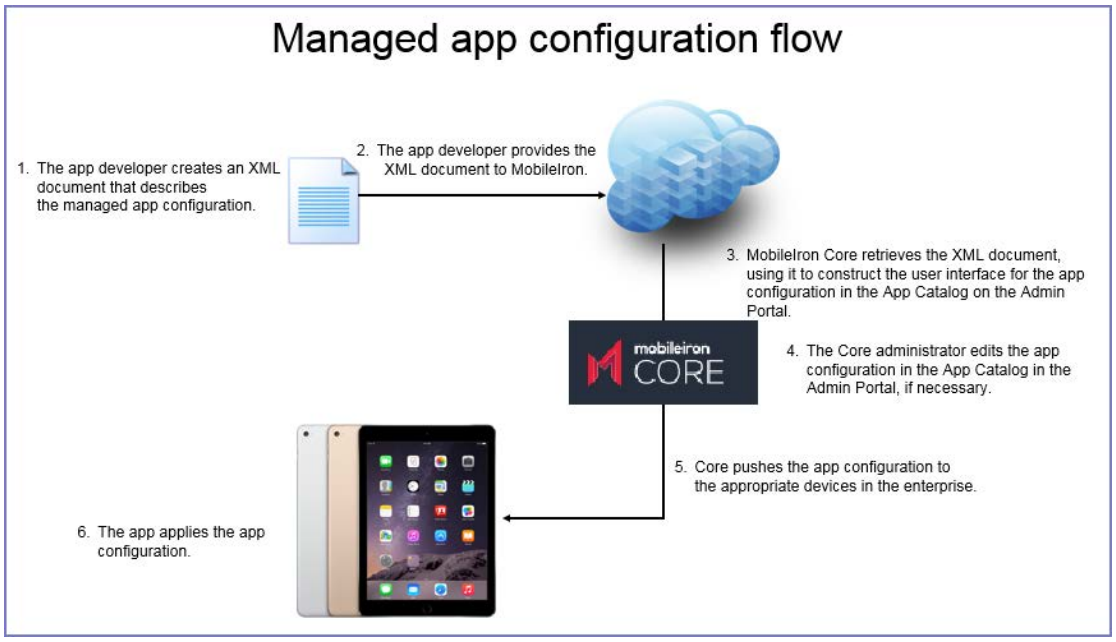
Related topics

Configuring the plist setting to take precedence over the iOS managed app configuration setting

Managed App Configuration settings for iOS apps in the App Catalog

This mechanism supports the iOS managed app configuration defined in the AppConfig Community at appconfig.org. Working with MobileIron, many registered MobileIron Technology Partners who are deploying their apps to the Apple App Store support this mechanism to make their apps easier to deploy in enterprises. This mechanism works as follows:

FIGURE 9. MANAGED APP CONFIGURATION FLOW



Using this mechanism makes it easy for you to configure an iOS managed app's configuration on MobileIron Core. Specifically:

- When you import the app into the App Catalog, Core automatically retrieves the default app configuration for viewing and editing.
- You edit the values for the app configuration in the Admin Portal in a graphical user interface.
- Depending on the app, the user interface includes descriptions about each field.

- You can create multiple app configurations, applying different labels to each app configuration. Multiple app configurations allow different sets of devices to receive different configuration values.

Refer to the app's documentation to find out:

- whether the app supports managed app configuration
- more details on its specific configuration settings.

NOTE: MobileIron Core supports this mechanism only for Apple App Store apps, not for in-house apps.

This topic includes the following sections:

Multiple app configurations per iOS app

Core allows you to create multiple app configurations per app:

- The default app configuration for the app is applied to devices with the same label that you applied to the app.
- Any additional app configurations that you create are applied to devices with the same labels that you specify for the additional app configuration.

Using multiple app configurations is useful when sets of users of the app require different configuration values. For example, consider a Human Resources app that users throughout the United States use. However, you want the app to connect to a different server depending on a user's region:

- Users in the Eastern region must connect to a server in the east.
- Users in the Western region must connect to a server in the west.
- Users in the Northern and Southern regions connect to a server in St. Louis.

Therefore, do the following:

- Label the app with the Human Resources label.
- Create an app configuration that specifies the server in the east, and label the app configuration with the Eastern Region label.

- Create an app configuration that specifies the server in the west, and label the app configuration with the Western Region label.
- In the default configuration, specify the server in St. Louis. Users who do not have the Eastern Region label or the Western Region label will use this server.

Priorities of iOS app configurations

Each app configuration you create has a priority. The highest priority has the value 1 and appears at the top of the list of app configurations. The default configuration always has the lowest priority and appears at the bottom of the list. Core assigns a device the app configuration with the highest priority that has a label that matches a label on the device.

You can change the priorities of app configurations by dragging and dropping them in the table of configuration choices for the app.

Substitution variables for configuring iOS apps

Substitution variables can be used for configuring values from LDAP or the MobileIron Core devices database, such as $EMAIL$ for the email address. You can prevent deleted default field values from repopulating when editing app configurations by entering the substitution variable $NULL$ for those values.

You may use the following variables when configuring app configuration fields:

TABLE 8. SUBSTITUION VARIABLES FOR CONFIGURING IOS APPS

Substitution variable	More information	Sample of substituted value
$USERID$	Login ID (email address format)	jdoe@myCompany.com
$EMAIL$	Email address	jdoe@myCompany.com
$EMAIL_DOMAIN$	The domain part of the email address (part after the '@')	myCompany.com
$EMAIL_LOCAL$	The local part of the email address (part before the '@')	jdoe
$PASSWORD$	Use not recommended because the managed app configuration values are not encrypted on the device	
$FIRST_NAME$	First name	Jane
$LAST_NAME$	Last name	Doe
$DISPLAY_NAME$	Display name	Jane Doe, CEO

TABLE 8. SUBSTITUION VARIABLES FOR CONFIGURING IOS APPS (CONT.)

Substitution variable	More information	Sample of substituted value
$USER_DN$	Distinguished Name	CN=Jane Doe, OU=NA,OU=Users, OU=XY, DC=myCompany, DC=com
$USER_UPN$	The Microsoft userPrincipalName attribute	jdoe@myCompany.com
$USER_LOCALE$	Locale	en_US
$DEVICE_UUID$	iOS Unique Device Identifier	c752e7052fe5e5ca8166e408c4b48573b5b5bd82
$DEVICE_UUID_NO_DASHES$		
$DEVICE_IMSI$	International Mobile Subscriber Identity	310150123456789
$DEVICE_IMEI$	International Mobile Equipment Identity	01 342300 291808 3
$DEVICE_SN$	Serial Number	DNRJVLP7DTTN
$DEVICE_ID$	Mobile Equipment Identifier	A0123456789012
$DEVICE_MAC$	Wi-Fi MAC Address	30:f7:c5:87:e8:78
$DEVICE_CLIENT_ID$	Unique device identifier	1073741831
$MODEL$	Device model	iPhone 6
$PHONE_NUMBER$	Device phone number	888-555-1212
$USER_CUSTOM1$	Custom field defined for LDAP	The value of the variable as defined in LDAP settings.
$USER_CUSTOM2$	Custom field defined for LDAP	The value of the variable as defined in LDAP settings.
$USER_CUSTOM3$	Custom field defined for LDAP	The value of the variable as defined in LDAP settings.
$USER_CUSTOM4$	Custom field defined for LDAP	The value of the variable as defined in LDAP settings.
CN	Common Name (CN) attribute extracted from the distinguished name	Jane Doe

TABLE 8. SUBSTITUION VARIABLES FOR CONFIGURING IOS APPS (CONT.)

Substitution variable	More information	Sample of substituted value
OU	Organizational Unit (OU) attribute extracted from the distinquished name	XY
$ICCID$	Integrated Circuit Card Identifier	89014104254287052057
$SAM_ACCOUNT_NAME$	The Microsoft sAMAccountName attribute	jdoe
$MI_APPSTORE_URL$	The URL of the MobileIron Core app store, as accessed by the Apps@Work web clip	https://myCore.mycompany.com/mifs/asfV3/appstore?clientid=$DEVICE_CLIENT_ID$&vspver=9.3.0.0
$REALM$	The domain component of an LDAP entry	mycompany.com
$TIMESTAMP_MS$	Unix time stamp of when Core sends the managed app configuration to the device	1485992717498
$NULL$	An empty string. Use this variable to prevent the re-population of deleted default values.	<no value>

Changes to managed app configurations for iOS apps

For iOS apps, when the app data is in View or Edit mode, Core loads the latest managed app schema from the AppConfig repository and displays the latest fields (including any new fields) in the “Managed App Configurations” section in the UI. MobileIron recommends that before saving the changes, you first carefully inspect the updated managed app configuration. Once you select **Proceed** and click **Confirm**, the updated managed app configuration settings are saved and the changes are pushed out to all associated devices.

When you change the values for the app configuration of an app in the App Catalog, either one or two device check-ins are necessary for the device to receive the new values from Core. If the iOS MDM terminates the connection between the device and Core before Core can deliver the update, a second device check-in may be necessary.

App version updates and managed app configuration for iOS apps

When you update an app in the App Catalog on Core to a newer version, the new version sometimes has an updated managed app configuration. However, Core does not push the updated managed app configuration until you edit and save the app in the App Catalog. Until that time, devices that upgrade to the new version of the app

still receive the older version of the app configuration. Because a new version of an app is typically backward compatible with the older app configuration, the app will still run successfully. However, the app will not use any new features that the updated app configuration provides.

Configuring the plist setting to take precedence over the iOS managed app configuration setting

Consider the case in which both of the following are true:

- Core has retrieved the managed app configuration for an app.
- A Managed App Config setting with a plist exists for the app.

By default, the managed app configuration included with the app overrides the Managed App Config setting with a plist. However, you can specify that the Managed App Config setting with a plist should override the managed app configuration with the following procedure.

Before you begin

Make sure you have created a Managed App Config setting with a plist and assigned the necessary labels to it. See "Managed App Config settings that use plists" in the *MobileIron Core Device Management Guide for iOS and macOS Devices.*

Procedure

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Select the app.
3. Click **Edit**.
4. In the **Managed App Configurations** section, select **Use the .plist file uploaded in a Managed App Config Setting instead of these Managed App Configurations**.
5. Click **Save**.

NOTE: If no Managed App Config setting is applied to the device, the app still uses the default managed app configuration in the App Catalog entry.

Adding a new managed app setting for an app

In addition to the default managed app configuration, you can add managed app settings from the AppConfig community or by uploading an XML file. The settings in the new managed app configuration can be edited in the Admin Portal. You add new managed app settings for an app by editing the app in the Admin Portal.

Procedure

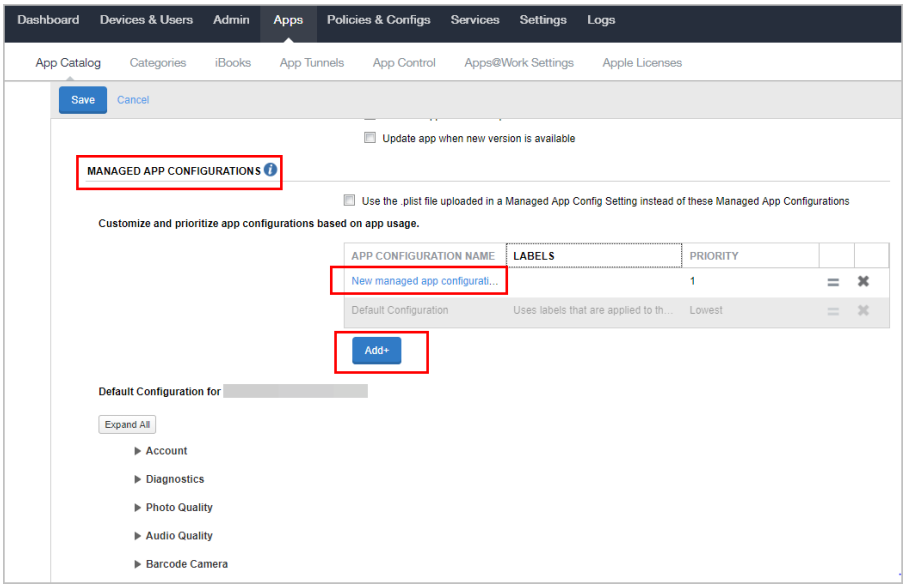
1. In the Admin Portal, go to **Apps > App Catalog**.
2. Select the app.

3. Click **Edit**.
4. In the **Managed App Configurations** section, for **Customize and prioritize app configurations based on app usage**, click **Add**.
5. Enter a name for the managed app configuration.
6. For **Source Type**, select one of the following:
   - **AppConfig Community**: This option is available only if the app has an app configuration available in the AppConfig community repository. If the configuration is available, the option is selected by default.
   - **Upload .xml spec**: Select the option to upload an XML schema to push a particular set of app configurations.
7. If your source type is **Upload .xml spec**, do one of the following:
   - Drag and drop the .xml file into the dotted box.
   - Click **Choose File** to navigate to the location and upload the .xml file.

   NOTE: Ensure that the .xml file contains the version and bundle ID for the app, and that the bundle ID in the .xml file matches the bundle ID for the app. An error message displays if the bundle ID in the file does not match with the bundle ID of the app.
8. Scroll down and select a label to apply the configuration.
9. Click **Add**.

   The new managed app configuration displays in the **Managed App Configurations** section.

FIGURE 10. ADD MANAGED APP CONFIGURATION



10. Update the configuration fields as needed.

    The configuration fields are populated with the values available in the .xml file. If the XML file does not contain any default values, an empty configuration will get pushed to devices. Therefore, check the configuration values and update as needed.

11. Click **Save**.

Core upgrade and iOS managed app configuration

Consider the case where:

- you upgraded to this version of MobileIron Core from a version of Core that did not support managed app configuration, and
- an app was already in the App Catalog before the upgrade.

After the upgrade, Core does not immediately retrieve the app's managed app configuration. Core retrieves it when you edit the app in the App Catalog.

Setting up Apps@Work for iOS and macOS

iOS device users cannot use Apps@Work by default. You must first set up Apps@Work for iOS by completing the following tasks:

1. Set authentication options.
   See Setting authentication options for Apps@Work for iOS devices on page 64.
2. Optionally, customize the icon for Apps@Work.
   See Apps@Work branding on page 42
3. Optionally, enable users to rate Apps@Work apps.
   See Enabling device users to rate and review apps in Apps@Work
4. Send the Apps@Work web clip to iOS devices.
5. See Sending the Apps@Work web clip to iOS and macOS devices.
   If you do not complete this step, then iOS devices will not have access to Apps@Work.

Note The Following:

- Because the Apps@Work web clip is deployed like any other configuration, there might be a considerable lag between device registration and the appearance of the web clip.
- As a web clip, Apps@Work is impacted by web content filters, available in supervised devices. Make sure your web content filters do not block access to MobileIron Core. If Core access is blocked, Apps@Work cannot work. For more information, see "Web content filter settings" in the *MobileIron Core Device Management Guide.*

Setting authentication options for Apps@Work for iOS devices

By default, both certificate-based authentication and HTTP basic authentication are enabled.

Note The Following:

- If neither authentication option is selected, then iOS devices will not have access to Apps@Work.
- When both options are selected, certificate authentication is used. If it fails, HTTP basic authentication is used.
- If only certificate authentication is selected, certificate authentication is used. If it fails, iOS devices will not have access to Apps@Work.
- This setting applies to both the Apps@Work iOS web clip and the Apps@Work container app for iOS.
- The option to use certificate authentication also impacts Android devices, as described in Apps@Work in Mobile@Work for Android.

Before you begin

Change the **Apps@Work Port** setting in the System Manager if all of the following are true:

- You are using certificate-based authentication for Apps@Work for iOS.
- You have enabled mutual authentication for devices at **Settings > System Settings > Security > Certificate Authentication**.
- You are using the iOS Apps@Work web clip.

To change the **Apps@Work Port** setting, see "Port Settings" in the *MobileIron Core System Manager Guide*.

Procedure

1. In the Admin Portal, go to **Apps > Apps@Work Settings**.
2. Under iOS App Storefront Authentication, select one or both of the following:
   - **Certificate Authentication**
   - **HTTP Basic Authentication (iOS only)**
3. Clear the authentication options you do not intend to support.
4. Click **Save**.

Related topics

- “Enabling mutual authentication for Apple and Android devices” in *MobileIron Core Device Management Guide for iOS and macOS Devices*
- "Port Settings" in the *MobileIron Core System Manager Guide*.
- Apps@Work in Mobile@Work for Android

Enabling device users to rate and review apps in Apps@Work

You can optionally allow users to rate and review the apps you push to Apps@Work.

Procedure

1. In the Admin Portal, go to **Apps > Apps@Work Settings**.
2. Under **App Storefront Reviews and Ratings**, select **Enable Ratings and Reviews for iOS, macOS and Android**.
3. Click **Save**.

Related topics

Malware prevention: app reputation

Sending the Apps@Work web clip to iOS and macOS devices

MobileIron Core sends the Apps@Work web clip to iOS and macOS devices only after you assign the iOS and macOS labels to the web clip, respectively.

NOTE: On macOS devices, the Apps@Work web clip is only supported on the Safari web browser.

Procedure

1. In the Admin Portal, go to **Policies & Configs > Configurations**.
2. Select the WEBCLIP configuration type called **System - iOS Enterprise AppStore**.
3. Go to **Actions > Apply to Label**.
4. Select the **iOS** or the **macOS** label, or both.
5. Click **Apply**.
   Core automatically applies the iOS and macOS labels to the **System - iOS Enterprise AppStore SCEP** setting, enabling Apps@Work to authenticate with Core.

Populating the iOS and macOS App Catalogs

You can search for iOS apps on the Apple App Store and add them to the App Catalog. You can also add your own in-house apps for iOS and macOS.

You can add apps to the App Catalog as follows:

- Search for and import App Store apps for iOS.
  Manually importing iOS apps from the Apple App Store
- Use the app wizard to add apps from the iOS and macOS App Store.
  Using the wizard to import iOS apps from the Apple App Store
- Add in-house apps for iOS and macOS.
  See Using the wizard to add an in-house iOS or macOS app to the App Catalog and Using the wizard to add an in-house macOS bundled app to the App Catalog.
- Adding new versions of an existing iOS or macOS app

You can also remove iOS and macOS apps from the App Catalog, as described in this section.

Before you begin populating the App Catalog with in-house apps, you may find it useful to understand provisioning profiles, which allow apps to function (see Provisioning profiles for in-house iOS apps).

macOS apps

Currently, Apple does not support managed applications on macOS devices. You can, however, distribute Apple Licenses, macOS apps, in-house macOS apps, and web applications to macOS devices.

Related topics

- App management action workflows
- Working with web applications for iOS and macOS
- Using Apple licenses
- Using the wizard to add an in-house iOS or macOS app to the App Catalog

Provisioning profiles for in-house iOS apps

You can distinguish app-specific provisioning profiles from wildcard provisioning profiles by examining the application identifier key value in the provisioning profile. App-specific provisioning profiles indicate the app in particular, whereas wildcard provisioning profiles have an asterisk, indicating a match with more than one app.

For example, the following application identifier key value indicates the provisioning profile is specific to an app signed by example.com:

```
<key>application-identifier</key>
<string>A1B2C3D4E5.com.example.webcontainer</string>
```

Conversely, the following application identifier key value indicates the provisioning profile is a wildcard profile, matching more than one app:

```
<key>application-identifier</key>
<string>A1B2C3D4E5.*</string>
```

When adding in-house iOS apps to the App Catalog, the UI will indicate if the provisioning profile is expired, and therefore needs replacing.

If you need to update the provisioning profile for an app, keep in mind the following rules:

TABLE 9. PROVISIONING PROFILE TYPES FOR APPS

Provisioning profile type	Action	Result
App-specific	Update expired profile	Profile updated
Wildcard (*)	Update expired profile	New provisioning profile added
App-specific	New profile uploaded to replace expired profile	MobileIron Core removes the label from the app used with the expired provisioning profile (using a daily background job configured in the common.properties file).
Wildcard (*)	Profile matches an app name or UUID	Existing provisioning profile is attached to the app, and labels applied to the profile are also applied to the app.
Wildcard (*)	Profile does not match an app name or UUID	MobileIron Core adds a new provisioning profile to the app, and applies to the app those labels applied to the profile.

For information about adding iOS app provisioning profiles to MobileIron Core using the Admin Portal, see "Provisioning profile settings" in the chapter "Managing Device Settings with Configurations" in the MobileIron Core Device Management Guide for iOS and macOS Devices.

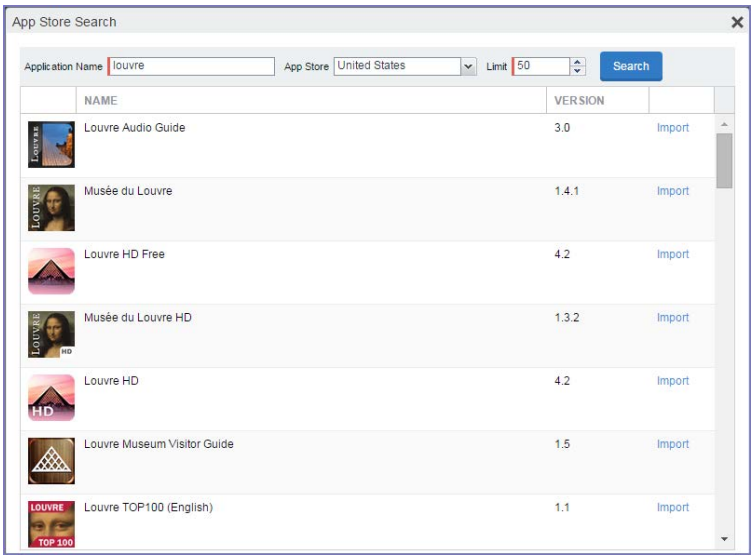
Manually importing iOS apps from the Apple App Store

You can manually import iOS from the Apple App Store directly into the App Catalog on MobileIron Core using **Quick Import**. This import configures the app with default app settings. You can later edit the app settings.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Click the **Quick Import** button.
3. Select **iOS** from the drop-down.
4. In the **Application Name** field, enter search text.
   iTunes matches the text against app names, app IDs, app authors, and app descriptions.
5. From the **App Store** drop-down list, select the country for the App Store you want to search.
6. In the **Limit** field, enter the number of entries you want to retrieve.
   To improve search performance, the default is set to 50. You can enter a number between 0 and 200.
7. Click **Search**.
   The matching apps displays.

FIGURE 11. SEARCH RESULTS IN APP STORE SEARCH



8. Click the **Import** or **Re-import** link for an app to import the relevant information.
   Import indicates an app that does not yet exist in the App Catalog.
   Re-import indicates an app that exists in the App Catalog, which can be re-imported. A newer version may, or may not be available from the Apple App store.

NOTE: You can import or update more than one app from the search results. Alternatively, you can run another search in the same dialog and import additional apps that way.

9. Close the dialog box by clicking **OK**.
   The app displays in the **App Catalog**.
10. Click the application name to view the app details.
11. Click **Edit** to change the app settings.
12. Make any necessary changes to the default settings. The settings are described in detail in Using the wizard to import iOS apps from the Apple App Store.

NOTE: When you import recommended apps from the Apple App Store that use licenses, clear the **This App Store App is Free** check box. This allows the device user to successfully download the app user licenses.

13. To apply a category, see Creating or changing a category for iOS and macOS apps.
14. To set per app VPN priority, see earlier in this section.

NOTE: Per app VPN settings are not supported for iOS apps when Core is configured for MAM-only iOS devices.

15. To set managed app configurations, modify the default configuration settings as required by your environment. This section displays only for apps which support managed app configuration. Optionally, click **Add+** to create alternative configuration settings with different values to apply to different devices based on labels.
    See Managed App Configuration settings for iOS apps in the App Catalog.

NOTE: Managed app configuration settings are not supported for iOS apps when Core is configured for MAM-only iOS devices.

16. Click **Save**.
17. Click the **Back to list** link to return to the App Catalog.
18. Select the **app**.
19. Click **Actions > Apply To Label** to set a label to the app in Apps@Work for devices associated with the label you select.

Related topics

Using the wizard to import iOS apps from the Apple App Store

Using the wizard to import iOS apps from the Apple App Store

You can use the Add App Wizard to import and configure iOS App Store apps in the App Catalog (rather than accept the default app settings). When the wizard finishes running, the apps are ready to be applied to labels and sent to Apps@Work as necessary.

NOTE: Although some settings listed here are supported by macOS apps, you cannot import public macOS apps directly into the App Catalog. Instead, use Apple Licenses to import macOS-licensed apps into Core. You can also import in-house apps for macOS. However, you can still edit certain settings for macOS apps in Core that are managed by Apple Licenses. These settings are noted in the following procedure.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Select **iOS** from the **Platform** list.
3. Click **Add+**.
4. Click **iTunes.**
5. To search for an app to import:
   a. Enter the name of the app or its iTunes ID. See Getting the iTunes app ID on page 75 for detailed instructions for getting the ID.
   b. From the App Store drop-down list, select the App Store country.
   c. Enter a limit for the number of search results (50 by default).
   d. Click **Search**.
6. Select the app from the search results list.
7. Click **Next**.
8. Use the following guidelines to complete this screen:

Item	Description
Application Name	Required. Shows the name of the app. You can edit this field only if you opted to manually provide all app details. The app name can be up to 255 characters long.
Min. OS Version	Indicates the minimum version of iOS the app can support. This field is only displayed if the .ipa file of the app you are importing includes a minimum OS version number. Required. NOTE: This field is not displayed for macOS apps.
Developer	Shows the name of the app developer. You can edit this field only if you opted to manually provide all app details.
Description	Enter any additional text that helps describe what the app is for.
iPad Only	Indicates whether the app is designed only for iPads. This ensures that the app is not displayed in Apps@Work for other iOS devices. You can edit this field only if you opted to manually provide all app details. NOTE: This field is not displayed for macOS apps.
Category	Select one or more categories to display this app in a category tab in Apps@Work or add a new category. a. Click **Add New Category** to define new categories. b. Enter a category **Name** (up to 64 characters). c. Enter a **Description** (up to 255 characters). d. In the **Category Icon** section, click the **Replace Icon** button. e. Browse and select an icon that will represent this Category. f. Click **Save**.

9. Click **Next**.
10. Use the following guidelines to complete this screen:

Item	Description
Apps@Work Catalog	
This is a Free App	Selected by default, this indicates free recommended Apple App Store apps. iOS allows Managed App features to be applied to free apps and apps purchased with Apple License credits, but not to apps paid for by the user. Specifying whether the app is free ensures successful download of apps that require user payment. NOTE: When importing recommended apps that use licenses, uncheck the **This App Store App is Free** option. This allows the device user to successfully download the app using licenses.
Hide this App from the Apps@Work catalog	Select to prevent this app from being displayed in Apps@Work. For example, you might want to hide apps that will be installed upon registration anyway. Hiding a mandatory app reduces clutter in Apps@Work, leaving device users with a concise menu of the approved apps they might find useful.
Allow conversion of apps from unmanaged to managed in Apps@Work (iOS 9 or later).	Select if you want to allow the app to be converted from an unmanaged app to an iOS managed app in Apps@Work on devices running iOS 9 through the most recently released version as supported by MobileIron. The unmanaged app will not require uninstallation, as it will be converted directly to an iOS managed app. NOTE: This setting is not applicable for macOS apps. It is also not displayed for iOS apps when Core is configured for MAM-only iOS devices.
Feature this App in the Apps@Work Catalog	Select if you want to highlight this app in the Featured apps list.
Featured Banner	Select to add the app to the featured banner at the top of the Apps@Work home screen on devices. When clicking the banner, device users see the details of the featured app. Add as many apps as you like to the featured banner, but the featured banner will only display the five most recent apps added to the featured banner. Apps in the featured banner are rotated every five seconds.

11. Click **Next**.
12. Use the following guidelines to complete this screen:

Item	Description
Per App VPN Settings	
Per App VPN by Label Only	Select to assign the device a VPN configuration by device label. Otherwise, de-select this option and prioritize the VPN configurations listed below. This feature is not currently supported on macOS devices.
VPN selection	In the left-hand column, select the VPN setting you created for per app VPN, and click the right arrow to move it to the set of selected VPNs in the right-hand column. If the app will use MobileIron Tunnel, select the MobileIron Tunnel VPN setting you created. You can select multiple per app VPN settings.

Item	Description
	To reorder the selected per app VPN configurations in the right-hand column, use the up and down arrows to sort the names in the list. See VPN settings in the *MobileIron Core Device Management Guide for iOS and macOS Devices* for information on creating a per app VPN or MobileIron Tunnel VPN setting. This feature is not currently supported on macOS devices. NOTE: Per app VPN settings are not displayed for iOS apps when Core is configured for MAM-only iOS devices.
License Required	Per-App VPN is supported only in iOS 7.0 and later and macOS 10.9 and later. Per-App VPN type IKEv2 (for iOS) is only supported in iOS 9.0 and above.
Managed App Settings	
Prevent backup of the app data	Select to ensure that iTunes will not attempt to back up possibly sensitive data associated with the given app. NOTE: This setting is not displayed for macOS apps. It is also not displayed for iOS apps when Core is configured for MAM-only iOS devices.
Remove app when device is quarantined or signed out	Select to enable configured compliance actions to remove the app if a policy violation results in a quarantined device or the device signs out in multi-user mode. To enable this feature, you must also configure a corresponding compliance action, and security policy with that compliance action selected. Once the device is no longer quarantined, the app can be downloaded again. Note The Following: • If you change the setting after the app is added, the changed setting will not be applied to the app. • This setting is not displayed for macOS apps. It is also not displayed for iOS apps when Core is configured for MAM-only iOS devices. For more information, see "Using Secure Sign-In and Sign-Out" in the *MobileIron Core Device Management Guide for iOS and macOS Devices.*
Send installation request or send convert unmanaged to managed app request (iOS 9 and later) on device registration or sign-in.	Select this option so that after device registration is complete, or after a user signs in on a multi-user device: • The device user is prompted to install this app. • If the app is already installed as an unmanaged app, the app will be converted to an iOS managed app. To allow conversion to an iOS managed app, select the option **Allow conversion of apps from unmanaged to managed in Apps@Work (iOS 9 or later)** in the **Apps@Work Catalog** section of the app's settings This setting is not selected by default.

Item	Description
	Note The Following: • This setting is not displayed for macOS apps. It is also not displayed for iOS apps when Core is configured for MAM-only iOS devices. • With this setting selected, a MDM profile re-push will cause apps to be re-installed. • For User Enrollment for Apple Business Manager, this field will only send installation request on device registration or sign-in. User Enrollment cannot convert unmanaged to managed apps. See *MobileIron Core Device Management Guide for iOS and macOS Devices* for more information.
Send installation or convert unmanaged to managed app request to quarantined devices	Select this option to enable the following on quarantined devices: • prompt the device user to install the app. • If the app is already installed as an unmanaged app, convert the app to an iOS managed app. Note The Following: • These settings are applied even if a compliance action blocks new app downloads for a quarantined device. • This setting is not displayed for iOS apps when Core is configured for MAM-only iOS devices.
Enforce conversion from unmanaged to managed app (iOS 9 or later)	Every hour, Core reviews the all the devices that had last checked-in for any unmanaged apps and, if applicable, sends the unmanaged to managed app conversion request to that device. If there is an unmanaged app installed on the device, device users will not immediately get the prompt for change management. Also applicable if the app is unmanaged on an iOS 9 and later device and the app is enabled to allow conversion.
Advanced Settings	
Remove app when MDM profile is removed	Select this option to remove this app from the device when the MDM profile is removed from the device. NOTE: This setting is not displayed for macOS apps. It is also not displayed for iOS apps when Core is configured for MAM-only iOS devices.

Item	Description
Update app when new version is available	Selecting this displays two additional fields. Select one or both of the following fields to enable updating the app when a new version is available: • Automatically update app when new version is available • Allow end user to (manually) update app through Apps@Work Applicable to public and private apps, including B2B apps. NOTE: This setting is not displayed for iOS apps when Core is configured for MAM-only iOS devices.
Managed App Configurations	
	This section displays only for apps that support managed app configuration. Modify the default configuration settings as required by your environment. Optionally, click **Add+** to create alternative configuration settings with different values to apply to different devices based on labels. See Managed App Configuration settings for iOS apps in the App Catalog. NOTE: Managed app configurations are not supported for iOS apps when Core is configured for MAM-only iOS devices.

13. Click **Finish**.
    The app displays in the **App Catalog**.
14. Associate the app with a label to list the app on iOS devices.

Related topics

- Changing iOS and macOS app information
- Changing the iOS or macOS app icon and screenshots
- Creating or changing a category for iOS and macOS apps

Next steps

Continue on to Making iOS and macOS apps available to users in Apps@Work.

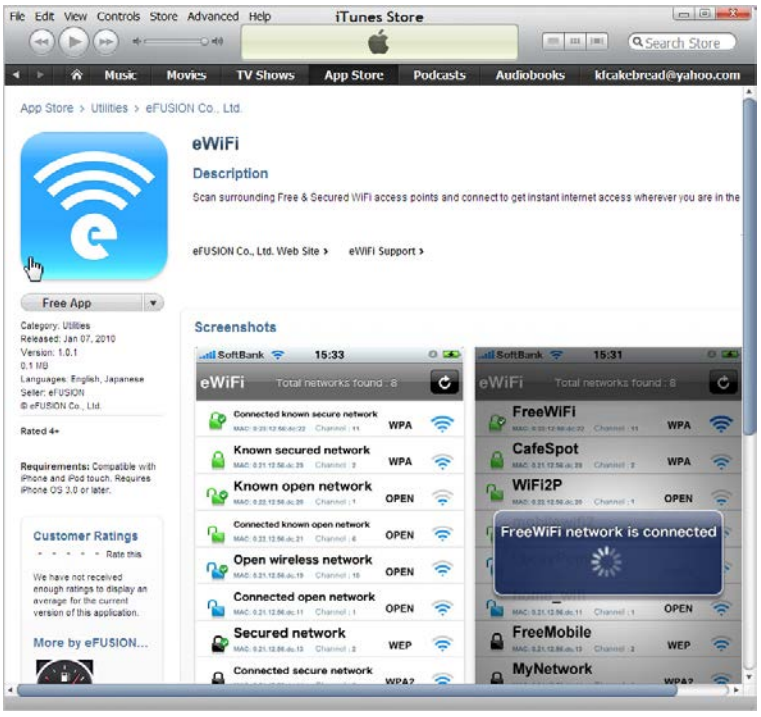
Getting the iTunes app ID

To manually configure a managed app in the Add App Wizard, you must supply the ID for the app as defined in iTunes. However, IDs are not always readily available.

Procedure

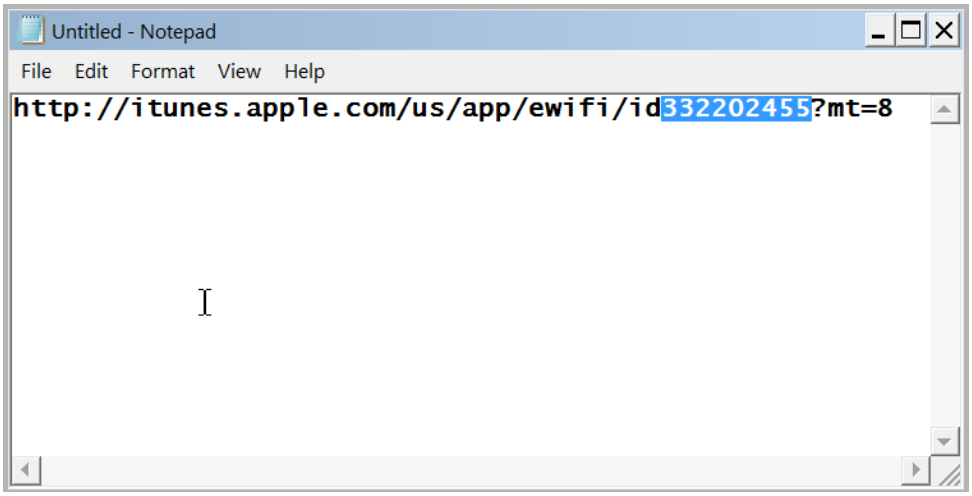
1. Open iTunes.
2. Navigate to the iTunes Store.
3. Navigate to the App Store.

4. Locate the app you want to configure.



5. Open a text editor.
6. Right-click the app icon, and select Copy Link.

7. For example, using Firefox, you can right-click on the icon and select Copy Link.
8. Paste the link into the text editor.
9. In the below example, the URL was pasted into Notepad. The application ID (selected) comprises the digits following “id” and before “?mt=8”.



Using the wizard to add an in-house iOS or macOS app to the App Catalog

You can use the Add App Wizard to import into the App Catalog in-house iOS and macOS apps developed by your organization. An enterprise-level Apple Developer account is required for developing in-house iOS and macOS apps.

Each in-house app for iOS must be no larger than 5 GB. Individual downloads of iOS in-house apps over 3G are generally limited to 20 MB per device. Use Wi-Fi to download larger in-house apps.

IMPORTANT: When developing an in-house iOS app to be used with Core, you must include the following keys in the info .plist file for the in-house app:

- `CFBundleName` or `CFBundleDisplayName`
- `CFBundleIdentifier`
- `CFBundleExecutable`
- `CFBundleVersion`
- `CFBundleShortVersionString`

If you are adding a new version of an existing app, see Adding new versions of an existing iOS or macOS app on page 87.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Select **iOS** or **macOS** from the **Platform** list.

3. Click **Add+**.
4. The **Add App Wizard** opens.
5. Click **In-House**.
6. Next to **Upload In-House App**, click **Browse** and navigate to the in-house iOS or macOS app (.ipa or .app) you want to upload.
7. Click **Next**.

Item	Description
Application Name	Displays the app name defined for the app bundle (up to 255 characters long). App names longer than 255 characters will be truncated when displayed on the device. NOTE: An iOS app is packaged as a bundle. A bundle is a directory in the file system that groups related resources together in one place. An iOS app bundle contains the app executable file and supporting resource files such as app icons, image files, and localized content.
Min. OS Version	Displays the minimum iOS version on which the in-house app can run. This value cannot be edited. For iOS apps only.
Display Version	Shows the version number displayed to users. The value of this field is a number with or without a period. This value cannot be edited.
Code Version	Displays the version of the app. This value cannot be edited. This value is not displayed for macOS in-house apps.
Developer	Enter the name of the app developer.
Description	Enter any additional text that describes the app.

Item	Description
iPad Only	Select if the app is designed only for iPads. This ensures that the app is not displayed in Apps@Work for other iOS devices. For iOS apps only.
Provisioning Profile	Shows the identifier for the provisioning profile incorporated in the bundle, and the expiration date of the provisioning profile. This value cannot be edited when adding a new app. You can only upload a current profile when editing an app with an expired provisioning profile that you have previously added to the App Catalog. To upload a current provisioning profile, you must complete the process of adding the app to the App Catalog, and then edit the app. The **Provisioning Profile Status** column in the App Catalog shows a red warning icon to indicate that the provisioning profile of a given managed app is expired, or otherwise invalid. If the provisioning profile is expired, you can upload a new profile as follows: 1. Click on the app name in the App Catalog. 2. Click **Edit**. 3. Click **Upload a New Profile** next to the **Provisioning Profile** field. 4. In the dialog that appears, browse for the new profile. 5. Click **Save**. NOTE: : After uploading a provisioning profile, the profile is not deleted from Core if you cancel out the app edit page. The provisioning profile will fail to upload if its expiration date is in the past, or if the profile is app-specific and does not match the bundle ID of the app itself. For more information about provisioning profiles, see Provisioning profiles for in-house iOS apps. For iOS apps only.
Category	Select one or more categories if you would like this app to be displayed in a specific group of apps on the device or add a new category: 1. Click **Add New Category** to define new categories. 2. Enter a category **Name** (up to 64 characters). 3. Enter a **Description** (up to 255 characters). 4. In the Category Icon section, click the **Replace Icon** button. 5. Browse and select an icon that will represent this Category. 6. Click **Save**. See Creating or changing a category for iOS and macOS apps for more information.

8. Click **Next**.
9. Use the following guidelines to complete this screen:

Item	Description
Apps@Work Catalog	
Hide this App from the Apps@Work catalog	Select to prevent this app from being displayed in Apps@Work. For example, you might want to hide apps that will be installed upon registration anyway. Hiding a mandatory app reduces clutter in Apps@Work, leaving device users with a concise menu of the approved apps they might find useful.
Allow conversion of apps from unmanaged to managed in Apps@Work (iOS 9 or later).	Select if you want to allow the app to be converted from an unmanaged app to an iOS managed app in Apps@Work on devices running iOS 9 through the most recently released version as supported by MobileIron. The unmanaged app will not require uninstallation, as it will be converted directly to an iOS managed app. For iOS apps only. NOTE: This setting is not displayed for iOS apps when Core is configured for MAM-only iOS devices.
Feature this App in the Apps@Work Catalog	Selected by default, this check box indicates this app will be highlighted in the Featured Apps list.
Featured Banner	Select to add the app to the featured banner at the top of the Apps@Work home screen on devices. When clicking the banner, device users see the details of the featured app. Add as many apps as you like to the featured banner, but the featured banner will only display the five most recent apps added to the featured banner. Apps in the featured banner are rotated every five seconds.
Allow app downloads over insecure networks	Select this if you are providing an Override URL (next field) that uses the HTTP URL scheme instead of HTTPS. Override URLs are intended for use behind a firewall, using a trusted and secure internal network. Before you use an HTTP URL, make sure you understand the risks of using an insecure connection.
Override URL	If you are using an alternate source for downloading in-house apps, enter that URL here. The URL must point to the in-house app in its alternate location. Override URLs are intended for use behind a firewall, using a trusted and secure internal network. Manual synchronization is required with the alternate HTTP server on which app are stored. See Override for in-house app URLs for the requirements for this configuration before using it.
Icon and Screenshots	

Item	Description
App Icon	Required. The app icon is automatically extracted from the IPA file. The file must be in PNG format. Click **Replace Icon** to replace the icon.
iPhone Screenshots	Click **Upload** to add a an iPhone screenshot. Select up to twelve optional screenshots to display for the app. Screenshots must be in JPG, PNG, or GIF format, with a minimum size of 320x480 pixels or 480x320 pixels, and a maximum size of 4096x4096 pixels. Click **Remove** to delete a screenshot. NOTE: The display of rotated screenshots in the Admin Portal might not be consistent with the display on devices.
iPad Screenshots	Click **Upload** to add a screenshot. Select up to twelve optional screenshots to display for the app. Screenshots must be in JPG, PNG, or GIF format. Each file must have a minimum size of 1024x768 or 768x1024 pixels, and a maximum size of 4096x4096 pixels. Click **Remove** to delete a screenshot. NOTE: For macOS apps, follow the instructions for iPad Screenshots.

10. Click **Next**.
11. Use the following guidelines to complete this page:

Item	Description
Data Protection Required	Select to require data protection to install this app. For iOS apps only. Note The Following: • Devices without data protection enabled will not see the app at all in the In-house Apps list and will not know that data protection compliance is required. Therefore, it is recommended to communicate this requirement directly to users. • Requiring data protection is not supported for MAM-only iOS devices.
Per App VPN Settings	
Per App VPN by Label Only	This feature requires a separate MobileIron license, and is not supported on macOS devices. In the left-hand column, select the VPN setting you created for per app VPN, and click the right arrow to move it to the set of selected VPNs in the right-hand column. If the app will use MobileIron Tunnel, select the MobileIron Tunnel VPN setting you created. You can select multiple per app VPN settings. To reorder the selected per app VPN configurations in the right-hand column, use the up and down arrows to sort the names in the list.

Item	Description
	See VPN settings in the *MobileIron Core Device Management Guide for iOS and macOS Devices* for information on creating a per app VPN or MobileIron Tunnel VPN setting. NOTE: Per app VPN settings are not displayed for iOS apps when Core is configured for MAM-only iOS devices.
License Required	Per-App VPN is supported only in iOS 7.0 and later and macOS 10.9 and later. Per-App VPN type IKEv2 (for iOS) is only supported in iOS 9.0 and above.
Managed App Settings	
Prevent backup of the app data	Select to ensure that iTunes will not attempt to back up possibly sensitive data associated with the given app. No further action is necessary to apply this restriction. For iOS apps only. NOTE: This setting not displayed for iOS apps when Core is configured for MAM-only iOS devices.
Remove app when device is quarantined or signed out	Select to enable configured compliance actions to remove the app if a policy violation results in a quarantined device or the device signs out in multi-user mode. To enable this feature, you must also configure a corresponding compliance action, and security policy with that compliance action selected. Once the device is no longer quarantined, the app can be downloaded again. For iOS apps only. Note The Following: • If you change the setting after the app is added, the changed setting is not applied to the app. • This setting is not displayed for iOS apps when Core is configured for MAM-only iOS devices.
Send installation request or send convert unmanaged to managed app request (iOS 9 and later) on device registration or sign-in.	Select this option so that after device registration is complete, or after a user signs in on a multi-user device: • The device user is prompted to install this app. • If the app is already installed as an unmanaged app, the app will be converted to an iOS managed app. To allow conversion to an iOS managed app, also select the option **Allow conversion of apps from unmanaged to managed in Apps@Work (iOS 9 or later)** in the **Apps@Work Catalog** section of the app's settings. This setting is not selected by default. This field is not applicable to User Enrollment for Apple Business Manager - see *MobileIron Core Device Management Guide for iOS and macOS Devices* for more information. For iOS apps only.

Item	Description
	NOTE: This setting is not displayed for iOS apps when Core is configured for MAM-only iOS devices.
Send installation or convert unmanaged to managed app request to quarantined devices	Select this option to enable the following on quarantined devices: • prompt the device user to install the app. • If the app is already installed as an unmanaged app, convert the app to an iOS managed app. For iOS apps only. Note The Following: • These settings are applied even if a compliance action blocks new app downloads for a quarantined device. • This setting is not displayed for iOS apps when Core is configured for MAM-only iOS devices.
Enforce conversion from unmanaged to managed app (iOS 9 or later)	Upon device check-in, send installation request to convert unmanaged to managed app. Also applicable if the app is unmanaged on an iOS 9 and later device and the app is enabled to allow conversion.
Advanced Settings	
Remove app when MDM profile is removed	Select this option to remove this app from the device when the MDM profile is removed from the device. For iOS apps only. NOTE: This setting is not displayed for iOS apps when Core is configured for MAM-only iOS devices.

12. Click **Finish**.
    The app displays in the **App Catalog**.
    The provisioning profile for the app is also stored on Core and displays in the **Policies & Configs > Configurations** page. It is displayed for viewing only, and is automatically deleted from Core if the app is deleted from Core.
13. Select the app in the App Catalog.
14. Click **Actions > Apply to Label,** and select the appropriate labels to make this app available to device users.

NOTE: You can edit the app's settings at any time. Select the app in the App Catalog, and click **Edit**.

Next steps

Making iOS and macOS apps available to users in Apps@Work

Related topics

- developing in-house iOS apps, see the Apple enterprise developer site at
- https://developer.apple.com/enterprise/
- building, signing, uploading, installing, and launching in-house apps, see this knowledge base article:
- https://community.mobileiron.com/docs/DOC-2073

Using the wizard to add an in-house macOS bundled app to the App Catalog

You can use the Add App Wizard to import into the App Catalog signed, in-house, bundled macOS apps. An enterprise-level Apple Developer account is required for developing in-house macOS apps.

If you are adding a new version of an existing app, see Adding new versions of an existing iOS or macOS app on page 87.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Select **macOS** from the **Platform** list.
3. Click **Add+**.
4. The **Add App Wizard** opens.
5. Click **In-House**.
6. Click **Browse** and navigate to the in-house bundled macOS app you want to upload.
7. Click **Next**.

Item	Description
Application Name	Displays the app name defined for the app bundle (up to 255 characters long). App names longer than 255 characters will be truncated when displayed on the device. NOTE: A macOS app is packaged as a bundle. A bundle is a directory in the file system that groups related resources together in one place. A macOS app bundle contains the app executable file and supporting resource files such as app icons, image files, and localized content.
Display Version	Shows the version number displayed to users. The value of this field is a number with or without a period. This value cannot be edited.
Developer	Enter the name of the app developer.

Item	Description
Description	Enter any additional text that describes the app.
Category	Select one or more categories if you would like this app to be displayed in a specific group of apps on the device. 1. Click **Add New Category** to define new categories. 2. Enter a category Name (up to 64 characters). 3. Enter a Description (up to 255 characters). 4. In the Category Icon section, click the Replace Icon button. 5. Browse and select an icon that will represent this Category. 6. Click Save. See Creating or changing a category for iOS and macOS apps for more information.

8. Click **Next**.
9. Use the following guidelines to complete this screen:

Item	Description
Apps@Work Catalog	
Hide this App from the Apps@Work catalog	Select to prevent this app from being displayed in Apps@Work. For example, you might want to hide apps that will be installed upon registration anyway. Hiding a mandatory app reduces clutter in Apps@Work, leaving device users with a concise menu of the approved apps they might find useful.
Feature this App in the Apps@Work Catalog	Select if you want to highlight this app in the Featured apps list.
Featured Banner	Select to add the app to the featured banner at the top of the Apps@Work home screen on devices. When clicking the banner, device users see the details of the featured app. Add as many apps as you like to the featured banner, but the featured banner will only display the five most recent apps added to the featured banner. Apps in the featured banner are rotated every five seconds.
Allow app downloads over insecure networks	Select this if you are providing an Override URL (next field) that uses the HTTP URL scheme instead of HTTPS. Override URLs are intended for use behind a firewall, using a trusted and secure internal network. Before you use an HTTP URL, make sure you understand the risks of using an insecure connection.
Override URL	If you are using an alternate source for downloading in-house apps, enter that URL here. The URL must point to the in-house app in its alternate location. Override URLs are intended for use behind a firewall, using a trusted and secure internal network. Manual synchronization is required with the alternate HTTP server on which app are stored. See Override for in-house app URLs for the requirements for this configuration before using it.

Item	Description
Icon and Screenshots	
App Icon	Required. The app icon is automatically extracted from the IPA file. The file must be in PNG format. Click **Replace Icon** to replace the icon.
Screenshots	Select up to twelve optional screenshots to display for the app. Screenshots must be in JPG, PNG, or GIF format. Each file must have a minimum size of 1024x768 or 768x1024 pixels, and a maximum size of 4096x4096 pixels. Click **Upload** to add a screenshot.

10. Click **Next**.
11. Use the following guidelines to complete this page:

Item	Description
Send installation request on device registration	Select this option to prompt macOS device users to install this app after device registration is complete. NOTE: If using Mobile@Work 1.4 for macOS, it is recommended you select this check box.
Per App VPN Settings	
Per App VPN by Label Only	This feature requires a separate MobileIron license. In the left-hand column, select the VPN setting you created for per app VPN, and click the right arrow to move it to the set of selected VPNs in the right-hand column. If the app will use MobileIron Tunnel, select the MobileIron Tunnel VPN setting you created. You can select multiple per app VPN settings. To reorder the selected per app VPN configurations in the right-hand column, use the up and down arrows to sort the names in the list. See VPN settings in the *MobileIron Core Device Management Guide for iOS and macOS Devices* for information on creating a per app VPN or MobileIron Tunnel VPN setting.

12. Click **Finish**.
    The app bundle displays in the **App Catalog**.
13. Associate the app with a label to list the app on macOS devices.

    NOTE: In the Apply to Labels dialog box, select the check box next to the app's name. Click in the **Mandatory** field, a drop-down displays. Selecting **Yes** makes the selected app mandatory; leaving it to the default **No** makes the app optional.

Next steps

Making iOS and macOS apps available to users in Apps@Work

App management action workflows

Adding new versions of an existing iOS or macOS app

When uploading a newer version of an app, an extra page opens to allow you to select whether to keep the app's old version information or to adopt the information from the app's new version. This feature is applicable to iOS and macOS in-house / private / self-hosted apps.

Procedure

1. In the App Catalog, click the **Add+** button.
   The Add App Wizard opens.
2. Click **In-House**.
3. Click **Browse** and navigate to the in-house iOS or bundled macOS app you want to upload.
4. Click **Next**.
   The An earlier version of this App exists page opens.
5. Select an option:
   - **Another version of this App was previously uploaded. Reuse its description, icon and screenshot(s).** If the Description, Icon or Screenshot fields of the new app are empty, then the system will populate those fields with information from the previous app version (default).
   - **Upload a new description, icon or screen shot.** Information related to the Description, Icon or Screenshot fields of the new App will be utilized. If those fields are empty, nothing will be copied from the previous app version.
6. Click **Next** and finish configuring the new version of your app (see Using the wizard to add an in-house iOS or macOS app to the App Catalog on page 77.)
   Once finished, the new version displays in the App Catalog.

Setting per app VPN priority for iOS and macOS apps

The per app VPN settings the app uses depends on:

- The label to which the per app VPN setting is applied (if the per app VPN is applied to a label).
- The assigned priority of the per app VPN setting in the Per App VPN field of the app.
- The first per app VPN listed in the right-hand column of the **Per App VPN Settings** has the highest priority; the last per app VPN has the lowest priority.

NOTE: To rearrange the per app VPN settings' priorities in the right-hand column (set of selected VPN settings), drag the setting names to the correct positions in the list. You can also use the up and down arrows.

The priority of per app VPN settings applied to labels is higher than per app VPN settings that are not applied to labels. For example, suppose the app lists VPN1, VPN2 and VPN3 as the possible per app VPN settings in the right-hand column (set of selected VPN settings).

- If VPN1 and VPN2 are applied to labels and VPN3 is not, then VPN1 is assigned to the app when the per app VPN list order is:
  - VPN1 (applied to label)
  - VPN2 (applied to label)
  - VPN3
- If VPN1 and VPN2 are applied to labels and VPN3 is not, then VPN1 is assigned to the app if the per app VPN list is:
  - VPN3
  - VPN1 (applied to label)
  - VPN2 (applied to label)

The **Apps** tab in device details (go to **Devices & Users > Devices** and click the caret to see the device details) lists the activated per app VPN for the device so that users and administrators can easily view which VPN the app is using on that device.

NOTE: Per app VPN is not supported for iOS apps when Core is configured for MAM-only iOS devices.

Per app VPN and the MobileIron Tunnel app on iOS and macOS devices

MobileIron Core pushes per app VPN profiles to devices regardless of whether devices have the VPN client (MobileIron Tunnel). Core will install apps to devices that require MobileIron Tunnel to function correctly, even if those devices do not have Tunnel installed or per app VPN enabled. If MobileIron Tunnel is not installed to devices with these apps, the apps will not function correctly. To enable the use of apps that require MobileIron Tunnel type per app VPN to function, you must ensure devices have MobileIron Tunnel installed and per app VPN functionality enabled.

MobileIron makes the following recommendations with regard to apps requiring per app VPN:

- When sending app installation messages to devices for apps requiring MobileIron Tunnel type per app VPN, Core installs the apps to devices even if Tunnel or per app VPN is not installed or enabled on these devices. To send app installation messages only to devices with MobileIron Tunnel type per app VPN, you must send the app installation message to a label you create that includes only devices with MobileIron Tunnel type per app VPN.
- When sending an app installation or conversion request (from unmanaged to managed) on registration or sign-in, Core installs to devices apps requiring Tunnel or per app VPN regardless of whether devices have Tunnel installed or per app VPN enabled. To send app installation or conversion requests only to devices with MobileIron Tunnel type per app VPN configurations, you must send the app installation or conversion message to a label you create that includes only devices with MobileIron Tunnel type per app VPN.
- When signing out of the multi-user web clip for iOS, Core triggers the removal of the per app VPN profile from the device twice.

- Apply the following dynamic label to the VPN configuration profile you apply to devices: `"common.mi_tunnel_app_installed" = "production"`
- When configuring per app VPN settings to an app, select Per app VPN by label only, then select the MobileIron Tunnel VPN configuration. You must move only the MobileIron Tunnel VPN configuration to the right side of per app VPN list, as Core does not support this functionality if other types of VPN configurations exist on the device.

Removing iOS or macOS apps from the App Catalog

Removing an app from the App Catalog removes the listing for the app from Apps@Work on iOS and macOS devices.

WARNING:

- Deleting apps from MobileIron Core also causes these apps to be uninstalled from devices to which the apps are installed.
- Deleting Apple License apps will cause the Apple Licenses associated with these apps to be reclaimed.
- Unmanaged iOS apps are **not** deleted from iOS devices.
- Removing a macOS app from the App Catalog deletes the app from Core only, and not from macOS devices, as macOS apps are not managed apps.
- On MAM-only iOS devices, iOS apps are removed from Apps@Work on the device, but **not** uninstalled from the device.

Procedure

1. In the Admin Portal, go to **Apps> App Catalog**.
2. Select **iOS** or **macOS** from the **Platform** list.
3. Select the app you want to remove.
4. Click **Delete**.
   A message displays warning that deleting the app from MobileIron Core will delete it from devices.
5. Click **Yes** to proceed.
6. For in-house apps, the app bundle and the provisioning profile are removed from MobileIron Core.

Making iOS and macOS apps available to users in Apps@Work

Note The Following:

- Making macOS app available to users in Apps@Work involves the use of Apple Licenses. For more information about publishing macOS apps to macOS devices, see Using Apple licenses.
- You can also import in-house apps for macOS. See Using the wizard to add an in-house iOS or macOS app to the App Catalog.

Managing iOS apps in Apps@Work involves:

Publishing iOS and macOS apps to Apps@Work

After adding any iOS or macOS app to the App Catalog, the app must be made available to the relevant users through Apps@Work. This is done by applying the app to a relevant label. The label determines the group of device users who will see the app in Apps@Work on their iOS or macOS devices.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Select **iOS** or **macOS** from the **Platform** list.
3. Select the app you want to work with.
4. Click **Actions > Apply to Label**.
5. Select the label that represents the iOS or macOS devices for which you want the selected app to be displayed.

NOTE: In the Apply to Labels dialog box, select the check box next to the app's name. Click in the **Mandatory** field, a drop-down displays. Selecting **Yes** makes the selected app mandatory; leaving it to the default **No** makes the app optional.

6. Click **Apply**.

Updating iOS apps in Apps@Work

When an update for an iOS managed app becomes available, you can update the iOS managed app in the App Catalog, as described in Manually importing iOS apps from the Apple App Store. MobileIron Core sends the update information to Apps@Work on devices associated with the same label as the updated app.

NOTE: When you update a managed app configuration, the changes will be pushed to all associated devices, including a newer version of that app found in the AppConfig repository. MobileIron recommends you first inspect the updated Managed App configuration schema in the AppConfig repository before proceeding.

Apps@Work includes an **Updates** category, where it lists iOS managed apps that are available for update. The list of iOS managed apps with updates displays when the user taps the **Updates** category.

An **Update** tag displays on the entry for the app with an update.

Updates to featured apps are published in the same way to all devices in the labels assigned to the apps. You can also send a message to devices to announce the availability of updates to featured apps (see Notifying users of new iOS and macOS apps or app updates).

Apps have a Re-Import link, allowing you to re-import the app into MobileIron Core at any time.

Note, however, that Core does **not**:

- contact the Apple App Store to check for updates to unmanaged apps,
- track the version numbers of public apps
- control which version a user can install.
- support updating apps on MAM-only iOS devices. Specifically, if you update the app in the App Catalog, Apps@Work on the device makes the updated app available, but does not indicate that it is an update. You also cannot send a message to the devices that an update is available.

MobileIron recommends that device users consult the Apple App Store to confirm the availability of new versions of apps. Alternatively, you can use the send message feature to inform device users of a new version of an app. For more information about using the send message feature, see Informing users of new apps and updates on iOS and macOS devices.

Unpublishing iOS apps from Apps@Work

You can unpublish an iOS app from Apps@Work by removing the app from the label to which it was originally applied.

Removing an app from a label causes that app to be uninstalled from the devices associated with that label, and removes the apps listed in Apps@Work on devices.

Note The Following:

- On MAM-only iOS devices, iOS apps are removed from Apps@Work on the device, but **not** uninstalled from the device.
- You cannot unpublish macOS apps from macOS devices, as macOS apps are not managed apps.

When you apply a label to an app, Core makes that app available to devices associated with that label. After the label is applied, device users can install the app from Apps@Work. Simply applying the label to the app does not install the app to devices belonging to the label.

If you want to install an app to all devices in a label, you can do so by sending a message to devices, or configuring the app to be installed upon sign-in (for multi-user devices).

NOTE: If you remove an app from a label, and then decide you want to apply the label to the app after all, the app will still be uninstalled and removed from devices. Re-applying the label to the app causes the app to be available in Apps@Work on the devices associated with that label. Re-applying the label to the app will not install the app to devices on that label. To re-install the app to devices on the label, send a message to devices or configure the app to be installed upon multi-user sign-in.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Select **iOS** from the **Platform** list.
3. Select the app you want to work with.
4. Click **Actions > Remove from Label.**
5. Select the labels from which you want to remove the app.
6. Click **Remove**.
   The app is immediately removed from the apps list on the devices associated with the given label.

Related topics

- App management action workflows
- Informing users of new apps and updates on iOS and macOS devices
- Manually importing iOS apps from the Apple App Store

Mandatory and optional in-house and secure apps

iOS and macOS in-house apps are made available through the **App Catalog** and can be designated as a *mandatory* app, which means that the app is always installed on the devices matching the app's labels. An app that is not marked as mandatory is *optional*, and enables the users to decide whether or not to install the app on their devices. The in-house app can be either an AppConnect app (secure app) or a regular, non-AppConnect app.

NOTE: Designating the Secure Apps Manager as optional and all secure apps as optional means that the device user sets up the secure apps container on-demand. See On-demand secure apps container setup.

NOTE: To set the prerequisite app for a dependent app, see App management action workflows.

Install and uninstall of mandatory apps

You can specify that mandatory in-house apps and secure apps are installed and uninstalled on iOS and macOS devices.

Although a mandatory app is always installed on the device, whether the device user sees a notification to install the app depends on whether the device is a supervised device.

Whether device users are notified to install a mandatory app

When a iOS or macOS app is set as Mandatory (the Mandatory field is set for the label that is applied to the app), device users will not immediately get the prompt for app installation if they have do not have that app installed on the device.

Every hour, Core reviews the all the devices that had last checked-in for any unmanaged apps and, if applicable, sends the unmanaged to managed app conversion request to that device. If there is an unmanaged app installed on these devices, device users will not immediately get the prompt for change management.

Device user experience with uninstalling a mandatory app

The device user experience when attempting to uninstall a mandatory app depends on the type of device, as specified by the following table:

TABLE 10. DEVICE USER EXPERIENCE WITH UNINSTALLING A MANDATORY APP

	iOS devices that support install/uninstall	macOS devices that support install/uninstall
Can device user uninstall a mandatory app when the install/uninstall feature **is** enabled?	Yes, but the app will be reinstalled.	Yes, but the app will be reinstalled.
Can device user uninstall a mandatory app when the install/uninstall feature is **not** enabled?	Yes, but the device user will be notified to re-install the app.	Yes, but the device user will be notified to re-install the app.

Designating an in-house app as optional or mandatory

After you have added the app to the App Catalog, you can designate whether it is an optional or mandatory app.

The below procedure applies to both iOS and macOS in-house and public apps.

Procedure

1. In the MobileIron Core Admin Portal, go to **Apps > App Catalog**.
2. Select an app and then select **Actions > Apply to Labels**.
3. In the Apply to Labels dialog box, select the check box next to the app's name.
4. Click in the **Mandatory** field, a drop-down displays. Selecting **Yes** makes the selected app mandatory; leaving it to the default **No** makes the app optional.
5. Click **Apply**.

Enforcement of specific iOS and macOS app versions for mandatory in-house apps

You can configure a mandatory iOS and macOS in-house app to limit its installation on devices to a specific version of the app, even if newer or older versions of the same app .ipa are uploaded to the MobileIron Core's app catalog. You can also ensure that any version of the same app is installed, regardless of which version. This option, called **Enforce this version for Mandatory Apps**, is available in the App Catalog app wizard.

The version enforcement feature is supported only with regular (non-AppConnect) in-house apps. It does not apply to AppConnect apps or Google Play apps.

Use the version enforcement feature to:

- Ensure devices have the in-house app installed, regardless of version number.
- Lock users to a particular version of the Mobile@Work app. This applies to organizations that install Mobile@Work as an in-house app instead of installing it from iTunes.
- Ensure users do not upgrade to a new version of an in-house app while the newer version is still undergoing testing.
- Downgrade users to a previous version of an in-house app.

Setting up version enforcement for an in-house app

You can enable or disable enforcing a specific app version for an in-house app on an iOS or macOS device when you upload the app to MobileIron Core.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog.**
2. Click **Add+.**
3. Click **In-House.**
4. Click **Browse...** to select your in-house app. (Must not be an AppConnect app.)
5. Fill out the app wizard as needed; under **App Installation Settings**, select **Enforce this version for Mandatory Apps.** If this check box is not selected, then enforcing a specific app version will not apply. See: Enforcing an app version when you have uploaded multiple versions to Core.
6. Finish filling out the app wizard as needed. Click **Finish**.
7. Select the app in the App Catalog.
8. Click **Actions > Apply to Labels.**
9. In the Apply to Labels dialog box, select the check box next to the app's name.
10. Click in the **Mandatory**field, a drop-down displays. Selecting **Yes** makes the selected app mandatory; leaving it to the default **No**makes the app optional. If the Mandatory field is not set to **Yes**, the latest version of the app will not be enforced.
11. Click **Apply**.

Enforcing an app version when you have uploaded multiple versions to Core

If you have multiple versions of the same mandatory in-house .ipa file uploaded to MobileIron Core, you may wish to ensure one of the following scenarios:

- Devices always get the latest version of the app (app updates are forced.)
- Devices have the app installed, regardless of the version number (app updates are not forced.)

- Devices remain on an older version of the app.
- Devices are downgraded to an older version of the app.

Assuming your in-house app has versions 1.0, 2.0, and 3.0 in order from oldest to newest, and all three are uploaded to Core, use the settings described in the following table to achieve the desired results.

Note that having a label means that same label is applied both to the device and to the app. If a device is assigned to many labels, but at least one label has the Mandatory field set to **Yes**, then the device will have that app as mandatory.

TABLE 11. LABEL AND APP SETTINGS IN APP CATALOG

Desired Result	Label and app settings (in App Catalog)
Ensure that any version of the app is installed on the device	For app version 1.0: Enforce this version is not selected For app version 2.0: Enforce this version is not selected For app version 3.0: Enforce this version is not selected Label must be applied to any or all versions of the app.
Allow only version 2.0	For app version 1.0: Enforce this version: irrelevant For app version 2.0: Enforce this version is selected For app version 3.0: Enforce this version: irrelevant Label must be applied to app version 2.0 only. Label must not be applied to all other app versions.
Ensure the latest version is always installed	Enforce this version is selected on the most recent app version (3.0). Enforce this version is irrelevant on older app versions (1.0, 2.0). Label must be applied to latest app version (3.0) Label may be applied to all app versions.
Downgrade users to version 1.0	App version 1.0: Enforce this version is selected; Label is applied. App version 2.0: Label is removed. App version 3.0: Label is removed.

Managing installed iOS and macOS apps

Managing installed iOS apps involves:

NOTE: When Core is configured for MAM-only iOS devices, you cannot view installed app status.

Viewing the status of installed iOS and macOS apps

The Installed Apps page shows, at a glance, which iOS and macOS apps are installed to managed iOS and macOS devices, respectively. Managed devices send the status of their apps to MobileIron Core, and the Installed Apps page indicates the number of devices to which the apps are installed. For instance, if an app has been installed to one managed device, this is indicated by the number 1 and the time stamp for when Core received the installation status. If a given app has not been installed to any device, that app is not displayed on the Installed Apps page.

You can search for apps using the following:

- Summary View: This search is based on app ID or bundle ID.
- Detailed View: This search is based on the app installation name, meaning the name of the app as installed on a device (as opposed to the App Catalog name, which may be different).

Procedure

1. In the Admin Portal, go to **Apps > Installed Apps**.
2. Select **iOS** or **macOS** from the **Platform** list.
3. The list of installed iOS or macOS apps displays the following information in **Summary View**, including the following columns:

Item	Description
Application Name	The name of the application.
Identifier	The bundle identifier for the application.
Platform	The operating system on which the app is designed to run: iOS or macOS.
Devices Installed	The number of devices to which this app is installed.
First Found	The date and time at which a registered device first reported the app to MobileIron Core.

4. To view more details about installed iOS apps, under **App Detail**, select the **Detailed View** radio button. The list of installed iOS apps displays in Detailed View, including the following columns:

Item	Description
Application Name	The name of the application.
Identifier	The bundle identifier for the application.
App Version	The version number of the installed app.
Platform	The operating system on which the app is designed to run: iOS or macOS.
Devices Installed	The number of devices to which this app is installed.
Permissions	For Android apps only.
First Found	The date and time at which a registered device first reported the app to MobileIron Core.

You can optionally sort the list of apps by any of the available columns.

5. To view details about the devices to which the app is installed, click the number in the **Devices Installed** column. The **Device Details** window displays the following information for each device:

Item	Description
Device UUID	The unique identifier for the device.
User Name	The name of the user to whom the device is registered.
User ID	The user ID of the user to whom the device is registered.
Platform	The version of the operating system installed to the device.
Model	The model name and number of the device.
Mobile Number	The mobile phone number associated with the device. NOTE: This setting is not displayed for macOS apps.
App Version	The version number of the installed app.

6. You can take any of the following actions on the devices shown here, by selecting one or more devices and then selecting one of the following options:
   - **Send Message**
   - **Force Device Check-In**
   - **Retire**
7. Click **Apply**.
8. Click **Export to CSV** to export the list of devices to a CSV file.
9. Click **Close**.

Selecting which installed iOS apps to track

By default, the App Catalog page lists all apps installed to all managed iOS devices. However, you can filter the types of iOS apps whose installation status you want to log on this page. For example, you can allow MobileIron Core to track the status of all apps, or only iOS managed apps, or certain apps specified by their bundle IDs.

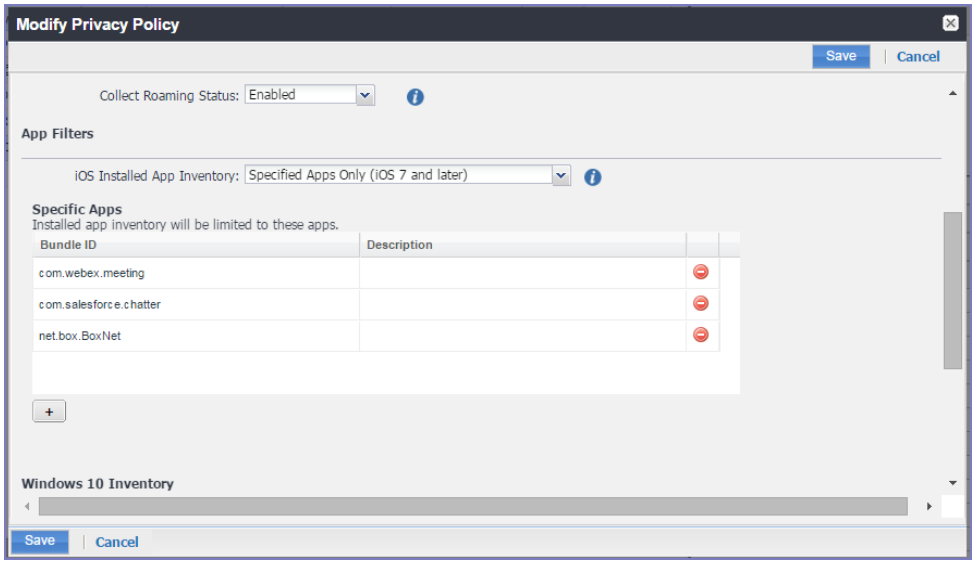
Procedure

1. In the Admin Portal, go to **Policies & Configs > Policies**.
2. Select the default Privacy policy.
3. Click **Edit**. The Modify Privacy Policy dialog box opens.
4. Scroll down to the **App Filters** section.
5. From the **iOS Installed App Inventory** drop-down list, select one of the following options:
   - All Apps: Managed devices will send the status of all iOS managed installed apps and unmanaged installed apps to MobileIron Core.

     NOTE: The administrator should update the Default Privacy Policy with highest priority and set the "iOS Installed App Inventory" field to "All Apps". This ensures unmanaged apps are reported to the Core and the Core can send MDM commands to convert unmanaged apps to managed apps. This applies if the app has the option "Enforce conversion from unmanaged to managed app (iOS 9 or later)" enabled and the unmanaged version of the same app is reported to the Core.

   - **Managed Apps Only (iOS 7 and later)**: Managed devices will only send the status of installed iOS managed apps to MobileIron Core.
   - **Specified Apps Only (iOS 7 and later)**: Managed devices will send to MobileIron Core the status of installed apps (whether iOS managed apps or unmanaged apps) with the specified identifiers (bundle IDs).
6. If you selected **Specified Apps Only (iOS 7 and later)**, a table called **Specific Apps** displays.
   a. From the **Bundle ID** drop-down list, select the identifier for the app you want to track.
   b. In the Specific Apps table, click the **+** icon to add the app identifiers whose status you wish to track.
   c. In the **Description** field, enter a brief description for the app.

d. To remove an entry, click the delete icon.



7. Click **Save**.
   A prompt displays, indicating that users will receive notification of the changes to the privacy policy.
8. Click **Yes**.

NOTE: The default policy is applied to the All-smartphones label and labels to which no other policy has been applied.

Related Topic:

- App management action workflows

Editing iOS and macOS apps and app settings in the App Catalog

You can edit iOS app settings as follows:

- Changing iOS and macOS app information
- Changing the iOS or macOS app icon and screenshots
- Creating or changing a category for iOS and macOS apps

Changing iOS and macOS app information

NOTE: You cannot edit the iTunes ID of an app. If you entered the wrong ID when you added this app to the App Catalog, then you need to delete the app entry and create a new one.

When the app data is in View or Edit mode, Core loads the latest managed app schema from the AppConfig repository and displays the latest fields (including any new fields) in the “Managed App Configurations” section in

the UI. MobileIron recommends that before saving the changes, you first carefully inspect the updated managed app configuration. Once you select Proceed and click Confirm, the updated managed app configuration settings are saved and the changes are pushed out to all associated devices.

NOTE: Prevent deleted default field values from repopulating by entering the substitution variable **$NULL$** when editing app configurations. See Substitution variables for configuring iOS apps.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Select **iOS** or **macOS** from the **Platform** list.
3. Click the name of the app.
4. Click **Edit**.
5. Make your changes.

Field name	Can be modified?
Description	
App Name	Yes
Display Version	No
Code Version	No
Developer	No
Description	Yes
Category	Yes
Apps@Work Catalog	
Hide this App from the Apps@Work catalog	Yes
Feature this App in the Apps@Work catalog	Yes
Featured Banner	Yes
Allow app downloads over insecure networks	Yes
Override URL	Yes
App Configurations	
Send installation request on device registration	Yes
Per App VPN by Label Only	Yes
Icon and Screenshots	Yes

6. Click **Save**.

Related topics

Using the wizard to import iOS apps from the Apple App Store

Changing the iOS or macOS app icon and screenshots

You can edit the app icon and screenshots associated with any app in the App Catalog.

Procedure

1. Obtain the icon or screenshot you want to use.
   See Using the wizard to import iOS apps from the Apple App Store for information on supported formats and dimensions.
2. In the Admin Portal, go to **Apps > App Catalog**.
3. Select **iOS** or **macOS** from the **Platform** list.
4. Click the name of the app you want to work with.
5. Click **Edit**.

   NOTE: When the app data is in View or Edit mode, Core loads the latest managed app schema from the AppConfig repository and displays the latest fields (including any new fields) in the "Managed App Configurations" section in the UI. MobileIron recommends that before saving the changes, you first carefully inspect the updated managed app configuration. Once you select **Proceed** and click **Confirm**, the updated managed app configuration settings are saved and the changes are pushed out to all associated devices.
6. Click **Remove** next to the icon or screenshot you want to remove.
   A new text field displays in the section from which you deleted the screenshot.
7. Next to the text field, click **Browse** to select the graphic file you want to use from the file system.
8. Click **Save**.

Creating or changing a category for iOS and macOS apps

You can create categories for organizing the apps displayed on iOS and macOS devices. The categories appear as dividers in the app lists.

Procedure

1. In the Admin Portal, go to **Apps > Categories**.
2. Select **iOS** or **macOS** from the **Select Platform** list.
3. Click the name of the app you want to add to a category.
   The app details are displayed.

4. Click **Add+**.
   The **Add New Category** dialog box opens.
5. Click **Add New Category**.
6. Enter your configurations.

Field name	Description
Name	Enter a category name (up to 64 characters).
Description	Enter a description (up to 255 characters) for the category.
Category Icon	a. Click the **Replace Icon** button. b. Browse and select an icon that will represent this **Category**.

7. Click **Save**.

Note The Following:

- Categories cannot be deleted.
- To remove a category and apply a different category, clear the check box next to the app to remove the category association.

Notifying users of new iOS and macOS apps or app updates

The following options are available for notifying users of new apps or app updates:

- Informing users of new apps and updates on iOS and macOS devices
- Editing the app distribution push notification template for iOS and macOS
- User notification of newly-published iOS apps
- Copying a direct link to an iOS app

NOTE: These features are not supported when MobileIron Core is configured for MAM-only iOS devices.

Informing users of new apps and updates on iOS and macOS devices

You can send out a message informing iOS and macOS device users about the availability of a new app or an update for an installed app. You can also request device users to convert installed, unmanaged apps to iOS managed apps, without having to uninstall the unmanaged app. You can only convert unmanaged apps to managed on iOS devices running iOS 9 through the most recently released version as supported by MobileIron.

When a user's device checks in, the Update tab displays a badge number indicating the number of in-house and public app updates available for the device user to download. Once the user updates the apps, the badge number will disappear on next device check-in.

Update messages can be sent to a particular label. MobileIron Core will not update the app, rather, Core sends a message to the device requesting the device user to install the app from the Apple App Store. If the Apple App Store indicates an update is required, Core installs the update. If no update is required, Core re-installs the app. For in-house apps, Apps@Work indicates whether an update is required, and Core provides the updated app.

While you can initiate a Send Message for Update request through MDM, the notification directs the device user to the public app in Apps@Work, providing an option for re-installing the app. However, the section for Updates in Apps@Work may not have this public app listed.

Note The Following: :

- Messages sent to iPad only apps will only be sent to iPads.
- If data protection is enabled on managed devices, messages will be sent only to those devices with a passcode.
- When sending a message regarding a hidden app, MobileIron Core shows a prompt asking whether to send the message to device users.
- Device users may not see a given app in the Apple App Store under the following circumstances: the app is Apple License device-based or B2B, or device access to the Apple App Store is disabled.

Before you begin

You must first assign Apple Licensed-macOS apps to a label before using this feature.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Select **iOS** or **macOS** from the **Platform** list.
3. Select the featured app you want to work with.
4. Go to **Actions > Send Installation Request**.
5. Use the following guidelines to select the app installation option:

Item	Description
Send request for new installations	Prompts the device user to install the app, if not already installed. This applies to those devices to which the app has not yet been installed.
Send request for updates	Prompts the device user to update the app. For public iOS apps only. Applies to devices with the app installed, where an update is available. Select one of the following options: • **Use the app install date on device to determine if updates are required**. Core determines whether to install the app based on the last installation date as reported to Core. When installing an app to a device, the device sends to Core a list of all apps installed to the device. Core stores this time stamp reported by the device as the date for the installation of that particular app. When selecting this option, Core only updates the app on devices where the last installed date is lower than the release date of the app. • **Ignore the app install date on device**. Core sends an install command to the device. The app is installed from the Apple App Store, without checking the install date on the device. NOTE: When sending a request to devices to update a public app in the App Catalog, Core is unable to use the app install date on the device to determine whether updates are required, if the app was recently converted from an unmanaged app to an iOS managed app. When sending a request to update a recently converted iOS managed app, select **Ignore the app install date on device**.
Send request for both new installations and updates	Prompts the device user to install or update the app. Applies to all devices, regardless of whether the app has been installed yet.
Send request to convert the app to Managed	Prompts the device user to convert the unmanaged app to an iOS managed app on devices running iOS 9 through the most recently released version as supported by MobileIron. NOTE: This setting does not display for macOS apps.
Use iOS managed app install/update action (iOS 5 and later)	Skip the Apps@Work display and install or update the app. Users will receive installation or update prompts at the next device check-in. Note The Following: • MobileIron Core will silently install or update the app on supervised devices, instead of prompting the supervised device user, even when selecting this option. This is due to changes in Apple's implementation

Item	Description
	of this feature. • This setting does not display for macOS apps.
Send message to all	Select this option to send a message to all devices in all labels.
Select labels to send message	Select this option to send a message to all devices in a selected label.
Select devices to send message	Select this option to search devices by name, model, platform, phone, then select the devices you want to send the message. The **Selected** tab tracks devices selected from the search.

6. To check the content of push notifications prior to sending:
   a. Select the **App Distribution** template from the list.
   b. Click **View Messages**.
7. Click **Apply**.

NOTE: MobileIron Core only sends the message regarding featured apps.

Related topics

Applying an Apple license label to an app

Editing the app distribution push notification template for iOS and macOS

You can customize the template MobileIron Core uses to send app distribution push notifications.

Procedure

1. In the Admin Portal, go to **Settings > Templates > Others**.
2. Click the edit icon for the template you want to edit.
3. The app distribution message displays.

   NOTE: App distribution messages must include the $APPNAME$ variable, which indicates the application name of the app being distributed.
4. Make changes to the displayed message.
5. Click **Save**.

User notification of newly-published iOS apps

When a featured app or an update to an installed app is published to device users, those users receive a notification in Apps@Work. The Updates category shows a number corresponding to the number of updates available. Tapping the Updates category shows the list of apps that are available for update.

MobileIron Core determines the availability of an update by comparing the version number for the installed app to that of the newly-published app. If the user deletes a published app, that app will not become available for reinstalling again until the status of the app is updated during the next sync with MobileIron Core.

- Updates: These include updates for in-house featured apps only. The Updates number shown on the left hand menu in Apps@Work includes app updates, new installations, and unmanaged to iOS managed app conversions. Unmanaged public apps (running iOS 9 through the most recently released version as supported by MobileIron) are also included in the total number of updates indicated in the Updates badge.
- New installations: When an app has a new installation available, a badge indicating the number of installations per app displays. Both in-house and featured public apps can have badges indicating a new installation. Note that you can only send an installation request, **not** an update message, for apps that are publicly available from the Apple App Store.

Note, however, that MobileIron Core does **not**:

- contact the Apple App Store to check for updates to unmanaged apps,
- track the version numbers of public apps
- control which version a user can install.

MobileIron recommends that device users consult the Apple App Store to confirm the availability of new versions of apps. Alternatively, you can use the send message feature to inform device users of a new version of an app. For more information about using the send message feature, see Informing users of new apps and updates on iOS and macOS devices.

Copying a direct link to an iOS app

After adding an app to the App Catalog in MobileIron Core, a direct link to the app is shown in the app details page on Core. You can copy the direct link to the app and include it in an email or notification to device users, allowing users to install the app directly, rather than searching for it in Apps@Work.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Scroll for the app whose direct link you want to copy.
3. Click the name of the app to view its details.
4. Scroll down to the Apps@Work Catalog section.
5. Next to **App URL**, click **Copy Link to Clipboard**.

Working with web applications for iOS and macOS

This section includes the following sub-sections:

Enabling installation of web applications to iOS and macOS devices

You must enable the installation of web applications on managed devices by selecting the relevant options for iOS and macOS in the Apps@Work settings. These options are enabled by default.

Procedure

1. In the Admin Portal, go to **Apps > Apps@Work Settings**.
2. Under **Web Applications**, select the following:
   - **Enable Installation of Web Applications on iOS**.
   - **Enable Installation of Web Applications on macOS**.

   The feature is enabled by default.

Adding a web application to the App Catalog on iOS and macOS devices

Web applications can be launched from Apps@Work and installed to iOS and macOS devices.

Before you begin

Enable the installation of web applications, as described in Enabling installation of web applications to iOS and macOS devices.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Select **Web Application** from the **Platform** list.
3. Click **Add**.
4. Click **Web Application**.
5. Enter the following information:

Item	Description
Application Name	Enter a name, no more than 255 characters, for the web application. This name displays on the device.
App URL	Enter the address or URL for the target of the web clip. The URL must include the prefix http://, https://, or mibrowser://. You can enter up to 255 characters. If you enter the prefix mibrowser://, the URL opens in Web@Work. Web@Work must be installed on the device.

6. Click **Next**.
7. Use the following guidelines to complete this page:

Item	Description
Developer	Enter the name of the developer for this web application.
Description	Enter additional information to describe the app.
Category	Select one or more categories to display this app in a category tab in Apps@Work or add a new category. a. Click **Add New Category** to define new categories. b. Enter a category **Name** (up to 64 characters). c. Enter a **Description** (up to 255 characters). d. In the **Category Icon** section, click the **Replace Icon** button. e. Browse and select an icon that will represent this Category. f. Click **Save**.
Feature this App in the Apps@Work catalog	Select Yes to display the app in the Featured List on the device. The app will also display in all the categories you selected.

8. Click **Next**.
9. Use the following guidelines to complete this page:

Item	Description
App Icon	Click **Browse** to navigate and select a graphic for the web clip.

10. Click **Finish**.

Taking actions on web applications for iOS and macOS

You can take the following actions on a selected web application:

TABLE 12. WEB APPLICATION ACTION ITEMS

Action	Description
Delete	Click **Delete** to delete the web application from MobileIron Core and remove it from Apps@Work.
Apply To Label	Click **Actions > Apply to Label** to select the label to apply. The web application will be available in Apps@Work for the devices associated with the label.
Remove From Label	Click **Actions > Remove From Label** to deselect the labels. The web application will be removed from Apps@Work for the devices associated with the label.

Viewing the number of iOS and macOS devices with web applications installed

You can view the number of devices to which a given web application is installed.

NOTE: This feature is not supported for iOS devices when MobileIron Core is configured for MAM-only iOS devices.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Under **Platform**, select **Web Application**.
   The web applications in the App Catalog are displayed.
   The number in the **Devices** column indicates the number of devices on which the web application is installed.

   NOTE: The number in the **Devices** column will display as 0 if **Enable Installation of Web Applications** is disabled.
3. Click on the number to see a list of devices.

NOTE: Web applications are not tracked in the **Installed Apps**.

Confirming web application installation to iOS and macOS devices

You can confirm a web application has been installed to a given device. You can also confirm that a web application has been installed to a device by checking that the web application icon is shown in Apps@Work.

NOTE: This feature is not supported for iOS devices when MobileIron Core is configured for MAM-only iOS devices.

Procedure

1. In the Admin Portal, go to **Devices & Users > Devices**.
2. Under **Platform**, select **Web Application**.
3. Click the upward arrow (**^**) next to the relevant device.
4. Click the **Configurations** tab.
5. Locate the web clip you sent to the devices.
6. Its status should read **Applied**.

Allow removal of web application from iOS device

You can allow iOS device users to remove web applications from their devices.

Procedure

1. In the Admin Portal, go to **Policies & Configs > Configurations**.
2. To filter by configuration type, in the Configuration Type field, select from the drop-down **WebClip**.
3. Select the preferred WebClip and click the **Edit** button.
   The Modify WebClips Setting dialog box opens. Note that the Removable column displays "false" to indicate device users are not allowed to remove / uninstall Web Clips themselves (default).
4. In the Web Clips field, select the link of the Web Clip name.
   The Edit Web Clip dialog box opens.
5. Select **Removable** and then click **Save**.
   In the Modify Web Clips Setting dialog box, the Removable column now displays "true" for the selected WebClip. Device users will be able to remove / uninstall the WebClip.
6. Click **Save**.

Troubleshooting web application installation for iOS

Enable Installation of Web Applications on iOS is not checked

When you tap the icon, the details page displays the **Launch** button. Tapping the Launch button brings up the web page in a browser.

If the web application points to a mibrowser:// URL, the web page opens in Web@Work. You must have Web@Work installed on your device to view a web page with the mibrowser:// prefix.

Enable Installation of Web Applications on iOS is checked

If the feature is enabled, when you tap the web application in Apps@Work, the details page displays the **Request** button.

NOTE: The details page will display the Launch button if **Enable Installation of Web Applications** is disabled.

Tapping the **Request** button installs the web clip to the device. The status of the button changes to **Installed** after the web application is installed on the device.

The device user can tap on the web clip to access the link. You do not have to go to the Apps@Work to access the link.

Confirming receipt of web clips on iOS devices

Device users can confirm that they have received the relevant web clips from MobileIron Core by going to **Settings > General > Device management > MobileIron > Web Clips** on their iOS devices.

Unmanaged to managed app conversion on iOS devices

You can convert an unmanaged app to an iOS managed app, with little to no device user input. There is no need to uninstall the unmanaged app. This feature applies to apps installed on devices running iOS 9 through the most recently released version as supported by MobileIron.

There are two ways to convert unmanaged apps to iOS managed apps:

- Message: You can convert unmanaged apps to managed by prompting all devices running iOS 9 through the most recently released version as supported by MobileIron. Device users simply respond by tapping **Manage**, which automatically converts the app to managed. On supervised devices, the app is silently converted.
- Apps@Work: Alternatively, you can configure an app as convertible to managed, by posting an update to the app in the **Updates** section of Apps@Work. This means that if the app is installed and unmanaged, the app will be listed in the **Updates** section of Apps@Work. Device users can find the app under **Updates** and tap **Update** to convert the app to managed. The version of the app will stay the same.

Note The Following:

- Apps cannot be converted back to unmanaged. If you uninstall an iOS Managed app and re-install it from the Apple App Store, it still displays as Managed. If an app needs to be converted back to unmanaged, you need to remove the label associated with it.
- Apps that are hidden from view in Apps@Work on the device can still be converted to managed apps.
- When converting Apple-Licensed apps from unmanaged to managed, Apple Licenses are not consumed.
- If you delete a managed app from the device and install an unmanaged app, Core will convert the unmanaged app to managed automatically without notifying or prompting the device user.
- This feature is not supported when MobileIron Core is configured for MAM-only iOS devices.

This section includes the following topics:

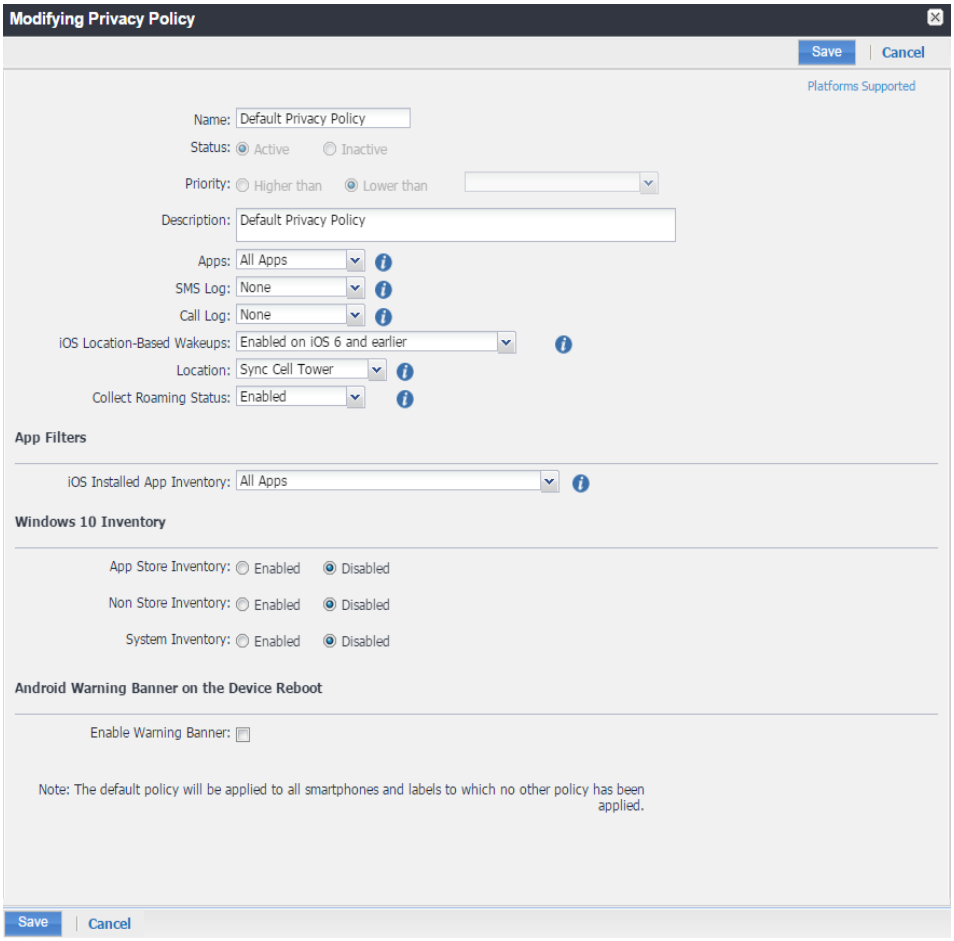
- Viewing the managed status of an iOS app
- Viewing the status of iOS managed apps for a given device
- User prompts to convert an app from unmanaged to managed on an iOS device
- Converting an app to managed on an unsupervised iOS device

Enabling app inventory synchronization in the privacy policy for iOS

Make sure to enable app inventory synchronization in your privacy policy.

Procedure

1. Go to **Policies & Configs > Policies**.
2. Click the privacy policy that is currently enabled.
3. In the Policy Details pane, click **Edit**. The Modify Privacy Policy dialog box opens.



4. Select **Apps > All Apps**, if it has not been selected already.

   NOTE: The administrator should update the Default Privacy Policy with highest priority and set the "iOS Installed App Inventory" field to "All Apps". This ensures unmanaged app are

reported to the Core and the Core can send MDM commands to convert unmanaged app to managed app. This applies if the app has the option "Enforce conversion from unmanaged to managed app (iOS 9 or later)" enabled and the unmanaged version of the same app is reported to the Core.

5. Click **Save**.
6. Apply the policy to the relevant labels.

Converting an unmanaged app to a managed app by prompting iOS device users

You can convert an unmanaged app to an iOS managed app by prompting users to accept a request to convert the app on their devices.

Procedure

1. Go to **Apps > App Catalog**.
2. From the Platform list, select **iOS**.
3. Click **Search** to display a list of iOS apps.
4. Select the check box next to the name of the iOS app you want to convert to managed.
5. Make sure you have applied the app to the relevant labels.
   a. From the list of apps, select the app you edited.
   b. Go to **Actions > Apply to Labels**.
   c. Select the labels you want to apply.
   d. Click **Apply**.
6. Go to **Actions > Send Message**.
7. In the **Send App Installation Request** dialog box, select **Send request to convert the app to Managed**.
8. Click **Apply**.
9. The next steps depend on the type of iOS device receiving the message, as described in the following table:

	iOS 5-8.4 devices	Unsupervised devices running iOS 9 through the most recently released version as supported by MobileIron	Supervised devices running iOS 9 through the most recently released version as supported by MobileIron
Prompt?	No prompt appears.	A prompt displays on the device, indicating that the app will be converted from unmanaged to managed.	No prompt appears.
Next steps?	None. This feature is not supported on devices running iOS 5-8.4.	The device user must tap **Manage**. The app is then converted to managed. If the device user has not yet installed the app, the app will be installed as managed. If the device user taps **Cancel**, the app is not converted to managed.	No user action required. The app is silently converted from unmanaged to managed.

Enabling device users to convert iOS apps from unmanaged to managed in Apps@Work

You can allow device users to convert an unmanaged app to an iOS managed app in Apps@Work.

Procedure

1. Go to **Apps > App Catalog**.
2. From the Platform list, select **iOS**.
3. Click **Search**.
   A list of iOS apps displays.
4. Click the name of the iOS app you want to convert to managed.
   The app details are displayed.
5. Click **Edit**.

   NOTE: For iOS apps, when the app data is in View or Edit mode, Core loads the latest managed app schema from the AppConfig repository and displays the latest fields (including any new fields) in the "Managed App Configurations" section in the UI. MobileIron recommends that before saving the changes, you first carefully inspect the updated

managed app configuration. Once you select **Proceed** and click **Confirm**, the updated managed app configuration settings are saved and the changes are pushed out to all associated devices.

6. Select **Allow conversion of apps from unmanaged to managed in Apps@Work (iOS 9 or later)**.
7. Click **Save**.
8. Click **Back to List** to return to the list of apps.
9. Make sure you have applied the app to the relevant labels.
   a. From the list of apps, select the app you edited.
   b. Go to **Actions > Apply to Labels**.
   c. Select the labels you want to apply.
   d. Click **Apply**.
   The app will be listed in the Updates section of Apps@Work.
10. Device users must tap **Update** next to the app to convert it to managed.

Viewing the managed status of an iOS app

You can view whether a given app is managed or unmanaged on a device by clicking the number in the Devices Installed column for that app.

NOTE: Currently, macOS apps cannot be managed apps. Therefore, any macOS app displays as unmanaged.

Procedure

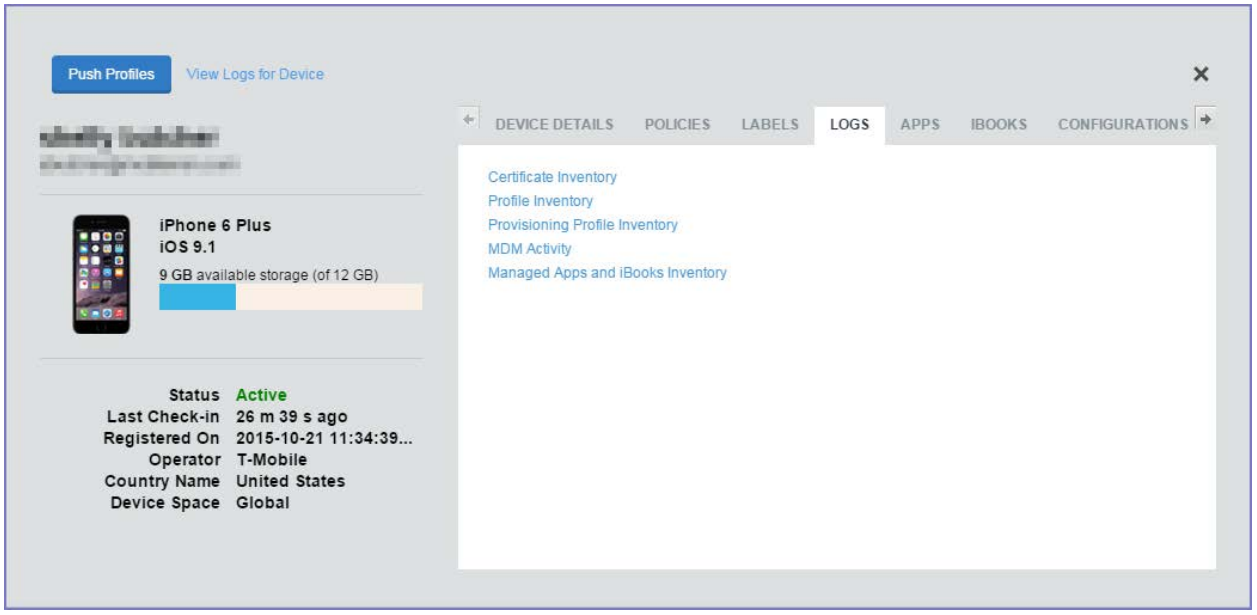
1. Go to **Apps > App Catalog**.
2. From the Platform list, select **iOS**.
3. Click **Search**. A list of apps displays.
4. Find the name of the app whose managed status you want to view.
5. In the **Devices Installed** column, click the number next to the app whose managed status you want to view.
6. The Device Details window displays. Locate the **Managed** column.
7. The **Managed** column indicates whether the app is managed on the devices listed.
8. Click **Close**.

Viewing the status of iOS managed apps for a given device

You can view the managed status of apps installed on a given device by drilling down to the device details on the **Devices** page in MobileIron Core.

Procedure

1. Go to **Devices & Users > Devices**.
2. Click the device to view device details.



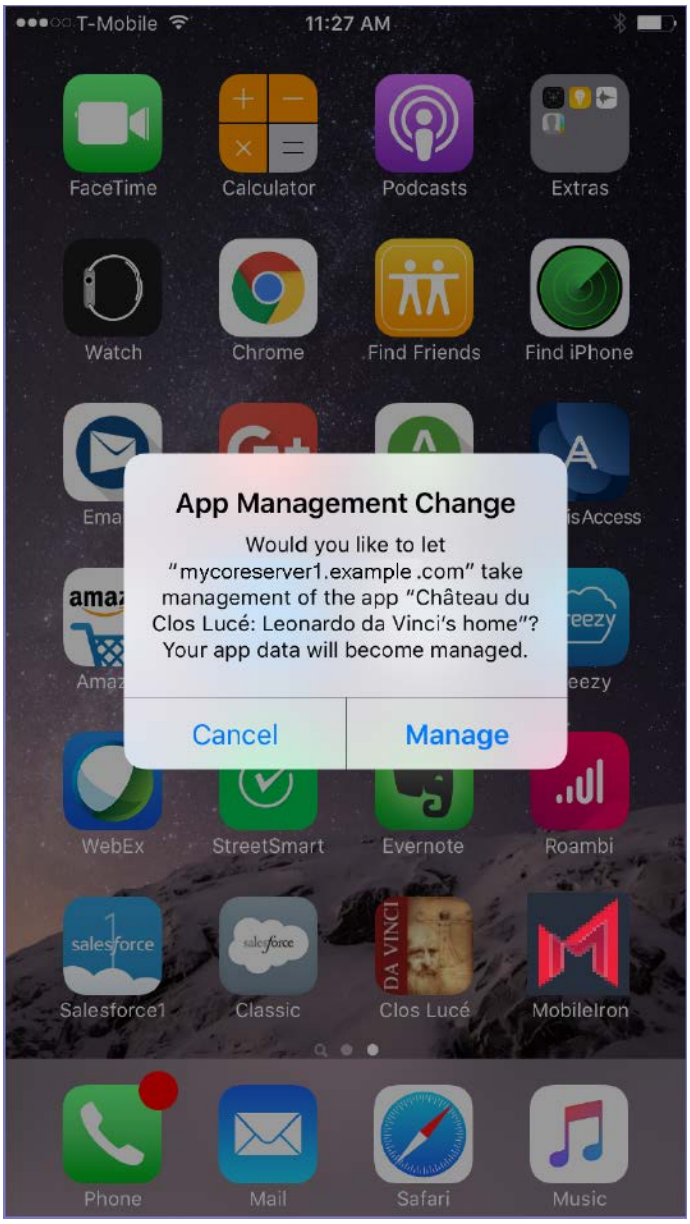
3. Click the **Managed Apps and iBooks Inventory**. A list of apps and iBooks installed on the device displays.
4. Locate the app whose managed status you want to view.
5. In the **Status** column, check the status of the app.
6. Possible status values are listed in the following table:

Status	Notes	Status	Notes
`Managed`	Indicates iOS managed app status (managed by MDM), or successful conversion from unmanaged to managed.	`ManagementRejected`	Indicates the device user tapped Cancel when prompted to convert the app to managed. Valid status for devices running iOS 9 through the most recently released version as supported by MobileIron.
`ManagedButUninstalled`	The app was managed, but the device user deleted it. If the device user installs the app again, the app will be managed.	`PromptingForManagement`	Indicates the device user is being prompted to convert the app and has not yet tapped **Manage** or **Cancel**. Valid status for devices running iOS 9 through the most recently released version as supported by MobileIron.
`UpdateRejected`	A transient state indicating the device user canceled the managed app update.	`Installing`	
`UserInstalledApp`		`UserRejected`	Indicates the device user canceled a managed app installation.
`Prompting`	Prompting for installation.	`PromptingForUpdate`	Prompting for app update.
`PromptingForLogin`	Prompting for App Store password.		

Status	Notes	Status	Notes
Failed		AppAlreadyQueued	
AppAlreadyInstalled		AppStoreDisabled	
NotSupported		NeedsRedemption	
CouldNotVerifyAppID		Redeeming	
Queued		Unknown	
NotAnApp		ValidatingPurchase	
PromptingForUpdate Login		Updating	
ValidatingUpdate		PurchaseMethodNot Supported	

User prompts to convert an app from unmanaged to managed on an iOS device

If a user has any unmanaged apps installed on the device, the user will be prompted to convert the unmanaged app to a managed app by selecting **Manage**. The unmanaged app will be converted to a managed app.



Supervised devices will silently convert the unmanaged app to an iOS managed app. This applies to devices running iOS 9 through the most recently released version as supported by MobileIron.

NOTE: This applies if the app has the option "Enforce conversion from unmanaged tomanaged app (iOS 9 or later)" enabled and the unmanaged version of the same app is reported to the Core.

On supervised devices, unmanaged apps are also silently converted to managed, such that the device user will have no indication that the app successfully converted to managed. You can verify that the app was converted to managed by checking the status of the app, as described in Viewing the status of iOS managed apps for a given device.

Converting an app to managed on an unsupervised iOS device

On unsupervised devices, the Updates section in Apps@Work indicates a new update for the app that is to be converted from unmanaged to managed.

To convert an app to an iOS managed app, instruct unsupervised device users to take the actions described in the following procedure.

Procedure

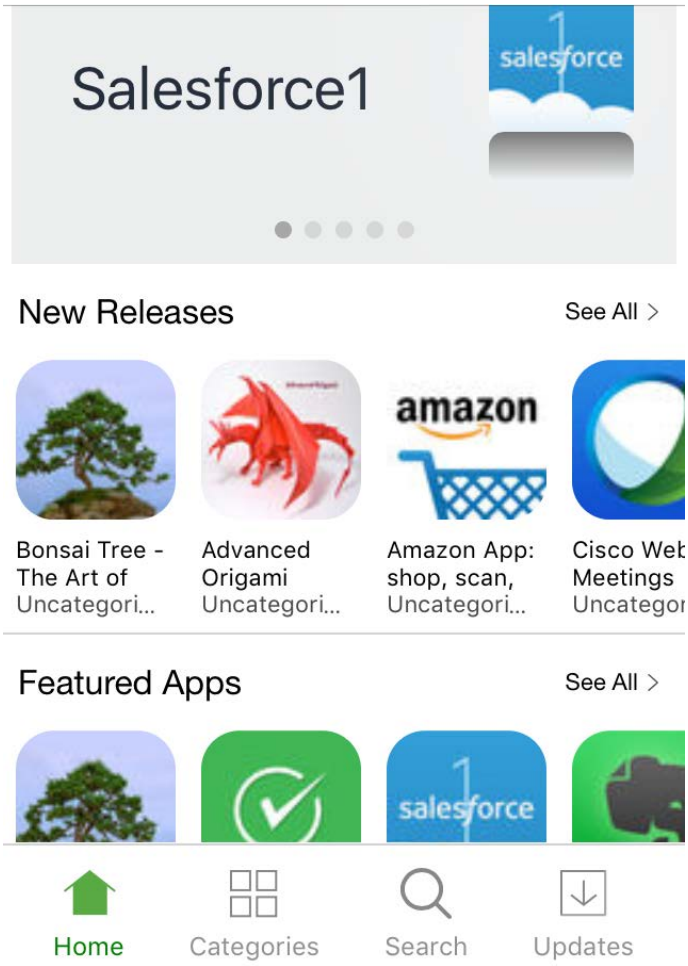
1. On the user device, tap the Apps@Work web clip.
2. Tap the category for the app, such as **Updates** or **Featured**.
   The app displays with an indication that it requires an update.
3. Tap **Update** next to the name of the app in Apps@Work.
   The app details page displays.
4. Tap **Update** on the app details page.
   A prompt displays, indicating that the app will be converted to managed.
5. Tap **Manage** to convert the unmanaged app to managed.

Apps@Work on the iOS or macOS device

Apps@Work as it appears on the device is meant to resemble the Apple App Store. The Apps@Work home screen includes the following components:

- Apps@Work title: By default, the Apps@Work home screen is titled Apps@Work, and bears the default Apps@Work icon. You can customize the title and icon, however, as described in Apps@Work branding.
- Featured banner: The featured banner shows the five latest apps that have been configured to display in the featured banner, where the featured app rotates every five seconds. When tapping an app in the featured banner, the app details are displayed. For details, see Featured Banner.
- New Releases: The New Releases section displays apps imported into the App Catalog on MobileIron Core since the last device synchronization.
- Featured Apps: The Featured Apps section shows App Catalog apps that have been configured as featured apps, to be displayed in the Featured Apps category.
  - Tap an app to view its details, tap **Install** to install the app.
  - Tap **Re-install** to install an app again.
  - Tap **Update** to request an update of the app. The Update option is only shown if a newer version of the installed app is available in the App Catalog.
  - Tap **View** to examine the details of a public app on the Apple App Store.
  - Apps being installed have a **Pending** status. Refresh the screen by pulling it down.
    For more details about featured apps, see Feature this App in the Apps@Work Catalog.

- Popular Apps: The Popular Apps sections shows up to 25 App Catalog apps with the greatest number of installations in descending order over the last 30, 60, or 90 days. Device users only see those popular apps applied to labels to which they belong, regardless of whether they have installed these apps. Popular apps not available for download to a given device will not be shown. Popular apps are updated once per hour. Uninstalled apps are not counted or shown.
- For more information about popular apps, see Configuring popular apps for display in Apps@Work (Android, iOS, macOS).
- Categories: The Categories section shows all the categories you defined for apps in the App Catalog in MobileIron Core. For more information about categories, see Creating or changing a category for iOS and macOS apps.
- Search: You can search for apps by name or description. The last item searched will be remembered.
- App details: When tapping an app, Apps@Work shows an app details page, which includes the following information:
  - App name and icon
  - Description
  - Version
  - Date published
  - Developer
  - Install status
  - Free/Prepaid/Price
  - Size
  - Compatibility
  - Ratings
  - Reviews
  - Screen shots



A series of icons at the bottom of the page allow the device user to more easily navigate Apps@Work. The icons include:

- Home: Tap to return to the Apps@Work home screen.
- Categories: Tap **See All** to browse apps by category, such as Sales, Marketing, Engineering. Tap a category to view a list of all apps in that category. On each category page, tap the **Install All Apps** button (an underlined downward arrow) to send an installation request to MobileIron Core and the Apple App Store. Tapping Install All Apps will also install Apple Licensed apps.

  NOTE: Unsupervised devices will request permission first before installing all apps. On supervised devices, MobileIron Core installs all apps without requesting device user permission.
- Search: Tap to search for apps by name.
- Updates: Tap to view a list of apps that have updates available. Tap an individual app to update it.
- Tap the **Update All button** (an underlined downward arrow) to update all apps. An update request is sent

for all apps. Apps being updated have a **Pending** status. Refresh the screen by pulling it down.

NOTE: Update status is not supported for MAM-only iOS devices.

Using Apple licenses

This section addresses using Apple licenses.

Note The Following:

- Using Apple Licenses is not supported with MAM-only iOS devices.
- Apple has renamed "VPP" to "Apps and Books" in the Apple portal.

Apple license management with MobileIron Core

Apple licenses allows participating organizations to purchase iOS and macOS apps and distribute these apps to their users and to multiple devices.

Before using Apple licenses

- For information about which port to open on MobileIron Core for access to https://vpp.itunes.apple.com, see the “Changing Firewall Rules” section in the *On-Premise Installation Guide for MobileIron Core and*

Enterprise Connector.

- If the Apple license servers are overloaded with requests, it may return a 503 Service Unavailable status to clients. This response may include a Retry-After header, which indicates the time period clients must wait before making additional requests. If Apple returns a Retry-After header for a specific Apple license account, MobileIron Core will block any actions for that Apple license account for the time period specified by the Retry-After header and will then retry appropriately.
- For devices running iOS 9 through the most recently released version as supported by MobileIron, apps can be purchased through Apple licenses in one country and assigned to devices in other countries, as long as the app is available from the Apple App Store in the countries where it is used.
- MobileIron supports B2B (Business to Business) Apple licenses globally.

Apple license features

The Apple license management feature provides the following benefits:

- **Reclaim Apple licenses**. Apple licenses are reclaimed in the following instances:
  - A user is removed from a group applied to an Apple License Label.
  - A device is retired.
  - The device user removes the app from the device.
  - The administrator manually reclaims individual or all licenses for a given app or account.
- **Sync Apple License usage with Apple**. The licenses associated with your Apple license account are not specific to your MobileIron Core; they are specific to the account in the Apple portal. Core syncs with the Apple servers once every 30 minutes to reconcile each Apple license account. The information reconciled includes:
  - Number of licenses purchased
  - Number of licenses used
  - Inventory of purchased apps
  - The user or the device to which the license is applied

    This gives the organization up-to-date visibility into app and license inventory for each Apple license account.
- **Manage multiple Apple license accounts**. You can manage multiple Apple license accounts on MobileIron Core. This allows you to support multiple buying centers that can purchase and distribute apps. For the same app, each license pool is segmented and managed separately.

Apple license use

Consider the following:

- **Free applications consume an Apple license**. MobileIron Core requests an Apple license. Because the apps are free, the MobileIron Core administrator can login to Apple's license management website and add more licenses without cost. This is the recommended best practice from Apple.
- **Converting an app from UNMANAGED to MANAGED consumes an Apple license**.

- **Device-based Apple-licensed apps cannot be updated via the Apple App Store.** Admins should send devices an install/update message for Device-based licensed apps when desired, as described in Notifying users of new iOS and macOS apps or app updates.
- **Apple licensed device-based apps are not backed up.** Due to Apple's design of device-based licensed apps, when the user backs up a device, the backup does not include Apple licensed device-based apps. Therefore, if the user resets and restores the device from the backup, licensed device-based apps will not be restored to the device and will need to be re-installed by the user.
- **MobileIron Core does not assign a device-based license if the app already has user-based license that was assigned by the current instance**.

Main steps for setting up Apple licenses

This section addresses the main steps involved in setting up Apple Licenses:

1. Linking MobileIron Core to an Apple licensed account
2. Importing licensed apps from an Apple licensed account
3. Importing additional apps from the App Catalog
4. Applying device-based licensing to an app
5. Applying a user-based license
6. Applying an Apple license label to an app
7. Removing an Apple license label from an app

Linking MobileIron Core to an Apple licensed account

To use Apple licenses with MobileIron Core, you must log into your Apple license account to retrieve a managed distribution token. You will then use the token to link MobileIron Core to your Apple license account. Licenses are automatically available after you link your Apple license account to Core. Licenses are updated each time Core syncs with Apple's servers.

Before you begin

Be sure to open the relevant HTTPS port for Apple license support. For information about which port to open on MobileIron Core for access to https://vpp.itunes.apple.com, see the "Changing Firewall Rules" section in the *On-Premise Installation Guide for MobileIron Core and Enterprise Connector.*

Procedure

Some of the instructions in this section describe how to perform a procedure on a third-party website whose topology may change, affecting the navigation path described here.

1. In the Admin portal, select **Apps > Apple Licenses**.
2. Click **+Add Server Token**.
   The Add Server Token dialog box opens.
3. Click on the Apple Business Manager or Apple School Manager link. A browser page opens to the correct login page on the Apple portal.
4. Click Settings > Apps and Books.
5. Under My Server Tokens, look for the location you want to use and click the Download link to get the token.
6. Open the token as a .txt file and copy the token string.
7. Return to Core > Add Server Token dialog box.
8. Paste the token into the Server Token field (required). The License expiration date displays below Server Token field.

   NOTE: If another user logged into the Apple site and has downloaded the same location's token and added to your MobileIron Core, your attempt to add the Apple license will fail. An error message indicates that an Apple license with the same license already exists on MobileIron Core.
9. In the **Account Name** field, enter a name for the account.
10. Enter an optional **Description** for your use.
11. MobileIron recommends that you leave the **VPP Account is shared with one of more MobileIron Cores** field de-selected. This indicates to Core to only ask Apple for the VPP license information from newly-assigned licenses. For full license sync, see Importing licensed apps from an Apple licensed account.
12. Click **Save**.

Importing licensed apps from an Apple licensed account

You can import Apple licensed apps into MobileIron Core on the App Licenses page. This is done by refreshing Core's connection to your account, updating the available Apple licenses, and then selecting the apps you want to import into Core. Core refreshes available Apple licenses several times per day. However, MobileIron recommends refreshing your licenses before importing Apple licensed-apps.

NOTE: Core supports the importation and distribution of macOS Apple licensed apps.

Before you begin

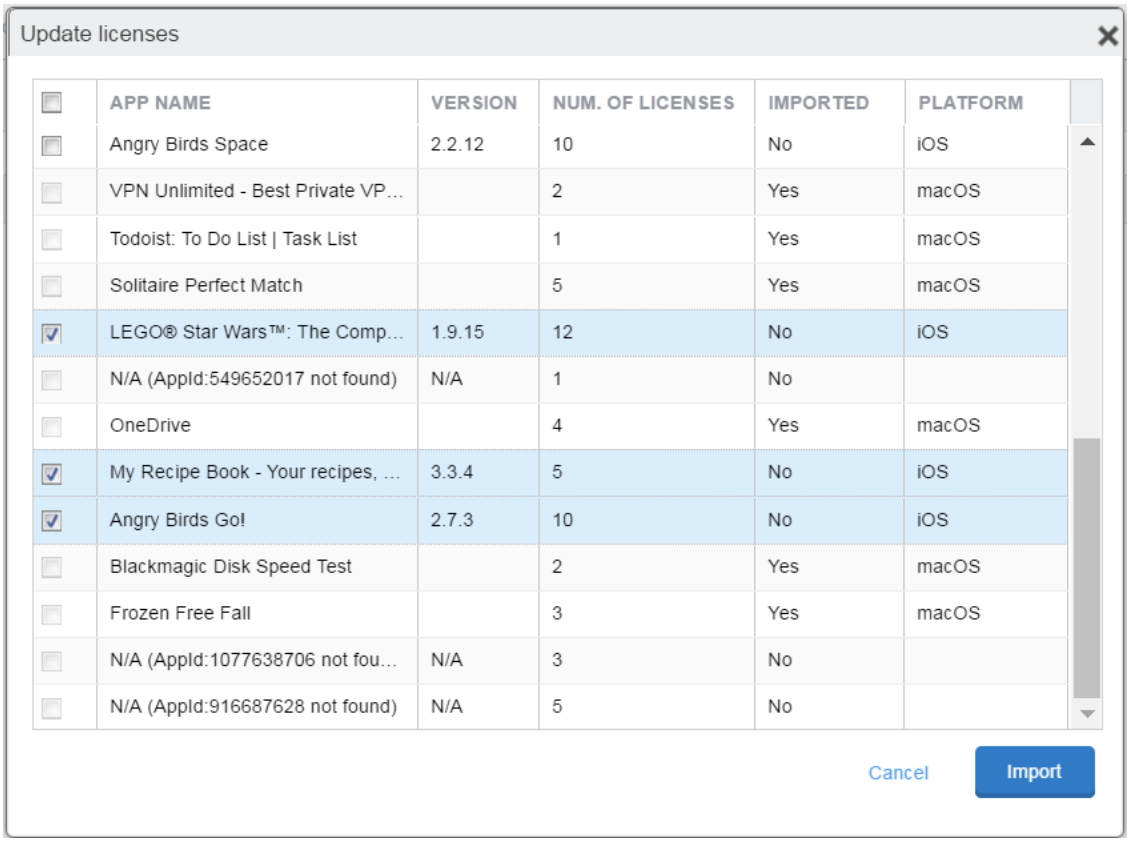
Before proceeding, you must have done the steps in Linking MobileIron Core to an Apple licensed account

Procedure

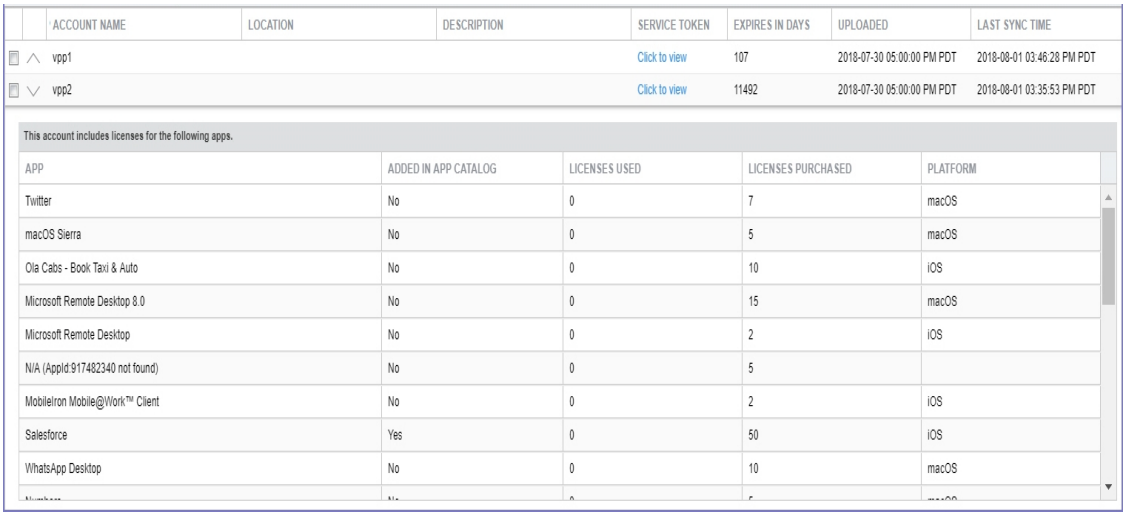
1. In the Admin portal, select **Apps > Apple Licenses**.
   The following information displays for each Apple licensed account:

Field	Description
Account Name	The account name entered when adding the Apple license account.
Location	The location name as defined in the Apple Business Manager. The location is empty if the account does not use Apple Business Manager or Apple School Manager.
Description	Additional information that describes this account.
Service Token	The credential used to link the Apple license account to MobileIron Core. This token has location information when created using Apple Business Manager.
Expires In Days	Number of days before the service token expires. Before the service token expires, you must download a new service token from Apple's license management portal.
Uploaded	Date when the service token was last added to MobileIron Core.
Details Sync Time	The time stamp, including the date, time, and time zone, for the last synchronization between Core and the Apple servers.
Count Sync Time	The time stamp, including the date, time, and time zone, for the last time the license count was synced. This is a quick query to get the number of used / available licenses for each app.

2. Select the account.
3. Click **Actions > Update licenses**.
   The Update licenses dialog box opens, displaying a list of licensed apps available from the Apple License account.
4. Select the apps you want to import into Core as Apple licensed apps.



5. Click **Import**. The selected Apple-licensed apps are imported.
6. Click the carat (^) next to the account name to view the Apple-licensed apps you imported.



The table columns are described in the following table:

Item	Description
App	Name of the app purchased with the Apple license account.
Added in App Catalog	Indicates whether you imported the app into Core for distribution. When you import an app, it is also displayed in the App Distribution page.
Licenses Used	Number of licenses redeemed for the app. This includes the totals of combined user licenses and device licenses. It also includes licenses that were redeemed by other MobileIron Core instances.
Licenses Purchased	Number of licenses purchased for the app. This includes the totals of combined user licenses and device licenses.
Platform	Indicates which platform the license is applicable to.

Related topics

Viewing Apple Licenses in the Audit Logs

Importing additional apps from the App Catalog

You can import Apple licensed apps from the App Catalog. For more information about importing these apps, see Using the wizard to import iOS apps from the Apple App Store on page 70 and Manually importing iOS apps from the Apple App Store on page 68.

For an app already listed in the App Catalog, the **Licenses Purchased / Used** column now displays the license information.

NOTE: When you import an app from the App Catalog that uses Apple licenses, deselect the **This App Store App is Free** option. This allows the device user to successfully download the app using an Apple license.

Applying device-based licensing to an app

After you have linked MobileIron Core to your Apple license account, you can apply device-based licensing to your iOS and macOS licensed apps, so that users do not need to enter an Apple ID when installing Apple licensed apps. This applies to macOS devices and iOS devices running iOS 9.0 through the most recently released version as supported by MobileIron.

Procedure

1. In the Admin portal, go to **Apps > App Catalog**.
   You can see the Licenses Used and Licenses Purchased for each app. The totals listed includes the combined number of user licenses and device licenses.
2. Use the check box to select the app to which to apply device-based licensing.
3. From the **Actions** menu, select **Manage Licenses**.
   The License Summary page opens, listing the registered Apple licensed accounts for the specified app. You can see the Available Licenses and Used Licenses information for the selected app.
4. Clicking the desired account opens the Account Detail page.
5. Expand **License Type**, select **Device-based License**, and then click **Save**.

   NOTE: If you have User-based License already applied to existing users/devices, MobileIron Core will not remove those licenses.
6. Expand **License Label Management** and select the desired labels so that target devices that request the selected app receive device-based licenses.

Applying a user-based license

Follow the instructions in Applying device-based licensing to an app on page 130, but select **User-based License** instead.

NOTE: While Apple supports user-based licensing for macOS apps, currently, there is an Apple issue with the installation of user-based licensed apps through MDM. As a result, MobileIron does not recommend applying user-based licenses to macOS.

For Apple-licensed apps that will be used by both User Enrolled devices and Device Enrolled devices, the license type should be set to Device-based licenses. Devices enrolled with Device Enrollment (DEP) where the Apple-licensed app is set to user based licenses will show the app as "Free" instead of "Prepaid" and the app will fail to install. User enrolled devices will always have the Apple-licensed app assigned as a user based license regardless of this setting.

Applying an Apple license label to an app

You must apply an Apple license label to your apps for licenses to be applied. Devices that are only applied to non-licened labels cannot redeem an Apple license. These devices are redirected to the Apple App Store to purchase the app.

Procedure

1. Go to **Apps > App Catalog**.
2. Select **iOS** or **macOS** from the **Platform** list.
3. Select the check box next to the desired app and then click **Actions > Manage Licenses**.
4. Select the license account to manage and then select **Apply To Labels** .
5. In the **Apply To Labels** dialog box, select the label(s) and then select **Apply**.

Removing an Apple license label from an app

Remove an Apple license label from an app if you want to free up an Apple license for that app.

Procedure

1. Go to **Apps > App Catalog**.
2. Select **iOS** or **macOS** from the **Platform** list.
3. Select the Apple license app you want to remove from the Apple license label.
4. Click **Actions > Manage Licenses.**
5. In the License Summary page, click the link of the label you want to remove.
6. In the License Label Management section, select the label and then click **Remove**.

Revoking licenses

You can revoke all licenses for a given app. Alternatively, you can revoke a license for a given app from a specific device.

- Revoking all licenses for an Apple licensed app
- Revoking a license for an Apple licensed app from a specific device

Revoking all licenses for an Apple licensed app

You can revoke all licenses for a given Apple licensed app.

Procedure

1. Log into In MobileIron Core.
2. Select **Apps > App Catalog**.
3. Use the check box to select the app to apply device-based licensing to.
4. Select **Actions > Manage Licenses**.
   The License Summary page opens, listing the registered Apple licensed accounts for the specified app.

5. Click the desired account. The License Detail page displays. Note that the number of Available Licenses and Used Licenses is listed at the top of the page.
6. Click the **Revoke All Licenses** button.
7. Click **Confirm**.

Revoking a license for an Apple licensed app from a specific device

You can revoke a license for a given Apple licensed app from a specific device.

Procedure

1. Log into In MobileIron Core.
2. Select **Apps > App Catalog**.
3. Use the check box to select the app that contains the device-based licensing.
4. Select **Actions** > **Manage Licenses**.
   The License Summary page opens, listing the registered Apple licensed accounts for the specified app.
5. Click the desired account. The License Detail page displays. Note that the number of Available Licenses and Used Licenses is listed at the top of the page.
6. Expand the **Licenses Distribution Details** section.
7. From the list of devices displayed, select the device whose license you want to revoke.
8. Click **Revoke**.

Exporting Apple license app distribution details to a CSV file

You can export to a CSV file the license distribution details of a given Apple licensed app.

Procedure

1. Log into In MobileIron Core.
2. Select **Apps > App Catalog**.
3. Use the check box to select the app that contains the Apple license.
4. Select **Actions** > **Manage Licenses**.
   The License Summary page opens, listing the registered Apple licensed accounts for the specified app.
5. Click the desired account. The License Information page displays.
6. Expand the **Licenses Distribution Details** section.
7. From the list of devices displayed, select the device whose license you want to revoke.
8. Click **Export to CSV**.

Managing your Apple license accounts

Managing your Apple license accounts involves the following tasks:

- Viewing Apple license accounts
- Viewing Apple license account information
- Viewing Apple license app information
- Viewing Apple Licenses in the Audit Logs
- Updating or deleting an Apple license account
- Full sync of all licenses

Viewing Apple license accounts

You can add more than one Apple license account to the MobileIron Core Admin Portal. You can view and manage these accounts on the License Summary page. You must have added at least one Apple license account to the MobileIron Core Admin Portal to view the License Summary page.

Procedure

1. Go to **Apps > Apple Licenses**.
   A list of Apple license accounts associated with MobileIron Core displays.
2. The following information is shown for each Apple license account:

Item	Description
ABM Account name	The name you assigned to the Apple license account when you added it to MobileIron Core.
Space	Indicates the type of device space, for example, Global.
Location	The location name as defined in the Apple Business Manager. The location is empty if the account does not use Apple Business Manager.
Description	A description of the Apple license account.
Server Token	The credential used to link the Apple license account to MobileIron Core. You can view the server token for the account by clicking the **Click to view** link. This token received location information when it was created using Apple Business Manager.
Expires In Days	Number of days before the server token expires. Before the server token expires, you must download a new server token from the Apple license management portal.

Item	Description
Uploaded	Date when the server token was last added to MobileIron Core.
Details Sync Time	The time stamp, including the date, time, and time zone, for the last synchronization between Core and the Apple servers.
Count Sync Time	The time stamp, including the date, time, and time zone, for the last time the license count was synced. This is a quick query to get the number of used / available licenses for each app.

Viewing Apple license account information

You can view the details of a given Apple license account.

Procedure

1. Go to **Apps > Apple Licenses**.
2. For the desired account, click the inverted V icon.
   The following information displays:
   - Apps purchased with the Apple license account
   - Whether the app was imported to the App Catalog in the MobileIron Core Admin Portal
   - Licenses Used and Licenses Purchased (includes the totals of combined user licenses and device licenses)
   - Whether the target platform for the app is iOS or macOS.

Viewing Apple license app information

You can view details about a given Apple license app.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Select the app and then select **Actions > Manage Licenses**.
   The License Summary page displays the following information:

Item	Description
Name	The name of the Apple license account that the app is associated to. You can click the Name link to change the license type. You can also apply or remove the selected app to / from a particular label. A license can be revoked or exported to CSV. If the Apple license account is expired, it will be highlighted.
License Type	The type of Apple license used for this app: User-based or Device-based.
Available Licenses	The number of Apple licenses available for this app. The number listed represents the total of combined user licenses and device licenses.
Used Licenses	The number of device-based and user-based licenses consumed for this app.
Location	The location name, as defined in the Apple Business Manager. If the account does not use Apple Business Manager, this field will be empty.
Applied Labels	Lists the labels to that are applied to the selected app.

3. To view the devices a particular app is installed on:
   a. In the **App Catalog** page, locate the app in the list of apps.
   b. Click the number in the **Devices Installed** column.
   A window opens, listing the devices the app is installed on and the associated Apple license account.

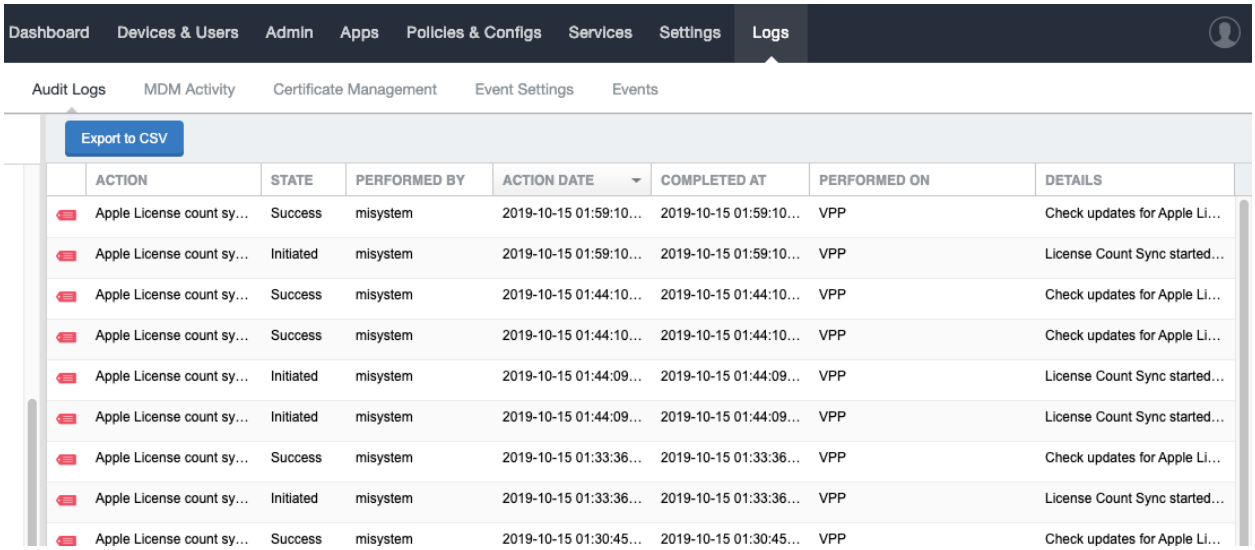
Viewing Apple Licenses in the Audit Logs

You can view the Apple licenses within the Audit Logs.

Procedure

1. Go to **Logs > Audit Logs**.
2. In the left column, find **VPP** and select the following four fields:
   - Apple License Count Sync Completed
   - Apple License Count Sync Started
   - Apple License Sync Completed
   - Apple License Count Sync Started

3. Run the **Search**. The search results display in the right pane.



Related topics

If you want to change specific settings for your Apple license app, see the following topics:

- Changing iOS and macOS app information
- Changing the iOS or macOS app icon and screenshots
- Creating or changing a category for iOS and macOS apps

Updating or deleting an Apple license account

You can update or delete an Apple license account from Core.

Procedure

1. Go to **Apps > App Licenses**.
2. Select a license and then click **Actions**.
   You can take the following actions on an Apple license account:

Action	Description
Update Server Token	Select to update the Apple license account into MobileIron Core. Before you continue, you will need the server token from Apple Business Manager or Apple School Manager.
Update Licenses	Select to edit the Apple license account information or to import apps.
Sync All Licenses	If this location is shared across other MobileIron Cores or any other UEM servers, select to run a full sync of all licenses. MobileIron recommends leaving this check box de-selected. WARNING: Running a fully sync of all licenses slows down Core. MobileIron suggests you do the full license sync on weekends or outside of regular office hours. Use this feature only for remediation of defective license information; do not do a full sync on a regular basis. See Full sync of all licenses.
Delete Server Token	Select to delete the Apple license account from MobileIron Core. When you delete an Apple license account: • All licenses for the apps purchased through the Apple license account are reclaimed. • Users have a grace period of up to 30 days to purchase the apps.

Full sync of all licenses

In previous versions, Core automatically did a full sync of all licenses. From version 10.6.0.0 through the latest version as supported by MobileIron, administrators will need to manually run a full sync of all licenses. This feature is useful if your license location is shared across other MobileIron Cores or any other UEM servers. A full sync of all licenses gets full details of all new and updated licenses in the system, including across multiple Cores.

WARNING: Running a fully sync of all licenses slows down Core. MobileIron suggests you do the full license sync on weekends or outside of regular office hours. Use this feature only for remediation of defective license information; do not do a full sync on a regular basis.

Procedure

1. In the Admin portal, select **Apps > Apple Licenses**.
2. Click **Actions > Sync All Licenses**. The Update Server Token dialog box opens.

3. Select the **VPP Account is shared with one of more MobileIron Cores** check box.
4. Review the information in the Update Server Token dialog box. When finished, click **Save**.
   Depending upon how many licenses you have and the number of users and devices the VPP licenses are assigned to, the full sync may take minutes to hours to conduct.

Turning user-paid apps into managed apps

If a user-paid app has been configured on MobileIron Core as convertible from an unmanaged app to an iOS managed app, on devices running iOS 9 through the most recently released version as supported by MobileIron, then:

- the iOS managed version of the app is installed on the user's device, and
- an Apple license is consumed for that app.

If a user's device is running iOS 8.0 through iOS 8.4, and they have installed an app directly from the Apple App Store, then:

- the user must uninstall that app, and
- install the iOS managed version of the same app from Apps@Work.

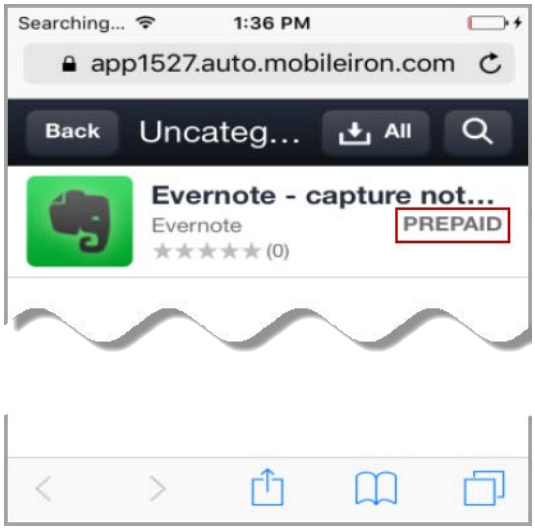
For example, if a new employee has already installed a paid app that your organization ordinarily manages through the Apple licensing program, then the employee must delete the app and reinstall it from the Prepaid tab in Apps@Work. Otherwise, the app will remain unmanaged.

Installing an Apple licensed app with a prepaid license to an iOS or macOS device

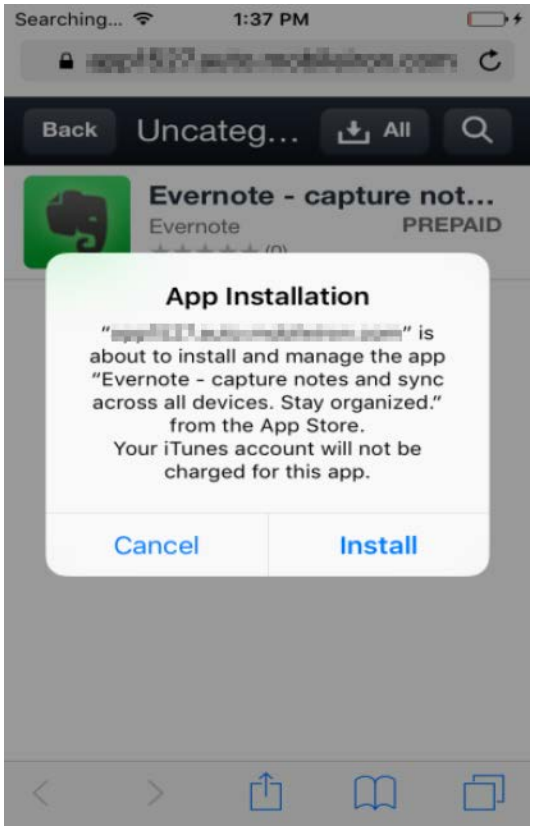
When tapping or clicking an Apple licensed app in Apps@Work, device users are prompted to enroll in their company's Apple licensing program. This applies only to Apple licensed apps with a user-based license.

Procedure

1. Tap or click the prepaid app.
2. Follow the prompts to enroll in your company's Apple license purchasing program.
3. After you successfully enroll in the program, tap or click the app in Apps@Work.
   The app details page now shows the PREPAID status.
4. Tap or click **Request** and follow the prompts to install the app.
   When launching Apps@Work and selecting a device-based licensed app, the app displays as **PREPAID:**



After requesting the iOS app, Apps@Work displays a message indicating that the device user's iTunes account will not be charged for the app. The iOS app is then installed without prompting the user for an iTunes ID.



On supervised iOS devices and macOS devices, the Apple licensed apps are installed silently.

Managing mobile apps for Android

This section addresses how you can manage apps for Android devices using MobileIron Core.

Related topics

Types of apps on Android devices

You can add the following kinds of apps for Android devices:

- Google Play Store apps
- In-house apps
- Secure apps (Secure apps are available only if you have configured the device to support AppConnect.)

NOTE: For information about apps on Android enterprise devices, see Managing mobile apps for Android enterprise.

What are Google Play Store apps?

Google Play Store apps are apps available for download from Google Play Store. You can add app recommendations from the Play Store to the App Catalog. When you apply labels to the apps, the apps are made available to the devices that have those labels. Device users see the apps made available to them in Apps@Work on their device.

NOTE: MobileIron Core can upload an Android Google Play Store app that has the same package name as a private in-house app, such as com.mobileiron.phoneatwork, that is already loaded on Core. Also, you can import an in-house app with the same package name as a public app that is already loaded on Core. This feature is always on and does not require any configuration in the user interface.

What are in-house apps?

In-house apps are mobile apps that you develop and distribute internally. MobileIron enables you to distribute and track in-house apps. You upload in-house apps to MobileIron Core. In-house apps appear in the Apps@Work list on the device for users to download.

NOTE: For Android enterprise, in-house apps are called "private apps" and you make them available after uploading them to your private Google Play Store for your domain. See also the MobileIron Core Device Management Guide for Android Devices.

What are secure apps?

Secure apps, also known as AppConnect apps, are apps that are developed internally or by third-party developers using AppConnect for Android. Secure apps are always in-house apps, but in-house apps are not always secure apps.

Access to secure apps and their data on Android devices are protected by AppConnect. You distribute secure apps internally like in-house apps. Device users log in with a single sign-on secure apps passcode to access these apps, and the data associated with the apps is encrypted. Secure apps can share data only with other secure apps.

Distributed secure apps appear in the "Secure Apps" menu in Mobile@Work for Android. Secure apps are not supported for Android enterprise.

For detailed information about AppConnect for Android and secure apps, see *MobileIron Core AppConnect and AppTunnel Guide*.

Adding Google Play apps for Android

You add a public Google Play app for Android devices to the App Catalog on the MobileIron Core Admin Portal.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Click **Add+.**
3. Click **Google Play** to open the app wizard.
4. Type the app name or package name and click **Search** to show the search results from Google Play.
5. Click the row to select the app you want to add to the App Catalog.
6. Click **Next**.
7. Use the following guidelines to complete the remaining options for each app.

Item	Description
Application Name	Displays the app name defined by the app developer. This is the name that displays to device users. This field is not editable.
Min. OS Version	The minimum OS version as retrieved from Google Play displays. Devices that don't have the minimum OS version installed will not be able to install the app.
Description	The app description as retrieved from Google Play displays. You can edit the description. Users will see this description in Apps@Work on their devices.
Category	Select one or more categories to display this app in a category tab in Apps@Work or add a new category. a. Click **Add New Category** to define new categories. b. Enter a category **Name** (up to 64 characters). c. Enter a **Description** (up to 255 characters). d. In the **Category Icon** section, click the **Replace Icon** button. e. Browse and select an icon that will represent this Category. f. Click **Save**.

8. Click **Finish**.
   The app displays in the **App Catalog** page when the **Platform** filter is set to **Android**.
9. Click **Next**.

Item	Description
Feature this App in the Apps@Work catalog	Select the check box to list the app in the Featured apps list in Apps@Work. This feature does not apply to AppConnect apps.
Featured Banner	Selecting this option will add this app as part of the top banner on Apps@Work Home screen on end user devices. The latest five apps will be picked to be part of Apps@Work Home page.

10. Click **Finish**.
    The app displays in the **App Catalog** page when the **Platform** filter is set to **Android**.
11. Apply the app to a label to have that app listed in Apps@Work on Android devices.

Delegated permissions for Google Play apps

For Android 8.0 and above devices, Mobile@Work allows delegation permissions for apps in Managed Device with Work Profile (COPE) mode.

- For Public or Self Hosted Apps (Google Play Private channel apps) pushed by Managed Google Play or regular Google Play:
  - Apps are assigned to device in Managed Device with Work Profile (COPE) mode and will be pushed and installed silently by Google Play services inside the Managed Device with Work Profile.
  - After the app is installed in the Managed Device with Work Profile mode, delegated permissions is applied by Mobile@Work.
  - This is supported for Samsung and non-Samsung devices running Android 8.0 through the latest version as supported by MobileIron.

Adding an app using Quick Import in the Core Admin Portal

Using Quick Import in the App Catalog is a fast way to add multiple apps using default settings. Options and configurations can be edited later as needed.

NOTE: If you have installed Android enterprise, the Quick Import option is disabled. You can use the Google Play iFrame to import apps.

Procedure

1. Go to **Apps > App Catalog** in the MobileIron Core Admin Portal.
2. Click **Quick Import > Google Play.**
3. Enter any part of an application name or package name.
4. Click **Search**. Search results from the Google Play Store appear.
5. Click **Import**, at the end of the line, to add the app to the App Catalog.
6. The store import dialog remains open so you can quickly search and add more apps.

7. Click **X** to close the dialog.
8. Edit the app details for the imported app and select **Install this app for Android enterprise**.

   NOTE: De-selecting this option does not uninstall the app from devices which already have it installed.
9. Fill out the Android enterprise-related restrictions as necessary.
10. Click **Save**.

All apps that are available to be installed for Android enterprise (because you have selected **Install this app for Android enterprise**) have the "suitcase" badge on their icon. These apps can also be installed on non-Android enterprise devices.

Note The Following:

- You can edit the app's settings at any time. Select the app in the App Catalog, and click **Edit**.
- The metadata and reviews for an app selected for installation from Google Play may not be displayed depending on the configuration of the customers firewall.

Whitelisting public apps for the Samsung Knox container

On Samsung Knox devices, you can whitelist public apps for the Samsung Knox container. When you add a public Android app in the App Catalog to a whitelist, device users can copy the app to the Samsung Knox container.

Note The Following:

- This feature is supported for Samsung Knox 2.1 through 2.6.
- In-house apps that you specify in the whitelist are automatically installed in the Samsung Knox container.
- Samsung Knox features are not supported on MAM-only Android devices.

Whitelisting a public app for the Samsung Knox container

You whitelist a public app for the Samsung Knox container by adding it to the Samsung Knox container setting for a device.

Procedure

1. Add Android apps from the Google Play Store in the **App Catalog**.
2. Edit a Samsung Knox Container configuration in **Policies & Configs > Configurations > Android > Samsung KNOX Container**.
3. In the **Apps** section of the configuration, add the Google Play Store app. The app is now whitelisted.

   NOTE: Adding the public app here does not install it into the Samsung Knox container on the device automatically. The device user must take action (see below). However, when

you add in-house apps to the **Apps** section, the in-house apps are automatically installed into the container.

4. To help users identify the whitelisted apps, MobileIron recommends that you add the apps to a distinct category in Apps@Work. For example, you can call the category “Whitelist for Knox”. The user will not see any distinctions for whitelisted apps.

Adding a whitelisted app into the Samsung Knox container

After you have whitelisted a public app for the Samsung Knox container, the device user takes the following steps on a Samsung Knox device to copy a whitelisted app into the Knox container.

Procedure

1. Launch Knox Settings.
2. Tap **Select apps to install**.
   The list of available apps appears. Note that the whitelisted apps are not distinguished from other apps. The user can refer to the special category in Apps@Work (if you set one up) to discover whitelisted apps.
3. Tap an app to install.
   - If the app selected is on the whitelist, the app will be installed inside the Knox container.

If the app is not on the whitelist, a notification informs the device user that a security policy prevented the app from installing.

Adding in-house apps for Android

In-house apps are the internally-developed apps that are uploaded to MobileIron Core. Core makes the apps available to Android devices based on labels that you assign to the apps and devices. You add in-house app to the App Catalog in the MobileIron Core Admin Portal.

Upon upgrade to Android 11, the Mobile@Work client no longer supports in-house apps for devices that migrate from Work Profile mode to Managed Device with Work Profile (COPE) mode. This also applies to new Android 11 devices provisioned as Work Profile for Company Owned Device mode.

NOTE: If your company needs time to figure out the migration plan for changing from Managed Device with Work Profile (COPE) mode to Work Profile for Company Owned Device mode, you can set the freeze firmware updates to Android 11 devices for up to 90 days. For more information, see "Setting the system update policy for Android devices" in the *MobileIron Core Device Management Guide for Android and Android enterprise Devices*.

If you are adding a new version of an existing app, see Adding new versions of an existing Android app.

Procedure

1. Go to **Apps > App Catalog.**
2. Click to **Add+**open the app wizard.
3. Click **In-house**.
4. Click **Browse** and navigate to the in-house app (.apk) you want to upload.

   NOTE: You cannot upload an in-house app that exceeds 2.15 GB.
5. Click **Next**.
   The app wizard examines the selected package to ensure that it meets requirements for in-house apps distributed for Android devices. If the package is acceptable, the next screen displays.
6. Use the following guidelines to complete the rest of the screens in the app wizard, clicking **Next** where applicable:

Section	Item	Description
General	Application Name	Displays the app name defined by the app developer. This is the name that displays to device users. This field is not editable.
	Display Version	Displays the version number defined by the app developer. This is the version that displays to device users. This field is not editable.
	Code Version	Displays the version defined for the package. This item is not editable.
	Description	Enter any additional text that helps describe what the app is for. Users can see this text in Apps@Work.
	Category	Select a category if you would like this app to be displayed in a specific group of apps on the device or add a new category. 1. Click **Add New Category** to define new categories. 2. Enter a category **Name** (up to 64 characters). 3. Enter a **Description** (up to 255 characters). 4. In the **Category Icon** section, click the **Replace Icon** button. 5. Browse and select an icon that will represent this Category. 6. Click **Save**.
Apps@Work Catalog	Feature this App in the Apps@Work catalog	If check box is selected, this app appears in the Featured Apps tab in Apps@Work.

Section	Item	Description
	Featured Banner	Selecting the check box will display this app as part of the top banner on the Apps@Work Home page on end users' devices. The latest five apps will be picked to be part of Apps@Work Home page.
	Allow app downloads over insecure networks	Select the check box if you are providing an Override URL (next field) that uses the HTTP URL scheme instead of HTTPS. Override URLs are intended for use behind a firewall, using a trusted and secure internal network. Before you use an HTTP URL, make sure you understand the risks of using an insecure connection.
	Override URL	If you are using an alternate source for downloading in-house apps, enter that URL here. The URL must point to the in-house app in its alternate location. Override URLs are intended for use behind a firewall, using a trusted and secure internal network. Manual synchronization is required with the alternate HTTP server on which app are stored. See Override for in-house app URLs for the requirements for this configuration before using it.
	App Icon	• Click the **Replace Icon** button to replace the icon.
	Screenshots	• Click **Upload** to select and upload optional screenshot files in PNG, GIF, or JPG formats. The supported dimensions are 480x800 pixels and 480x854 pixels. We recommend PNG for best resizing. • To upload additional screenshots, click **Upload.** • To clear the field, click **Remove**.
App Installation Settings	Require the user to install the latest version of the app in order to run it	Select the check box to ensure the user installs the latest version of this app. IMPORTANT: You must select this check box for the entries for each version of this same app in order for this feature to take effect. Clear the check box for all versions of this app to allow users to work with any version of this app. For more information, see Specify latest version

Section	Item	Description
		required for a secure app.
	Silent install for Mandatory Apps	This feature only applies to devices that support silent installation. • Clearing the check box means the device user will need to manually install the app. • Selecting the check box will install the app silently. The app is installed when the device checks in with Core. User action is not required. If "Silent install for Mandatory Apps" is enabled along with "Silent install for work managed devices", then "Silent install for Mandatory Apps" will take precedence and the app will be installed on the device irrespective of the constraints set for the "Silent install for work managed devices" option. Administrators will need to disable "Silent install for Mandatory Apps" if they want to configure the apps via the "Silent install for work managed devices" option. For more information, see Silent install and uninstall of mandatory apps. NOTE: Silent install is not supported for MAM-only Android devices.
Per App VPN Settings	Per App VPN by Label Only	Select this check box to require the Per App VPN configuration to be assigned to a label that matches the device. If there is no associated label between the VPN configuration and the device, Per App VPN will not be installed on the device. Clear this check box to assign the per App VPN based on the selections in the **Per App VPN** field, ignoring labels. NOTE: Per app VPN is not supported for MAM-only Android devices.
	License Required	The **Selected VPNs** column lists the VPN configuration that may be installed on the device, in priority order: • If **Per App VPN by Label Only** is selected, then the VPN configuration must be assigned to a label matching the device in order to be installed.

Section	Item	Description
		The first VPN in the list that is also assigned to a label associated with the device has the highest priority. • If **Per App VPN by Label Only** is not selected, then the VPN configurations listed are in priority order and do not need to be assigned to a label matching the device. To populate the **Selected VPNs** column, select the VPN configuration you created for per app VPN in the **All VPNs** column, and click the right arrow. You can select multiple per app VPN settings. To reorder the per app VPN configurations in the **Selected VPNs** column, drag the configuration names to the correct positions in the list. See "VPN settings" in the *MobileIron Core Device Management Guide* for information on creating a per app VPN. NOTE: Per app VPN is not supported for MAM-only Android devices.
Android Enterprise (All Modes)	Install this app for Android enterprise	Selecting this check box displays additional fields for Android enterprise app settings. You must be a Global Space administrator to use this setting. Select to enable public and private apps available to device users for download to Android devices. You can change the "Install this app for Android enterprise" setting for each app in the app's details page at any time.
	Silent install for work managed devices	This feature only applies to devices that support silent installation. • Clearing the check box means the device user will need to manually install the app. • Selecting the check box will install the app silently. The app is installed when the device checks in with Core. User action is not required. If "Silent install for Mandatory Apps" is enabled along with "Silent install for work managed devices", then "Silent install for Mandatory Apps" will take precedence and the app will be installed on the device irrespective of the constraints set for the "Silent install for work managed devices" option.

Section	Item	Description
		Administrators will need to disable "Silent install for Mandatory Apps" if they want to configure the apps via the "Silent install for work managed devices" option. NOTE: Silent install is not supported for MAM-only Android devices. Additional settings can be made for silent installs of work managed devices. These settings are applicable for public and private apps. Prerequisite apps are pushed before dependent apps. • Auto Install Mode - Self hosted apps will not be auto installed. - Do not Auto Install - Auto Install Once - Force Install (default) • Install Priority - You can prioritize downloading of specific apps before other apps. For example, prioritizing the download of Tunnel and Email apps before other non-critical apps. - Low - Medium (default) - High • Install only when connected to Wi-Fi - Default is de-selected. • Install only when charging - Default is de-selected. • Install only when Idle - Default is de-selected. For more information, see Silent install and uninstall of mandatory apps.
	Block Widget on Home Screen	If selected, the app cannot place widgets on the home screen on work profile devices. For example, calendar apps are not permitted to place calendar widgets on the home screen.
	Block Uninstall	Select this feature to prevent the device user from uninstalling the app.
	Quarantine app when device is quarantined	Required for: • Work Profile mode • Managed Device with Work Profile (COPE) mode on Android devices versions 8-10

Section	Item	Description
		• Work Profile on Company Owned Devices mode (Android 11 through the latest version as supported by MobileIron) Selected by default, this field enables configured compliance actions to hide the app if a policy violation results in a quarantined device. A second step is required to enable this feature: configure a corresponding compliance action and security policy with that compliance action selected. Once the device is no longer quarantined, the app can be used again. If this option is deselected, the app is available for usage, even when the device is quarantined. Note The Following: • If you change the setting after the app is added, the changed setting will be applied to the app.
Delegated Permissions	Delegated Permissions	Expand this section to apply delegated permissions to this app. Applicable on managed devices. For more information, see Delegated permissions for in-house apps.
	Configure third-party app runtime permissions	Select this check box to modify runtime permissions for other apps. • Applicable to in-house and public / private apps for managed devices and Managed Devices with Work Profile (COPE) mode starting from Android 8. • Applicable to public / private apps on managed profiles. • Applicable to public / private apps on Work Profile for Company Owned Device mode starting from Android 11.
	Hide and suspend third-party apps	Select this check box to delegate access to this app to have permission to hide and suspend apps. • Applicable to in-house and public / private apps for managed devices and Managed Devices with Work Profile (COPE) mode starting from Android 8. • Applicable to public / private apps on managed

Section	Item	Description
		profiles. • Applicable to public / private apps on Work Profile for Company Owned Device mode starting from Android 11.
	Manage certificates	Select this check box to allow this app to have access to certificate APIs on the device. • Applicable to in-house and public / private apps for managed devices and Managed Devices with Work Profile (COPE) mode starting from Android 8. • Applicable to public / private apps on managed profiles. • Applicable to public / private apps on Work Profile for Company Owned Device mode starting from Android 11.

7. Click **Finish**.
   The app displays in the **App Catalog** screen. The **Source** column displays the app as an in-house app.
8. In order to distribute your app from Google Play store, you need to download the APK Definition file and add the app license key to MobileIron Core.

Delegated permissions for in-house apps

For Android 8.0 and above devices, Mobile@Work allows delegation permissions for in-house apps in Managed Device with Work Profile (COPE) mode. See also Delegated permissions for Google Play apps

- For in-house Apps (Apps pushed by Core):
  - Apps are assigned to devices in Managed Device with Work Profile (COPE) mode and will be installed silently by Mobile@Work on the personal (device owner) side.
  - After the app is installed, delegated permissions are applied by Mobile@Work.
  - This is supported for Samsung and non-Samsung devices running Android 8.0 through the latest version as supported by MobileIron.
- For In house Apps on Samsung Knox V3 devices (Android 8.0 and above):
  - Apps are assigned to device in Managed Device with Work Profile (COPE) mode and whitelisted for Knox V3 workspace.
  - Apps are silently installed by Mobile@Work on the personal (Device Owner) side and then immediately hidden and moved to the Knox V3 workspace (Managed Device with Work Profile (COPE) mode.)
  - At the time the app is moved into the Knox V3 workspace, delegated permissions are applied.

NOTE: Installing regular in-house apps inside the Managed Device with Work Profile (COPE) mode is not supported.

Adding new versions of an existing Android app

When uploading a newer version of an app, an extra page opens to allow you to select whether to keep the app's old version information or to adopt the information from the app's new version. This feature is applicable to Android in-house / private / self-hosted apps.

Procedure

1. In the App Catalog, click the **Add+** button.
   The Add App Wizard opens.
2. Click **In-House**.
3. Click **Browse** and navigate to the in-house Android or Android enterprise app you want to upload.
4. Click **Next**.
   The An earlier version of this App exists page opens.
5. Select an option:
   - **Another version of this App was previously uploaded. Reuse its description, icon and screenshot(s).** If the Description, Icon or Screenshot fields of the new app are empty, then the system will populate those fields with information from the previous app version (default).
   - **Upload a new description, icon or screen shot.** Information related to the Description, Icon or Screenshot fields of the new App will be utilized. If those fields are empty, nothing will be copied from the previous app version.
6. Click **Next** and finish configuring the new version of your app (see Adding your Android enterprise private app using the app wizard in the Core Admin Portal.)
   Once finished, the new version displays in the App Catalog.

Adding secure apps for Android

You upload all secure apps and the Secure Apps Manager to MobileIron Core as in-house apps. Core makes the apps available to Android devices based on labels that you assign to the apps and devices.

The apps that you upload include:

- the Secure Apps Manager that MobileIron provides.
- The Secure Apps Manager is required for AppConnect to work. See *MobileIron Core AppConnect and AppTunnel Guide* for more information about Secure Apps Manager.
- the AppConnect apps that MobileIron provides that your enterprise uses.
- the AppConnect apps that your enterprise wrapped.

- See the MobileIron Core AppConnect and AppTunnel Guide for more information about AppConnect and third-party/in-house secure apps.

NOTE: MobileIron Core has the ability to upload an Android Google Play Store app that has the same package name as a private in-house app, such as com.mobileiron.phoneatwork, that is already loaded on Core. Also, you can import an in-house app with the same package name as a public app that is already loaded on Core. This feature is always on and does not require any configuration in the user interface.

Before you begin: Get the Secure Apps Manager and the other AppConnect apps that MobileIron provides from the support.mobileiron.com site. Save them to a location accessible from your MobileIron Core.

To add a secure app to the App Catalog:

1. Go to **Apps > App Catalog.**
2. Click **Add +** to open the app wizard.
3. Click **In-house.**
4. Click **Browse** and navigate to the secure app (.apk) you want to upload.

   NOTE: You cannot upload an in-house app that exceeds 2.15 GB.
5. Click **Next**.
   The app wizard examines the selected package to ensure that it meets requirements for in-house apps distributed for Android devices. If the package is acceptable, the next screen displays.
6. Use the following guidelines to complete the rest of the screens in the app wizard:

Item	Description
Application Name	Displays the app name defined by the app developer. This is the name that displays to device users. This field is not editable.
Display Version	Displays the version number defined by the app developer. This is the version that displays to device users. This field is not editable. NOTE: The version number for AppConnect apps includes: • the version number defined by the app developer • additional numbers provided by the wrapping process
Code Version	Displays the version defined for the package. This item is not editable.
Description	Enter any additional text that helps describe what the app is for. This text appears on the target devices under the app name in the Secure Apps list.

Item	Description
	MobileIron recommends that you add the following descriptions for the AppConnect apps that MobileIron provides: • the Secure Apps Manager The Secure Apps Manager works with the Mobile@Work app to secure and manage secure apps on your device. • TouchDown for SmartPhones TouchDown for SmartPhones provides secure access to your company email, contacts, calendar, and tasks. • File Manager File Manager allows you to securely navigate and manage your company files. • Email+ for Android Email+ for Android provides the native email client experience with ease of setup and important other features. • Web@Work for Android Web@Work for Android is a secure browser that allows your device users to easily and securely access your organization's web content.
Category	Select one or more categories to display this app in a category tab in Apps@Work or add a new category. 1. Click **Add New Category** to define new categories. 2. Enter a category **Name** (up to 64 characters). 3. Enter a **Description** (up to 255 characters). 4. In the **Category Icon** section, click the **Replace Icon** button. 5. Browse and select an icon that will represent this Category. 6. Click **Save**.
Feature this App in the Apps@Work catalog	By default, the check box is selected to list the app in the Featured apps list in Apps@Work. This feature does not apply to AppConnect apps.
Featured Banner	Checking this option will add this app as part of the top banner on Apps@Work Home screen on end user devices. The latest five apps will be picked to be part of Apps@Work Home page.
Allow app downloads over insecure networks	Select this if you are providing an Override URL (next field) that uses the HTTP URL scheme instead of HTTPS.

Item	Description
	Override URLs are intended for use behind a firewall, using a trusted and secure internal network. Before you use an HTTP URL, make sure you understand the risks of using an insecure connection.
Override URL	If you are using an alternate source for downloading in-house apps, enter that URL here. The URL must point to the in-house app in its alternate location. Override URLs are intended for use behind a firewall, using a trusted and secure internal network. Manual synchronization is required with the alternate HTTP server on which app are stored. See Override for in-house app URLs for the requirements for this configuration before using it.
App Icon	NOTE: Icon and Screenshots appear when editing an app entry. • The icon retrieved from Google Play displays. • To replace the icon, click **Replace Icon** button. Select the icon to represent this app. The file must be no larger than 1024 x 1024 pixels and in JPG, PNG, or GIF format. We recommend PNG for best resizing results. Icon height and width must be equal.
Screenshots	NOTE: Icon and Screenshots appear when editing an app entry. • The screenshots retrieved from Google Play are displayed. • Click **Upload** to select and upload optional screenshot files in PNG, GIF, or JPG formats. The supported dimensions are 480x800 pixels and 480x854 pixels. We recommend PNG for best resizing. • To delete a screenshot, click **Remove** under the screenshot.
Require the user to install the latest version of the app in order to run it	This feature applies only to AppConnect apps. Select the check box to ensure the user installs the latest version of this app. IMPORTANT: You must select this check box for the entries for each version of this same app in order for this feature to take effect.

Item	Description
	Clear the check box for all versions of this app to allow users to work with any version of this app. For more information, see Specify latest version required for a secure app.
Silent install for Mandatory Apps	This feature only applies to devices that support silent installation. • Clearing the check box means the device user will need to manually install the app. • Selecting the check box will install the app silently. The app is installed when the device checks in with Core. User action is not required. For more information, see Silent install and uninstall of mandatory apps. NOTE: Silent install is not supported for MAM-only Android devices.
Enforce this version for Mandatory Apps	This feature applies only to mandatory in-house apps. Version enforcement is not available for AppConnect apps or apps from Google Play. Select the check box to require this version of the in-house app on devices, even if newer or older versions of the same app .apk are uploaded to the App Catalog. NOTE: In order for this to take effect, you will need to set the **Mandatory**field in the Apply to Labels dialog box to **Yes**. See Enforcement of specific app versions for mandatory in-house apps for more information, including how to achieve desired results when multiple versions of the same app are in the App Catalog.

Item	Description
Per App VPN by Label Only	Select this check box to require the Per App VPN configuration to be assigned to a label that matches the device. If there is no associated label between the VPN configuration and the device, Per App VPN will not be installed on the device. Clear this check box to assign the per App VPN based on the selections in the **Per App VPN** field, ignoring labels. NOTE: Per app VPN is not supported for MAM-only Android devices.
License Required	The **Selected VPNs** column lists the VPN configuration that may be installed on the device, in priority order: • If **Per App VPN by Label Only** is selected, then the VPN configuration must be assigned to a label matching the device in order to be installed. The first VPN in the list that is also assigned to a label associated with the device has the highest priority. • If **Per App VPN by Label Only** is not selected, then the VPN configurations listed are in priority order and do not need to be assigned to a label matching the device. To populate the **Selected VPNs** column, select the VPN configuration you created for per app VPN in the **All VPNs** column, and click the right arrow. You can select multiple per app VPN settings. To reorder the per app VPN configurations in the **Selected VPNs** column, drag the configuration names to the correct positions in the list. See "VPN settings" in the *MobileIron Core Device Management Guide* for information on creating a per app VPN. NOTE: Per app VPN is not supported for MAM-only Android devices.

7. Click **Finish**.

   The app displays in the **App Catalog** screen with an icon that identifies the app as an in-house app.

   NOTE: You know the app is an AppConnect app by looking at its version number. The version number for an AppConnect app is a concatenation of the original app's version number and a version number from wrapping the app.

Mandatory and optional in-house and secure apps

An Android in-house app made available through the **App Catalog** can be designated as a *mandatory* app, which means that the app is always installed on the devices matching the app's labels. An app that is not marked as mandatory is *optional*, and enables the users to decide whether or not to install the app on their devices. The in-house app can be either an AppConnect app (secure app) or a regular, non-AppConnect app.

NOTE: Designating the Secure Apps Manager as optional and all secure apps as optional means that the device user sets up the secure apps container on-demand. See On-demand secure apps container setup.

NOTE: To set the prerequisite app for a dependent app, see App management action workflows.

Silent install and uninstall of mandatory apps

You can specify that mandatory in-house apps and secure apps are silently installed and uninstalled on:

- Samsung Knox devices MDM version 1.0 and through the most recently released version as supported by MobileIron
- Zebra MX running version 4.4 and through the most recently released version as supported by MobileIron
- LG devices that support silent installation - The LG device must be running:
  - Android 7.0 through the most recently released version as supported by MobileIron
  - Mobile@Work 9.7 for Android through the most recently released version as supported by MobileIron

The **Silent install for Mandatory Apps** feature eliminates any dependency on the device user to install or uninstall the app. Also, when you retire a device (when, for example, it is lost or stolen or the employee has left the company), the silently installed in-house and secure apps are silently uninstalled, thereby protecting the apps and their data.

Note The Following:

- Samsung Knox devices prevent the user from uninstalling the app.
- Silent install and uninstall are not supported on MAM-only Android devices.
- Silent install and uninstall are not supported for apps from the Google Play Store.
- AppConnect apps are not supported on devices using the Samsung Knox container. Do not install AppConnect apps if you are using the Samsung Knox container on the same device.

Uninstall behavior for silently installed apps

Installed apps are silently uninstalled when:

- No label maps the in-house or AppConnect app to the device.
  You apply labels to in-house and AppConnect apps to make the apps available to devices. Removing the label from the app or the device causes Mobile@Work to uninstall the app.
- You retire the device.
- You remove the in-house or AppConnect app from MobileIron Core.

Whether device users are notified to install a mandatory app

Although a mandatory app is always installed on the device, whether the device user sees a notification to install the app depends on whether the silent installation feature is enabled. The following table specifies when a device user sees the notification:

TABLE 13. MANDATORY APP INSTALLATION INTERACTION WITH SILENT INSTALLATION

Is app marked for silent installation?	Samsung Knox, Zebra, and LG devices that support silent installation	Devices that do not support silent installation
Yes	Silently installed with no notification to user	Notification to user to install
No	Notification to user to install	Notification to user to install

Device user experience with uninstalling a mandatory app

The device user experience when attempting to uninstall a mandatory app depends on the type of device, as specified by the following table:

TABLE 14. DEVICE USER EXPERIENCE WITH UNINSTALLING A MANDATORY APP

	Samsung Knox	Zebra devices that support silent install/uninstall	LG devices that support silent install/uninstall	Devices that do not support silent install/uninstall
Can device user uninstall a mandatory app when the silent install/uninstall feature **is** enabled?	No	Yes, but the app will be silently reinstalled.	Yes, but the app will be silently reinstalled.	Yes, but the device user will be notified to re-install the app.
Can device user uninstall a mandatory app when the silent install/uninstall feature is **not** enabled?	No	Yes, but the device user will be notified to re-install the app.	Yes, but the device user will be notified to re-install the app.	Yes, but the device user will be notified to re-install the app.

Designating an in-house app as optional or mandatory

After you have added the app to the App Catalog, you can designate whether it is an optional or mandatory app.

The below procedure applies to:

- Android in-house apps
- iOS in-house and public apps
- MacOs in-house and public apps
- AppConnect apps

Procedure

1. In the MobileIron Core Admin Portal, go to **Apps > App Catalog**.
2. Select an app and then select **Actions > Apply to Labels**.
3. In the Apply to Labels dialog box, select the check box next to the app's name.
4. Click in the **Mandatory** field, a drop-down displays. Selecting **Yes** makes the selected app mandatory; leaving it to the default **No** makes the app optional.
5. Click **Apply**.

Related topics

Enforcement of specific app versions for mandatory in-house apps

Enforcement of specific app versions for mandatory in-house apps

You can configure a mandatory in-house app to limit its installation on devices to a specific version of the app, even if newer or older versions of the same app .ipa are uploaded to the MobileIron Core's app catalog. You can also ensure that any version of the same app is installed, regardless of which version. The option called **Enforce this version for Mandatory Apps** is available in the App Catalog app wizard.

The version enforcement feature is supported only with regular (non-AppConnect) in-house apps. It does not apply to AppConnect apps or Google Play apps.

Use the version enforcement feature to:

- Ensure devices have the in-house app installed, regardless of version number.
- Lock users to a particular version of the Mobile@Work app. This applies to organizations that install Mobile@Work as an in-house app instead of installing it from Google Play.
- Ensure users do not upgrade to a new version of an in-house app while the newer version is still undergoing

testing.

- Downgrade users to a previous version of an in-house app.

Setting up version enforcement for an in-house app

You can enable or disable enforcing a specific app version for an in-house app on an Android device when you upload the app to MobileIron Core.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog.**
2. Click **Add+.**
3. Click **In-House.**
4. Click **Browse...** to select your in-house app. (Must not be an AppConnect app.)
5. Fill out the app wizard as needed; under **App Installation Settings**, select **Enforce this version for Mandatory Apps.** If this check box is not selected, then enforcing a specific app version will not apply. See: Enforcing an app version when you have uploaded multiple versions to Core**.**
6. Finish filling out the app wizard as needed. Click **Finish**.
7. Select the app in the App Catalog.
8. Click **Actions > Apply to Labels.**
9. In the Apply to Labels dialog box, select the check box next to the app's name.
10. Click in the **Mandatory**field, a drop-down displays. Selecting **Yes** makes the selected app mandatory; leaving it to the default **No**makes the app optional. If the Mandatory field is not set to **Yes**, the latest version of the app will not be enforced.
11. Click **Apply**.

Enforcing an app version when you have uploaded multiple versions to Core

If you have multiple versions of the same mandatory in-house .ipa file uploaded to MobileIron Core, you may wish to ensure one of the following scenarios:

- devices always get the latest version of the app. (App updates are forced.)
- devices have the app installed, regardless of the version number. (App updates are not forced.)
- devices remain on an older version of the app.
- devices are downgraded to an older version of the app.

Assuming your in-house app has versions 1.0, 2.0, and 3.0 in order from oldest to newest, and all three are uploaded to Core, use the settings described in the following table to achieve the desired results.

Note that having a label means that same label is applied both to the device and to the app. If a device is assigned to many labels, but at least one label has the Mandatory field set to **Yes**, then the device will have that app as mandatory.

TABLE 15. APP VERSION SETTINGS

Desired Result	Label and app settings (in App Catalog)
Ensure that any version of the app is installed on the device	For app version 1.0: Enforce this version is not selected For app version 2.0: Enforce this version is not selected For app version 3.0: Enforce this version is not selected Label must be applied to any or all versions of the app.
Allow only version 2.0	For app version 1.0: Enforce this version: irrelevant For app version 2.0: Enforce this version is selected For app version 3.0: Enforce this version: irrelevant Label must be applied to app version 2.0 only. Label must not be applied to all other app versions.
Ensure the latest version is always installed	Enforce this version is selected on the most recent app version (3.0). Enforce this version is irrelevant on older app versions (1.0, 2.0). Label must be applied to latest app version (3.0) Label may be applied to all app versions.
Downgrade users to version 1.0	App version 1.0: Enforce this version is selected; Label is applied. App version 2.0: Label is removed. App version 3.0: Label is removed.

NOTE: Mandatory apps can be silently installed and uninstalled on some devices. When not silently installed, the device user is prompted to install or uninstall a mandatory app.

Related topics

Mandatory and optional in-house and secure apps

Apps@Work in Mobile@Work for Android

Apps@Work enables device users to view, install, update, reinstall, and search for the apps made available to them by the MobileIron Core administrator. On Android, Apps@Work is available to users as a menu item in the Mobile@Work app. Apps@Work authenticates to MobileIron Core using either certificate authentication or token-based authentication.

Apps@Work displays the apps that you make available to the device through labels. In the Admin Portal, you assign an app to one or more labels. A device that is assigned to the same label as the app will have access to that app in Apps@Work.

Within Apps@Work, apps are organized into the Featured and Category tabs. If you have enabled ratings and reviews, the device user sees reviews, and can rate apps and write reviews. You can choose apps to be displayed as Featured Apps in the Apps@Work home screen.

Apps@Work for Android authentication to MobileIron Core

You determine whether Apps@Work authenticates to MobileIron Core using:

- Token authentication - Apps@Work uses a token to authenticate to Core. Core sends Mobile@Work the token when Mobile@Work registers with Core.
- Certificate authentication - Apps@Work authenticates to Core using an identity certificate.

  This certificate is specified by the certificate enrollment setting in the mutual authentication setting in the Admin Portal at **Settings > System Settings > Security > Certificate Authentication**.

  Using certificate authentication for Apps@Work on Android devices requires:

  - mutual authentication is enabled on Core.
  - the device is running Mobile@Work 10.2.0.0 through the most recently released version as supported by MobileIron.
  - the device is running Android 5.0 through the most recently released version as supported by MobileIron.

Note The Following:

- If certificate authentication is selected, but some of the requirements for certificate authentication are not met, token-based authentication is used.
- If certificate authentication is selected, and all of the requirements for certificate authentication are met, if the authentication fails for some reason, the device user cannot use Apps@Work. There is no fallback to using token-based authentication in this case.
- By default, certificate authentication is selected.

Related topics

- Configuring Apps@Work for Android authentication to MobileIron Core
- Setting up Apps@Work for iOS and macOS
- "Mutual authentication between devices and MobileIron Core" section in the *MobileIron Core Device Management Guide for Android and Android enterprise Devices*

Configuring Apps@Work for Android authentication to MobileIron Core

To configure how Apps@Work authenticates to MobileIron Core:

Procedure

1. In the Admin Portal, go to **Apps > Apps@Work Settings**.
2. To enable certificate authentication to Core, in the App Storefront Authentication box, select **Certificate Authentication**.
   Note that requirements for using certificate authentication are listed in Apps@Work for Android authentication to MobileIron Core. If any of these requirements are not met, Apps@Work uses token-based authentication to authenticate to Core, even when **Certificate Authentication** is selected.
3. To disable certificate authentication to Core, and use only token-based authentication, in the App Storefront Authentication box, deselect **Certificate Authentication**.
   Note that deselecting certificate authentication also means that Apps@Work on iOS devices does not use certificate authentication.

Related topics

- Apps@Work for Android authentication to MobileIron Core
- Setting up Apps@Work for iOS and macOS
- "Mutual authentication between devices and MobileIron Core" section in the *MobileIron Core Device Management Guide for Android and Android enterprise Devices*

Adding apps to Apps@Work for Android devices

Apps in the App Catalog must be assigned to one or more labels to be available in Apps@Work on the devices.

Procedure

1. In the Admin Portal, select **Apps > App Catalog**.
2. Select **Android** from the **Platform** list.
3. Select the app you want to work with.
4. Click **Actions > Apply to Label**.
5. Select the label that represents the Android devices on which you want the selected app to be listed.
6. If, during the installation of the selected app, the **Enforce this version for Mandatory Apps** field was selected (checked), the Apply to Label dialog box will display **Yes** in the Mandatory field. Otherwise, the Mandatory field displays **No**.
7. Click **Apply**.
   To set the prerequisite app for a dependent app, see App management action workflows.

Device user experience of Apps@Work on an Android device

The device user taps **Apps@Work** on the menu in Mobile@Work to access the app store. Apps@Work organizes the apps under three main tabs:

- Featured tab
  - The featured screen lists all apps that are designated as featured apps by the administrator.
- Categories tab
  - An app can be listed under Featured as well as under multiple categories.
  - Uncategorized apps are displayed under Uncategorized in the Categories tab.
  - Only categories that have at least one app are displayed.
  - Categories are defined by administrators as they add apps in the App Catalog in Core.
- Updates tab
  - When a user's device checks in, the Update tab displays a badge number indicating the number of in-house and public app updates available for the device user to download. Once the user updates the apps, the badge number will disappear on next device check-in.

Apps are listed in alphabetical order.

Notification of newly-published apps

When a featured app or an update to an installed app is published to device users, those users receive a notification in the form of a badge that appears next to the appropriate app list. The number on the badge indicates the number of apps available.

If the user deletes a published app, that app will not become available for reinstalling again until the next sync interval causes MobileIron Core to be updated. You can address device user concerns by using the **Force Device Check-In** command to force Mobile@Work to update Core.

App details in Apps@Work on an Android device

Tap the app to view its details screen. If the administrator enabled ratings and reviews, tap the Reviews tab to read reviews, or write a review if you have already installed the app.

One of the following buttons appears on the details screen:

- View: takes you to view or install the app in the Google Play Store.
- Install: installs the app.
- Reinstall: downloads and reinstalls the app.
- *Open*: launches the app.

Searching for an app in Apps@Work on an Android device

Tap the search icon on the title bar to initiate a search within Apps@Work. Type any part of an app's name and tap the return key. The search results are displayed. Tap **Cancel** next to the search text entry box to exit search mode.

Localized Apps@Work on an Android device

Apps@Work is available translated to the languages supported by Mobile@Work. The text and messages in Apps@Work appear in the device's local language when the language is enabled in the MobileIron Core language preferences.

To enable languages in the MobileIron Core Admin Portal:

1. Go to **Settings > System Settings > General > Language**.
2. In the **Language** section, select the desired languages.
3. Click the right arrow to move the selection to **Enabled Languages**.

On-demand secure apps container setup

Sometimes when you configure device users to be able to use secure apps, some users do not immediately need to use the apps. These users can set up the secure apps container on-demand if all their secure apps are optional apps.

Recall that you designate an app as optional or mandatory when you upload it to the App Catalog on MobileIron Core. When all the secure apps are optional, the device users install the Secure Apps Manager and create a secure apps passcode only when they install their first secure app.

To configure MobileIron Core to allow on-demand secure apps container setup, you designate the Secure Apps Manager as optional in Core's App Catalog.

The following table summarizes the behavior:

	One or more mandatory secure apps	**No mandatory secure apps**
Secure Apps Manager is mandatory	User is prompted to create the AppConnect container during setup.	User is prompted to create the AppConnect container during setup.
Secure Apps Manager is optional	User is prompted to create the AppConnect container during setup.	User does not create AppConnect container until he requests a secure app.

On-demand secure apps container setup improves the device user's experience. Until the device user needs a secure app, the device user does not have to set up the AppConnect container. Setting up the AppConnect container requires the device user to:

- download and install the Secure Apps Manager
- create an AppConnect passcode

Now device users have to go through this process only when they are ready to use a secure app.

Interactions with on-demand secure apps container setup

File Manager interaction

The secure File Manager provides some capabilities that secure apps use. These capabilities include:

- An image viewer
- An HTML viewer
- A text viewer
- A ZIP file extractor
- A file download manager

If other secure apps require these features, File Manager must be installed.

Do one of the following:

- If you make secure apps container setup on-demand, inform device users to always install the File Manager if they install any other secure app.
- Make the File Manager app mandatory, thereby not using on-demand container setup.

Email client interaction

The Exchange setting on the MobileIron Core Admin Portal allows you to list a priority order for Android email apps. This order indicates the preferred app for Mobile@Work to set up as the device's email client. If you do not specify a list, Mobile@Work looks for the following unsecured apps in this order: Email+, TouchDown for SmartPhones, the email app native to the device.

IMPORTANT: Always specify an Exchange setting email app priority list if you are using on-demand AppConnect container setup. If the list is empty, Mobile@Work will set up an unsecured email app such as the native email app.

To ensure the use of a secure email app, do one of the following:

- Make all secure apps optional, and put only secure email apps in the priority list.
  In this case, the device user cannot use email until he installs a secure email app, but you ensure the device user does not use an unsecured email app.
- Make a secure email app mandatory, and put it in the Exchange setting list.
  In this case, the device user is prompted to set up the AppConnect container, and the secure email app is installed and set up. However, using this method means the device users do not benefit from on-demand secure apps container setup.

NOTE: For MAM-only Android devices, this priority list is not applicable because the Exchange setting is not supported. For MAM-only Android devices, use the AppConnect-enabled Email+ for Android.

Secure Apps Manager or secure app upgrade interaction

Consider the scenario when:

- You add a newer version of the Secure Apps Manager or a secure app to the App Catalog
- You designate the newer version as optional.

If the older Secure Apps Manager or secure app is already installed on a device, the device user is prompted to install the upgraded app, even though it is designated optional. This behavior ensures that the user installs the upgraded version.

Configuring on-demand secure apps container setup

To configure on-demand secure apps container setup:

- Designate all secure apps as optional.
- Designate the Secure Apps Manager as optional.

IMPORTANT: If you set the Secure Apps Manager and secure apps as optional, set all versions of Secure Apps Manager and secure apps in the App Catalog to optional.

Designating the Secure Apps Manager or secure app as optional during upload

You can designate the Secure Apps Manager or secure app as optional when you upload it to the App Catalog on the MobileIron Core Admin Portal.

For example, for the Secure Apps Manager:

1. Go to **Apps > App Catalog**.
2. Click **Add +** to open the app wizard.
3. Click **In-house**.
4. Click **Browse** to select and upload the Secure Apps Manager.
5. Continue through the app wizard filling out fields as needed until you reach **Silent install for Mandatory Apps** field. To make the Secure Apps Manager optional, make sure the check box is cleared.
6. Fill out the remaining fields as needed.
7. Click **Finish**.

Designating the Secure Apps Manager or secure app as optional after upload

You can change an app to optional on the MobileIron Core Admin Portal at any time.

For example, for the Secure Apps Manager:

1. Go to **Apps > App Catalog.**
2. Under **Platform**, select **Android**.
3. Find the **Secure Apps Manager** and click the app.
4. Click **Edit**.
5. Find the **Silent Install for Mandatory Apps** field and clear the check box to make the **Secure Apps Manager** optional.
6. Click **Save**.

Device user view of on-demand secure apps container setup

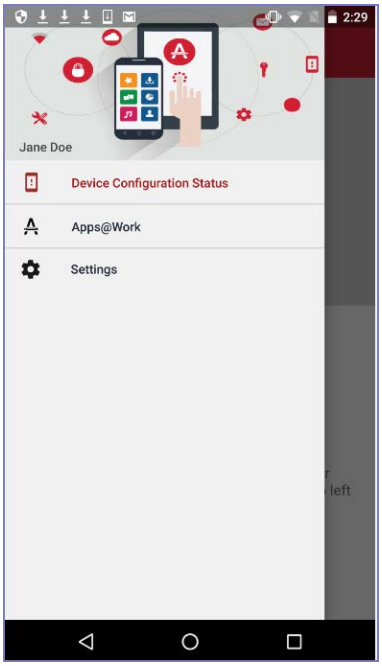
When you configure on-demand secure apps container setup, Mobile@Work does not prompt the user to set up the AppConnect container until the user requests to download a secure app. On-demand setup occurs if both of the following are true:

- The Secure Apps Manager is optional
- All the secure apps assigned to the device are optional.

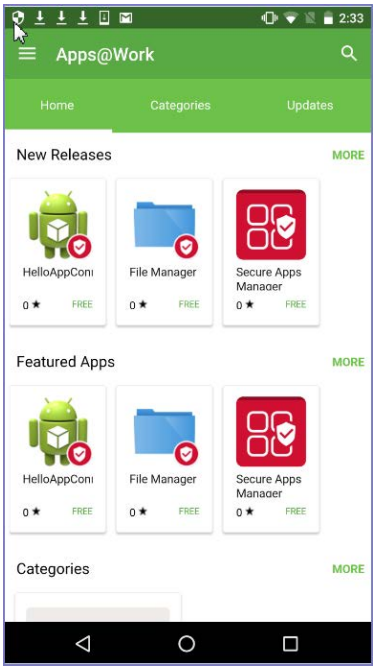
Designating the Secure Apps Manager as optional impacts what displays in the Secure Apps menu item in Mobile@Work. The Secure Apps menu item shows the Secure Apps Manager only if it is mandatory. It shows a secure app only if it is mandatory. If the Secure Apps Manager is optional, and all the secure apps are optional, the Secure Apps menu item does not appear.

The following steps illustrate the process that the device user experiences when he requests a secure app for the first time.

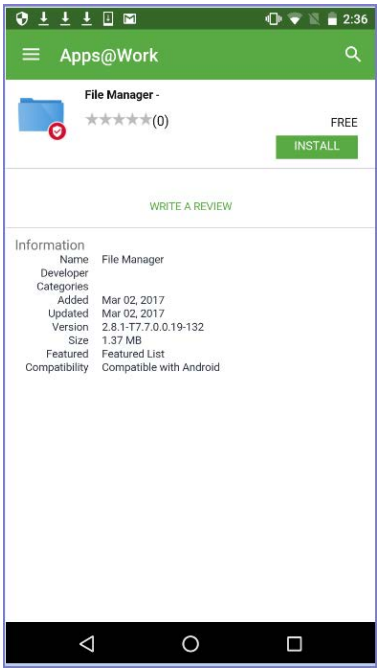
1. In Mobile@Work 9.3 menu, the user taps **Apps@Work**.



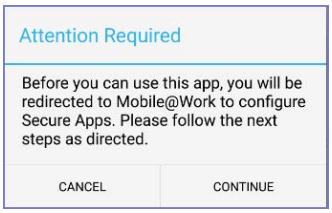
Apps@Work displays the available apps.



2. The user taps a secure app, such as File Manager.

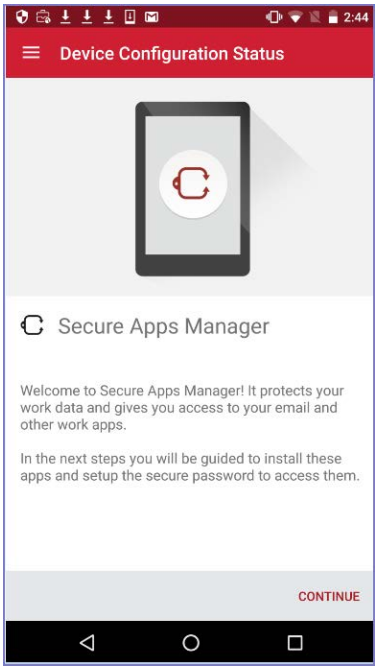


3. The user taps **Install**. Mobile@Work informs the user that Secure Apps must be configured.

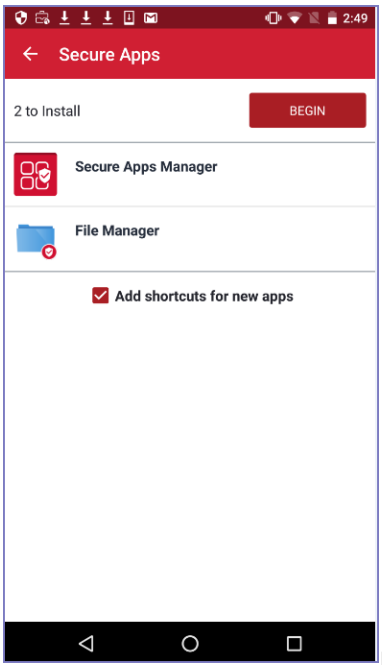


4. The user taps **Continue**.

NOTE: The remaining steps are similar to what the user experiences at registration when the Secure Apps Manager is mandatory.



5. The user taps **Continue**.



NOTE: The Secure Apps Manager is included for installation along with the secure app that the user requested.

6. The user taps **Begin**, and follows the instructions to install the Secure Apps Manager and the requested app.
   After the apps are installed, the user is prompted to set up the secure apps passcode.

After the user sets up the secure apps passcode, the AppConnect container is set up. The Secure Apps Manager and the selected secure app are installed and ready to use.

Specify latest version required for a secure app

You can specify that the latest version of a secure app is required. When you upload a new version of a secure app to the App Catalog on the MobileIron Core Admin Portal, you can specify this requirement. You can also later edit the app to specify this requirement. This requirement means that the device user can no longer run the older version of the app. When the device user attempts to run the older version, he is prompted to install the newer version.

This feature is available only for secure apps, not for unsecured apps.

By requiring that device users upgrade to the latest version of a secure app:

- You can ensure that all users have the latest features, fixes, and security upgrades.
- You can ensure all users are using the same set of secure apps. This consistent deployment across all devices simplifies your environment and support needs.

A special case involves the Secure Apps Manager. You can specify that the latest version of the Secure Apps Manager is required. In this case, the device user cannot run *any* secure app until he upgrades the Secure Apps Manager. Normally, you do **not** select this option for the Secure Apps Manager unless it contains security fixes that you require.

NOTE: Do not specify that the latest version of the Secure Apps Manager is required for the typical Secure Apps Manager upgrade scenario. If a device user installs a secure app that requires the latest version of the Secure Apps Manager, the latest Secure Apps Manager is automatically installed.

Requiring the latest version of a secure app

First, specify that a secure app's latest version is required when you upload it to the App Catalog on the MobileIron Core Admin Portal:

1. Go to **Apps > App Catalog**.
2. For Platform, select **Android**.
3. Click **Add +**.
4. Select **In-house**.
5. Click **Browse** to select and upload a secure app.
6. Fill out fields as needed until you find **Require the user to install the latest version of the app in order to run it**. Select **Yes**.
7. Fill out the remaining fields as needed.
8. Click **Finish**.

Next, do the same for each older version of the same app in the App Catalog. Set the field **Require the user to install the latest version of the app in order to run it** to **Yes**. (If you select **Yes**, select **Yes** for every version of the app.) Click **Finish**.

NOTE: You can also edit an app at a later time to select the option **Require the user to install the latest version of the app in order to run it.**

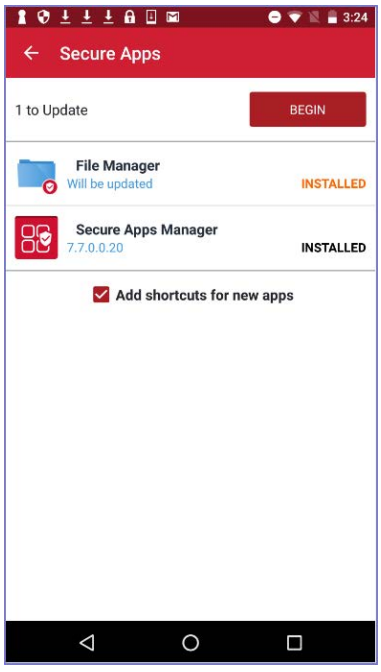
Device user experience when latest version of an app is required

The device user experiences the following:

- If you require the latest version of the Secure Apps Manager, and the device user has an older version, he cannot run any secure app until he updates the Secure Apps Manager.
- The device user cannot run an app if you have required the latest version of the app and the device user has an older version.

When the user attempts to launch the older version of an app, he is automatically taken to Mobile@Work. A toast message appears briefly over the Mobile@Work Secure Apps screen. The toast message says “Please update this Secure App and wait for sync to complete”. The device user follows the instructions to install the app.

Similarly, if the device user launches Mobile@Work when a newer version of an app must be installed, Mobile@Work prompts the user to configure secure apps:



The device user taps **Begin**, and follows the instructions to install the updated version of the app.

Now when the device user launches the secure app, he is launching the latest version.

Related topics

App management action workflows

Secure apps installation order

You can specify the installation order for AppConnect apps (also called secure apps).

Specifying the installation order is not typically necessary. It is necessary only for a secure app that has a particular dependency on another secure app, which is not a common situation. Installing such apps in the wrong order can result in the apps not working properly. The app developers or app vendor will indicate whether such a dependency exists.

When such a dependency does exist, you configure an installation order only for the interdependent secure apps, not for all secure apps. Secure apps that are not part of the installation order are installed in alphabetical order by app name after the configured interdependent apps.

By specifying the installation order for secure apps, you ensure that apps that depend on each other can work properly.

Secure app installation order with optional secure apps

You can designate an Android secure app as optional. If you list an optional secure app in the installation order configuration, it is not installed unless the device user specifically chooses to install it.

Therefore, typically, if you list a secure app in the installation order configuration, designate it as a mandatory app. The reason the app is in the installation order configuration is because other apps depend on it to work properly, or it depends on other apps. If other apps depend on it, but it is optional and not installed, the device user will experience functionality issues with the apps that depend on it.

Specifying the installation order for secure apps

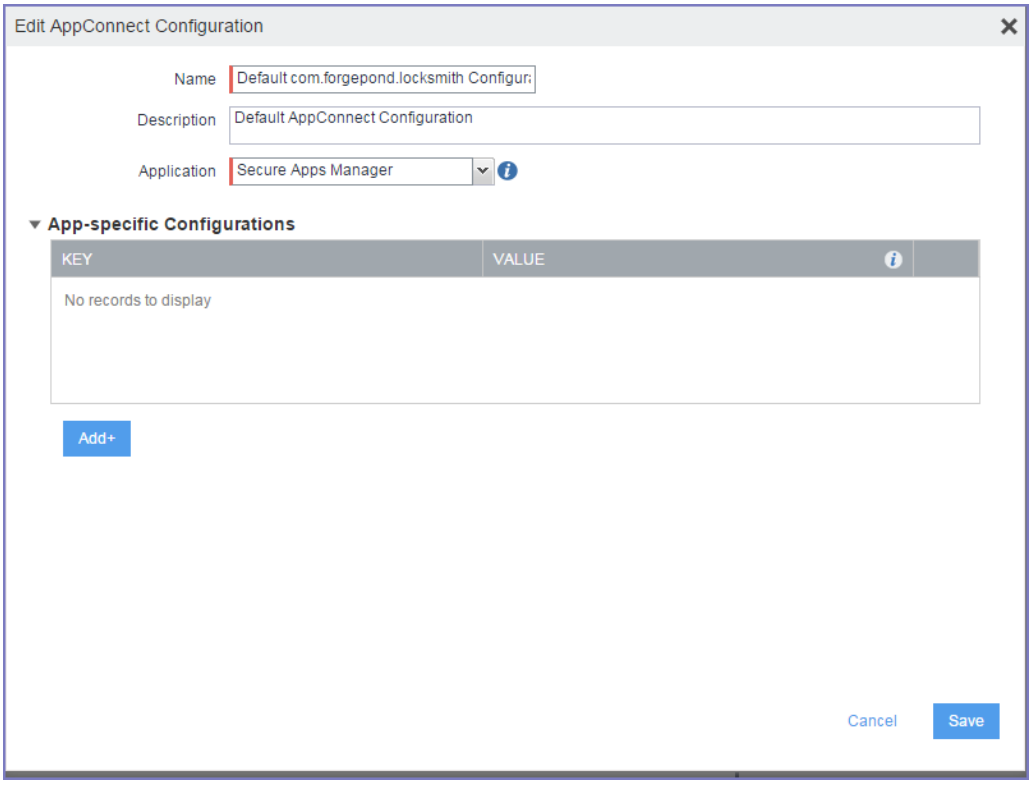
You specify that the installation order for secure apps using the AppConnect app configuration of the Secure Apps Manager in MobileIron Core. You create a special key-value pair that lists the secure apps' package IDs in the required installation order.

Note The Following: :

- You only specify the order for secure apps that have dependencies on each other.
- No action is required for other secure apps. The other secure apps are installed last, in alphabetical order by application name.
- The Secure Apps Manager is always installed first.

Procedure

1. In the Admin Portal, go to **Policies & Configs > Configurations**.
2. Select the AppConnect app configuration (**Setting Type** is **APPCONFIG**) for the Secure Apps Manager.
3. Click the **Edit** icon.



4. Click **Add+.**
5. For the **Key**, enter **AC_APP_INSTALL_ORDER.**
6. For the **Value**, enter a list of app package IDs, separated by semi-colons, in the required installation order. For example:
   `forgepond.com.acmesecureapps.financetracker;forgepond.com.acmesecureapps.saleslogger`
   The financetracker app will be installed first, followed by saleslogger.
   After the listed apps are installed, all other secure apps will be installed in alphabetical order by application name, as it appears in the MobileIron Core App Catalog in the **App Name** column. For example:
   - Divide PIM
   - File Manager
   - ThinkFree Office Viewer
7. Click **Save**.

Uninstall order when you specify installation order of secure apps

If you change the labels for the device or apps so that the apps no longer apply to the device, the apps are uninstalled in the following order:

1. Apps that are not part of the AC_APP_INSTALL_ORDER configuration are uninstalled, in reverse alphabetical order.
2. Apps that are part of the AC_APP_INSTALL_ORDER configuration are uninstalled, in the reverse order than they are listed.

Upgrading apps when you specify the installation order of secure apps

If you add new versions of secure apps to the App Catalog, the apps are upgraded on the device in the following order:

1. New versions of apps that are part of the AC_APP_INSTALL_ORDER configuration are upgraded in the order they are listed.
2. New versions of apps that are not part of the AC_APP_INSTALL_ORDER configuration are upgraded in alphabetical order.

Related topics

App management action workflows

Android app versions and device counts

To see all the versions of an app that are installed throughout all your users' devices, go to the **Apps > Installed Apps** page and select the **Details View**. The **App Version** column displays the version number, and the **Devices Installed** column shows the number of devices associated to that version of the app.

On the **Apps > Apps Catalog** page, the **Devices Installed** column displays the number of devices associated with the latest version of the app. To see collective information on all installed versions of an app, go to **Apps > Installed Apps** page.

Troubleshooting Android apps

Issue: A newly-added app does not display in Apps@Work on the device.

Troubleshooting:

1. Confirm that you have applied the app to a label to which the device has been added.
2. Confirm that the device meets the minimum OS requirement you specified for the app when you added the app.

3. If Mobile@Work app is running, select **Force Device Check-in** from the **Settings** menu (or **Connect Now** from the main menu in older versions).

Issue: A newly-added app does not display in the in-house apps list on the device.

Troubleshooting:

1. Confirm that you have applied the app to a label to which the device has been added.
2. Confirm that the device meets the minimum OS requirement you specified for the app when you added the app.
3. Confirm that the device has been configured to accept apps from outside the Google Play Store. (On the device, select **Settings > Applications > Unknown sources**).
4. If Mobile@Work app is running, select **Force Device Check-in** from the **Settings** menu (or **Connect Now** from the main menu in older versions).

Managing mobile apps for Android enterprise

You can deploy public and private Android enterprise apps to devices.

- About apps for Android enterprise
- Features specific to Android enterprise apps
- App configuration for Android enterprise apps
- Public and private Android enterprise app deployment
- Setting up Chrome with Android enterprise

Related topics

- App management action workflows

About apps for Android enterprise

Android enterprise apps are either *public apps* or *private apps*:

- Public apps are apps that are available to the general public in the Google Play Store.
- Private apps are apps developed for your organization in-house or by 3rd party developers that you distribute privately through Google Play. Only members of your domain have visibility into your private apps. Your private enterprise apps are available through the Google Play Store to registered users.

Besides making apps available in Google Play, you can make public and private (in-house) apps available for download to Android enterprise devices in the App Catalog on MobileIron Core. They can be installed on user devices and supported for Work Managed Device and Managed Device with Profile modes.You do this by selecting the **Install this app for Android enterprise** check box in the app details.

NOTE: You can select the **Install this app for Android enterprise** check box only if you are a global administrator which is an administrator assigned to the global space.

MobileIron Core supports various features that are specific to Android enterprise apps. You specify your choices for these features when you add an Android enterprise app to the App Catalog. Otherwise, working with the Android enterprise apps in the App Catalog is the same as for any other platform:

- you can mark an app as Featured
- you can assign an app to one or more Categories
- you must apply an app to a label to make it available to users.

You can change the "**Install this app for Android enterprise**" setting for each app in the app's details, on the App Catalog page, at any time.

NOTE: An app designated as available to Android enterprise devices can also be available to all Android devices. The app will install appropriately on Work profiles or non-Android enterprise devices.

Features specific to Android enterprise apps

MobileIron Core supports the following features for Android enterprise apps, on all Android enterprise modes. You set these features when you add the app to the MobileIron Core App Catalog, or later edit it.

- **Install this app for Android enterprise**: Selecting this check box is required for all Android enterprise apps. For In-house apps, further options for configuring Android enterprise display.
- **Silent Install for work managed devices**: (Applicable only to in-house apps) When you select this feature, the Android enterprise app is silently installed on devices with a work profile. This is selected by default.
- **Auto Update for this App**: (Applicable only to public and private apps) When you select this feature, the app is automatically updated on users' devices whenever a new version of the app is available on Google Play.

  If you select auto update, but the app fails to update on a user's device (for example, if the device has an incompatible Android version), then the app may attempt to update repeatedly. The workaround is to deselect **Auto Update this App** for that app.

  If you do not select auto update, the Android enterprise will still be updated if the app is updated on the personal side of the device.
- **Silent install for Mandatory Apps**: (Applicable only to public and private apps) Select this check box to silently install the app upon device check-in. De-selected means the device user will need to manually install the app.
- **Block Widget on Home Screen**: If selected, the app cannot place widgets on the home screen on work profile devices. For example, calendar apps are not permitted to place calendar widgets on the home screen.
- **Block Uninstall**: Select this feature to prevent the device user from uninstalling the app.
- **Quarantine app when device is quarantined**: Selected by default, this enables configured compliance actions to hide the app if a policy violation results in a quarantined device. This is a required selection for Work Profile mode, Work Managed Device mode and Managed Device with Work Profile mode.

  A second step is required to enable this feature: configure a corresponding compliance action and security policy with that compliance action selected. Once the device is no longer quarantined, the app can be used again. If this option is deselected, the app is available for usage, even when the device is quarantined.
- **Configure third-party app runtime permissions**Select this check box to modify runtime permissions for other apps.
  - Applicable to public / private apps on Work Managed Device mode on Android 8.0 through the most recently released version as supported by MobileIron.

- Applicable to in-house apps and public / private apps on Managed Device with Work Profile (COPE) on Android devices versions 8-10.
- Applicable to only public / private apps on all managed Work Profiles, including Work Profiles on Company Owned Devices Android versions 11 through the latest version as supported by MobileIron.

- **Hide and suspend third-party apps**: Select this check box to allow this app to hide / unhide, suspend, and remove suspension for other apps.
  - Applicable to in-house and public / private apps for managed devices and Managed Devices with Work Profile (COPE) starting from Android 8.
  - Applicable to public / private apps on managed profiles.
  - Applicable to public / private apps on Work profiles Company Owned Devices starting from Android 11.
- **Manage certificates**: Select this check box to allow this app to have access to certificate APIs on the device.
  - Applicable to in-house and public / private apps for managed devices and Managed Devices with Work Profile (COPE) starting from Android 8.
  - Applicable to public / private apps on managed profiles.
  - Applicable to public / private apps on Work Profile for Company Owned Device modes starting from Android 11.

Note The Following:

- run-time permission settings are supported only on Android 6.0 through the most recently released version as supported by MobileIron.
- If an app version has new permissions that you have not yet accepted on behalf of users, an icon appears in the **New Permissions** column on the App Catalog page. Until you accept new app permissions on behalf of users, new app installs for newly registered devices and app updates for currently registered devices will not proceed.
- To assign an app as a device owner silent in-house app, you must select both the **Install this app for Android enterprise** and **Silent install for Mandatory Apps** check boxes. (The Mobile@Work client does not consider "Mandatory" and "Silent install" options as selections for the device owner silent in-house app.)

- App configuration for Android enterprise apps allows you to provide configurable options to apps. Details are in App configuration for Android enterprise apps.

Related topics

- “Enabling run-time permissions for Android enterprise apps” in *MobileIron Core Device Management Guide for Android and Android enterprise Devices*
- App configuration for Android enterprise apps

App configuration for Android enterprise apps

App configurations (also referred to as app restrictions) are key-value pair settings that are provided by the app developer. When you select the **Install this app for Android enterprise** check box when adding a public app, the **Configuration Choices** section appears in the app wizard. Refer to the app's documentation and help hints for information on its configuration settings. These settings allow you to configure the app, without involving the device user.

MobileIron Core supports multiple bundle definitions in a bundle array for apps that have the capability to use this feature. For example a VPN app may support multiple VPN configurations by clicking the **Add New Configuration** button and entering the Profile Name and Server for a specific VPN and optionally specify your web log on credentials.

When using Mobile@Work 9.6 through the most recently version as supported by MobileIron, MobileIron Core delivers app configurations using Google Play. Therefore, the app and its app configurations are installed at the same time on the device, avoiding the potential issue of device users launching the app before the app configurations are received.

Creating multiple app configurations

Core allows you to create multiple app configurations per app:

- The default app configuration for the app is applied to devices with the same label that you applied to the app.
- Any additional app configuration that you can create is applied to devices with the labels you specify.

Using multiple app configurations is useful when sets of users of the app require different configuration values. For example, consider a Human Resources app that users throughout the United States use. However, you want the app to connect to a different server depending on a user's region:

- Users in the Eastern region must connect to a server in the east.
- Users in the Western region must connect to a server in the west.
- Users in the Northern and Southern regions connect to a server in St. Louis.

Therefore, do the following:

- Label the app with the Human Resources label.
- Create an app configuration that specifies the server in the east, and label the app configuration with the Eastern Region label.
- Create an app configuration that specifies the server in the west, and label the app configuration with the Western Region label.
- In the default configuration, specify the server in St. Louis. Users who do not have the Eastern Region label or the Western Region label will use this server.

Priorities of app configurations

Each app configuration you create has a priority. The highest priority has the value 1 and appears at the top of the list of configuration choices. The default configuration always has the lowest priority and appears at the bottom of the list. Core assigns a device the app configuration with the highest priority that has a label that matches a label on the device.

You can change the priorities of app configurations by dragging and dropping them in the table of configuration choices for the app.

Substitution variables for configuring Android enterprise apps

Substitution variables can be used for configuring values from LDAP or the MobileIron Core devices database, such as $EMAIL$ for the email address. You can prevent deleted default field values from repopulating when editing app configurations by entering the substitution variable $NULL$ for those values.

You may use the following variables when configuring any Android enterprise app:

```
$USERID$
$EMAIL$
$PASSWORD$
$FIRST_NAME$
$LAST_NAME$
$DISPLAY_NAME$
$USER_DN$
$USER_UPN$
$USER_LOCALE$
$DEVICE_UUID$
$DEVICE_UUID_NO_DASHES$
$DEVICE_IMSI$
$DEVICE_IMEI$
$DEVICE_SN$
$DEVICE_ID$
$DEVICE_MAC$
$DEVICE_CLIENT_ID$
$USER_CUSTOM1$
$USER_CUSTOM2$
$USER_CUSTOM3$
$USER_CUSTOM4$
$MI_APPSTORE_URL$
$REALM$
$TIMESTAMP_MS$
$NULL$
$GOOGLE_AUTOGEN_PASSWORD$
```

NOTE: Enable Google Apps Integration for the substitution to work properly.

Substitution variable for certificate aliases in Android enterprise apps

Some Android enterprise apps, including Gmail, MobileIron Tunnel for Android enterprise, and Pulse Secure, use certificates generated based on a certificate enrollment setting. These apps accept certificate aliases in the app configuration. The substitution variable to provide a certificate alias is:

`$CERT_ALIAS:<certificate enrollment setting name>$` where

`<certificate enrollmnent setting name>` is the name you gave to the certificate enrollment setting.

To use a certificate with apps, in the Core Admin Portal:

1. Go to **Policies & Configs > Configurations**
2. Locate your certificate enrollment setting. Note its name. You will need the name for the alias variable.
   Note: The certificate enrollment setting must be created before continuing with these steps.
3. Ensure the certificate enrollment setting is assigned to a label that is also used for distributing the apps that require the certificate.
4. Go to **Apps > App Catalog**.
5. Edit the app by clicking the app name, then clicking **Edit**.
6. Ensure that the Android enterprise check box **Install this app for Android enterprise** is selected.
7. In the **Configurations** section, type in the certificate alias in the field that requires it:
   `$CERT_ALIAS:<certificate enrollment setting name>$`
8. Click **Finish** to save your changes.

Note The Following:

- Certificate aliases are not supported for user-provided certificate enrollment settings. For more information about Certificate Enrollment Settings, see “Certificate Enrollment Settings” in *MobileIron Core Device Management Guide for Android and Android enterprise Devices*.
- For identity certs applied to Android devices, Mobile@Work will require a passcode for the device or work profile, if the user has not already created one.
- On Android 6.0 devices or higher, and with Mobile@Work 9.6, identity certs will be automatically assigned for apps. Users will not be prompted to select a certificate.

Public and private Android enterprise app deployment

MobileIron Core provides administrators with the following options for deploying apps to Android enterprise device users.

- **Public apps**: These apps are developed outside of your organization and are available to Android enterprise device users from the public Google Play store. They are hosted by Google, but administrators can manage public apps using Core.
- **Private apps**:These apps are available only to your organization. Private apps are hosted by Google and available from the Google Play Apps Catalog. They are hosted by Google, but administrators can manage private apps using Core.

  These apps are available to only users of your domain and can be available in a non-English language that is supported by MobileIron. The following private apps are described below.

  - **Private in-house apps**: These apps are developed in-house, available only to your organization and can be available in a non-English language that is supported by MobileIron. Private in-house apps are more secure because they are hosted by Core (not Google), but are available from the Google Play

Apps Catalog. The apps generate an APK definition file you upload to the Google Play Developer Console to use for installing the apps. These apps not available through Apps@Work; see Distributing your enterprise apps in the Google Play App catalog or in Apps@Work for details.

NOTE: When the API connection in Core's Access Control List is enabled, device attempts to download private self-hosted apps from an IP address range that is not listed in that Access Control List will be rejected. This is expected behavior. In order for devices to download private self-hosted apps, devices must have an IP address that is on Core's Access Control List.

To deploy apps see:

- Deploying public Android enterprise apps
- Deploying private Android enterprise apps
- Distributing your enterprise apps in the Google Play App catalog or in Apps@Work

Related topics

App management action workflows

Deploying public Android enterprise apps

A public app is available in the public Google Play store. You can add public apps to the App Catalog using the app wizard that helps you through all the options and configurations. You can also add public apps using the Google Play iFrame. See Adding an Android enterprise public app using the app wizard in the Core Admin Portal.

Adding an Android enterprise public app using the app wizard in the Core Admin Portal

Before you begin

Enable Android enterprise in MobileIron Core. See "Enabling Android enterprise" in the *MobileIron Core Device Management Guide for Android and Android enterprise Devices*.

When adding the app, the app wizard guides you through all options and configurations of public and private apps on Android enterprise. In-house and self-hosted apps are applicable to Android enterprise, but are not configured using the app wizard.

NOTE: Once Android enterprise is installed, the Quick Import option for Google Play is disabled.

Procedure

1. In the Admin Portal, go to **Services > Google**.
2. Use the browse button to navigate to the JSON file you downloaded as part of the Android enterprise enrollment and then click Connect.
3. A confirmation displays stating that you have been enrolled in Google Services.

4. Go to **Apps > App Catalog**.
5. Click **Add+**.
6. Click **Google Play**. The Google Play store opens below displaying only Android enterprise apps.
7. The pop-out sidebar displays three options:
   - Search Play store – search for a specific app in the Google Play store. Only public apps and enterprise domain / package name apps can be searched upon. Once a private app has been uploaded, you can search for the private app.
   - Private apps – allows you to import private Android apps into Google Play for Android device users to download and use.
   - Web app – allows you to create a web app.

   NOTE: This flow is generated by Google Play and may change in the future as Google adds new features.
8. In the Search for app field or clicking on the sidebar and selecting **Search Play store**, enter the app name and click **Search**. Google Play Store displays app icons in the search results.
9. Click on an app's icon to select it.
10. You will need to approve the app to be part of the Android enterprise app collection for device users' consumption. Click the **Approve** button. The app's Approval Settings and Notifications dialog box opens.
11. Every app requires permissions to access specific aspects of an Android phone, for example, contacts. As an administrator, you will need to review these permissions because you will be accepting or revoking them on behalf of your organization. Select one of the options:
    - **Keep approved when app requests new permissions** – permissions can change due to app updates. If this option is selected, it means the device user may not know about the access permission changes.
    - **Revoke app approval when this app requests new permissions** – If this option is selected, when a new update has changed its access permissions, the device user will be notified of the access permissions when the app is updated. The device user must accept the new permissions otherwise the app will be disabled for that user.
12. Click the Notifications tab.
13. Enter the email addresses of people to be notified if an app has been updated.
14. After you **Save**, a confirmation email is sent to the listed person(s). The button in the confirmation email needs to be clicked to activate the email subscription. People successfully subscribed will be listed in the Notification tab of the app.

    NOTE: If you selected the "Keep approved when app requests new permissions" option and no email is entered into the Notification tab, all updates are silent.
15. Click **Save**.
16. The app information displays with a check mark next to "Approved". If you want to review the access

permissions or notifications, click the **Approval Preferences** button.

17. Click **Select**.
18. Click **Next**. Now that you have set the access permissions to the app, you can finish configuring the app. Configurations are determined by the app developer and are key-value pairs unique to each app. Fill out the configurations sections as needed. If needed, refer to the app's documentation.
19. Use the following guidelines to complete the page.

Item	Description
Application Name	Displays the app name defined by the app developer. This is the name that displays to device users. This field is not editable.
Description	The app description as retrieved from Google Play displays. You can edit the description. Users will see this description in Apps@Work on their devices.
Category	Select one or more categories to display this app in a category tab in Apps@Work or add a new category. a. Click **Add New Category** to define new categories. b. Enter a category **Name** (up to 64 characters). c. Enter a **Description** (up to 255 characters). d. In the **Category Icon** section, click the **Replace Icon** button. e. Browse and select an icon that will represent this Category. f. Click **Save**.

20. Click **Next**.
21. Use the following guidelines to complete the page.

Section	Item	Description
Apps@Work Catalog	Feature this App in the Apps@Work catalog	If check box is selected, this app appears in the Featured Apps tab in Apps@Work.
	Featured Banner	Selecting the check box will display this app as part of the top banner on the Apps@Work Home page on end users' devices. The latest five apps will be picked to be part of Apps@Work Home page.
Per App VPN Settings	Per App VPN by Label Only	Select this check box to require the Per App VPN configuration to be assigned to a label that matches

Section	Item	Description
		the device. If there is no associated label between the VPN configuration and the device, Per App VPN will not be installed on the device. Clear this check box to assign the per App VPN based on the selections in the **Per App VPN** field, ignoring labels. NOTE: Per app VPN is not supported for MAM-only Android devices.
	License Required	The **Selected VPNs** column lists the VPN configuration that may be installed on the device, in priority order: • If **Per App VPN by Label Only** is selected, then the VPN configuration must be assigned to a label matching the device in order to be installed. The first VPN in the list that is also assigned to a label associated with the device has the highest priority. • If **Per App VPN by Label Only** is not selected, then the VPN configurations listed are in priority order and do not need to be assigned to a label matching the device. To populate the **Selected VPNs** column, select the VPN configuration you created for per app VPN in the **All VPNs** column, and click the right arrow. You can select multiple per app VPN settings. To reorder the per app VPN configurations in the **Selected VPNs** column, drag the configuration names to the correct positions in the list. See "Managing VPN settings" in the *MobileIron Core Device Management Guide* for information on creating a per app VPN. NOTE: Per app VPN is not supported for MAM-only Android devices.
Android Enterprise (All Modes)	Install this app for Android enterprise	You must be a Global Space administrator to use this setting. Select to enable public and private apps available to device users for download to Android devices. You can change the "Install this app for Android enterprise" setting for each app in the app's

Section	Item	Description
		details page at any time.
	Silent install for work managed devices	(Applicable only to in-house apps) When you select this feature, the Android enterprise app is silently installed on devices with a work profile. This is selected by default.
	Auto Update this App	You must be a Global Space administrator to use this setting. Select this check box to automatically update this app on devices when a new version is available on Google Play. If this check box is de-selected, users will not receive automatic app updates. Note that when an app in a Work Profile is updated, the same app is updated in the personal profile.
	Silent install for Mandatory Apps	This feature only applies to devices that support silent installation. • Clearing the check box means the device user will need to manually install the app. • Selecting the check box will install the app silently. The app is installed when the device checks in with Core. User action is not required. For more information, see Silent install and uninstall of mandatory apps. NOTE: Silent install is not supported for MAM-only Android devices.
	Block Widget on Home Screen	Applicable only to apps installed in the Managed profile.
	Block Uninstall	Select this feature to prevents the device user from uninstalling the app.
	Quarantine app when device is quarantined	Required for Work Profile mode, Work Managed Device mode, and Managed Device with Work Profile mode. Selected by default, this enables configured compliance actions to hide the app if a policy violation results in a quarantined device. This is a required selection for Work Profile mode, Work Managed Device mode and Managed Device with Work Profile

Section	Item	Description
		mode. A second step is required to enable this feature: configure a corresponding compliance action and security policy with that compliance action selected. Once the device is no longer quarantined, the app can be used again. If this option is deselected, the app is available for usage, even when the device is quarantined. Note The Following: • If you change the setting after the app is added, the changed setting will be applied to the app.
Delegated Permissions	Configure third-party app runtime permissions	Select this check box to modify runtime permissions for other apps. • Applicable to public / private apps on Work Managed Device mode on Android 8.0 through the most recently released version as supported by MobileIron. • Applicable to in-house apps and public / private apps on Managed Device with Work Profile (COPE) mode on Android devices versions 8-10. • Applicable to only public / private apps on all managed work Profiles, including Work Profiles on Company Owned Devices mode Android versions 11 through the latest version as supported by MobileIron.
	Hide and suspend third-party apps	Select this check box to delegate third-parties to have permission to hide and suspend the selected app. • Applicable to in-house and public / private apps for Work Managed Device mode and Managed Devices with Work Profile (COPE) mode starting from Android 8. • Applicable to public / private apps on managed profiles. • Applicable to public / private apps on Work Profile for Company Owned Device mode starting from Android 11.
	Manage certificates	Select this check box to delegate permission for managing certificates.

Section	Item	Description
		• Applicable to in-house and public / private apps for managed devices and Managed Devices with Work Profile (COPE) mode starting from Android 8. • Applicable to public / private apps on managed profiles. • Applicable to public / private apps on Work Profile for Company Owned Device mode starting from Android 11.

22. The Configurations Choices section relates to configuring the app you are adding. You will need to refer to the app's documentation for how to proceed with these configurations. For example, MobileIron Core supports the Knox Service Plugin app. In order to enter the configurations for this app, you will need to access the Knox Developer documentation for Knox Service Plugin at https://docs.samsungknox.com/dev/knox-service-plugin/index.htm?Highlight=KSP. A login may be required to access app documentation.
23. Click **Finish**.
24. In the App Catalog, select the newly-added app.
25. Click **Actions > Apply to Label**. Select the appropriate labels to make the app available to device users. You can edit the app's settings at any time. Select the app in the App Catalog, and click **Edit**.

All apps that are available to be installed for Android enterprise have the "suitcase" badge on their icon. These apps can also be installed on non-Android enterprise devices. For more information about labels for Android enterprise, see Distributing alternate Release Tracks for Android enterprise apps.

Note The Following:

- You can edit the app's settings at any time. Select the app in the App Catalog, and click **Edit**.
- The metadata and reviews for an app selected for installation from Google Play may not be displayed depending on the configuration of the customers firewall.

Related topics

- Features specific to Android enterprise apps
- App configuration for Android enterprise apps
- Setting up Chrome with Android enterprise
- "Setting up Gmail with Android enterprise" in the *MobileIron Core Device Management Guide for Android and Android enterprise Devices*

Deploying private Android enterprise apps

The high-level steps to deploy a private Android enterprise app are:

Publishing your private app on Google Play to your organization only

Before you begin

These steps are performed on Google's websites.

1. If you are doing icon customization and plan on sharing the private app with other UEMs, your Google Enterprise account must be registered as a Google developer.

   NOTE: If you are using iFrame option via Cloud / Core, you can import private apps without registering the Enterprise account as a developer.
2. Follow Google's instructions to publish the app on Google Play.
3. To make the app available privately to other UEMs or organizations, please refer to this KB article: How to share private Android Enterprise Apps with other UEMs.

Adding your Android enterprise private app using the app wizard in the Core Admin Portal

This procedure is how you add a private Android enterprise app to the MobileIron Core App Catalog. In-house apps are supported with Android enterprise but you cannot configure them using the app wizard.

If you are adding a new version of an existing app, see Adding new versions of an existing Android enterprise app.

Procedure

1. In the MobileIron Core Admin Portal, go to **Apps > App Catalog** .
2. Click **Add+**.

   Click **Google Play**. The app icons for the private apps you published to Google Play display.

   NOTE: If you need to update the email address associated with the app, click **Update**.
3. Select the desired app and then click **Next**.
4. The Choose page displays the private app's title and APK file name.
5. Click **Select** and then click **Next**.
6. The Describe page displays. Use the following guidelines to complete the page.

Item	Description
Package Name	You must provide the app's package name. MobileIron Core can upload an Android Google Play Store app that has the same package name as a public app, such as com.mobileiron.phoneatwork, that is already loaded on Core. This feature is always on and does not require any configuration in the user interface.
Application Name	Displays the app name defined by the app developer. This is the name that displays to device users. This field is not editable.
Min OS Version	The minimum OS version as retrieved from Google Play displays. Devices that don't have the minimum OS version installed will not be able to install the app.
Description	The app description as retrieved from Google Play displays. You can edit the description. Users will see this description in Apps@Work on their devices.
Category	Select one or more categories to display this app in a category tab in Apps@Work or add a new category. a. Click **Add New Category** to define new categories. b. Enter a category **Name** (up to 64 characters). c. Enter a **Description** (up to 255 characters). d. In the **Category Icon** section, click the **Replace Icon** button. e. Browse and select an icon that will represent this Category. f. Click **Save**.

7. Click **Next**.
8. The App Store page displays. Use the following guidelines to complete the page.

Section	Item	Description
Apps@Work Catalog	Feature this App in the Apps@Work catalog	If check box is selected, this app appears in the Featured Apps tab in Apps@Work.
	Featured Banner	Selecting the check box will display this app as part of the top banner on the Apps@Work Home page on end users' devices. The latest five apps will be picked to be part of Apps@Work Home page.
Icon and Screenshots	App Icon	NOTE: Icon and Screenshots appear when editing an app entry. • The icon retrieved from Google Play displays. • To replace the icon, click **Replace Icon** button. Select the icon to represent this app. The file must be no larger than 1024 x 1024 pixels and in JPG, PNG, or GIF format. We recommend PNG for best resizing results. Icon height and width must be equal.
	Screenshots	NOTE: Icon and Screenshots appear when editing an app entry. • The screenshots retrieved from Google Play are displayed. • Click **Upload** to select and upload optional screenshot files in PNG, GIF, or JPG formats. The supported dimensions are 480x800 pixels and 480x854 pixels. We recommend PNG for best resizing. • To delete a screenshot, click **Remove** under the screenshot.
Android Enterprise (All Modes)	Install this app for Android enterprise	Selecting enables public and private apps available to device users for download to Android devices. You can change the “Install this app for Android enterprise” setting for each app in the app’s details page at any time.

9. Click **Next**. The App Configuration page displays.
10. Use the following guidelines to complete the page.

Section	Item	Description
Per App VPN Settings	Per App VPN by Label Only	Select this check box to require the Per App VPN configuration to be assigned to a label that matches the device. If there is no associated label between the VPN configuration and the device, Per App VPN will not be installed on the device. Clear this check box to assign the per App VPN based on the selections in the **Per App VPN** field, ignoring labels. NOTE: Per app VPN is not supported for MAM-only Android devices.
	License Required	The **Selected VPNs** column lists the VPN configuration that may be installed on the device, in priority order: • If **Per App VPN by Label Only** is selected, then the VPN configuration must be assigned to a label matching the device in order to be installed. The first VPN in the list that is also assigned to a label associated with the device has the highest priority. • If **Per App VPN by Label Only** is not selected, then the VPN configurations listed are in priority order and do not need to be assigned to a label matching the device. To populate the **Selected VPNs** column, select the VPN configuration you created for per app VPN in the **All VPNs** column, and click the right arrow. You can select multiple per app VPN settings. To reorder the per app VPN configurations in the **Selected VPNs** column, drag the configuration names to the correct positions in the list.

Section	Item	Description
		See "Managing VPN settings" in the *MobileIron Core Device Management Guide* for information on creating a per app VPN. NOTE: Per app VPN is not supported for MAM-only Android devices.
Android Enterprise (All Modes)	Install this app for Android enterprise	You must be a Global Space administrator to use this setting. Selecting enables public and private apps available to device users for download to Android devices. You can change the "Install this app for Android enterprise" setting for each app in the app's details page at any time.
	Silent install for work managed devices	This feature only applies to devices that support silent installation. • Clearing the check box means the device user will need to manually install the app. • Selecting the check box will install the app silently. The app is installed when the device checks in with Core. User action is not required. For more information, see Silent install and uninstall of mandatory apps. NOTE: Silent install is not supported for MAM-only Android devices.
	Block Widget on Home Screen	If selected, the app cannot place widgets on the home screen on work profile devices. For example, calendar apps are not permitted to place calendar widgets on the home screen.
	Block Uninstall	Select this feature to prevent the device user from uninstalling the app.
	Quarantine app when	(Required for Work Profile mode,

Section	Item	Description
	device is quarantined	Work Managed Device mode, and Managed Device with Work Profile mode.) Selected by default, this enables configured compliance actions to hide the app if a policy violation results in a quarantined device. This is a required selection for Work Profile mode, Work Managed Device mode and Managed Device with Work Profile mode. A second step is required to enable this feature: configure a corresponding compliance action and security policy with that compliance action selected. Once the device is no longer quarantined, the app can be used again. If this option is deselected, the app is available for usage, even when the device is quarantined. Note The Following: • If you change the setting after the app is added, the changed setting will be applied to the app.
Delegated Permissions	Configure third-party app runtime permissions	Select this check box to grant delegation permissions to third parties. • Applicable to public / private apps on Work Managed Device mode on Android 8.0 through the most recently released version as supported by MobileIron. • Applicable to in-house apps and public / private apps on Managed Device with Work Profile (COPE) mode on Android devices versions 8-10. • Applicable to only public / private apps on all managed work profiles, including Work Profiles on Company Owned Devices Android versions 11 through the

Section	Item	Description
		latest version as supported by MobileIron.
	Hide and suspend third-party apps	Select this check box to delegate third-parties to have permission to hide and suspend the selected app. • Applicable to in-house and public / private apps for managed devices and Managed Devices with Work Profile (COPE) mode starting from Android 8. • Applicable to public / private apps on managed profiles. • Applicable to public / private apps on Work Profile for Company Owned Device mode starting from Android 11.
	Manage certificates	Select this check box to delegate permission for managing certificates. • Applicable to in-house and public / private apps for managed devices and Managed Devices with Work Profile (COPE) mode starting from Android 8. • Applicable to public / private apps on managed profiles. • Applicable to public / private apps on Work Profile for Company Owned Device mode starting from Android 11.

11. Click **Finish**.
12. Select the app in the App Catalog.
13. Click **Actions > Apply to Label,** and select the appropriate labels to make this app available to device users.

    NOTE: You can edit the app's settings at any time. Select the app in the App Catalog, and click **Edit**.

Manually provide an app's package name

You can manually provide the package name of an Android app along with the app details.

1. In the MobileIron Core Admin Portal, go to **Apps > App Catalog** .
2. Click **Add+**.
3. Click **Google Play**. The app icons for the private apps you published to Google Play display.
4. Scroll down to the bottom of the page and select the check box for **Skip this step and manually provide Bundle ID and all app details.**
5. Click **Next**. The App Configuration page displays.
6. Complete the remaining steps of the app wizard (see step 6 at Deploying private Android enterprise apps
7. Click **Finish**.
8. Select the app in the App Catalog.
9. Click **Actions > Apply to Label,** and select the appropriate labels to make this app available to device users.

   NOTE: You can edit the app's settings at any time. Select the app in the App Catalog, and click **Edit**.

Related topics

- Features specific to Android enterprise apps
- App configuration for Android enterprise apps

Deploying a self-hosted app

Self-hosted apps allow administrators to publish in-house app entries in the Google Play Apps Catalog without uploading binaries to Google. For security reasons, self-hosted apps are hosted by Core and not Google, however they are still available in the Google Play Apps Catalog. Self-hosted apps require the definition of APK location to be uploaded to Google Play. Revisions are required to be published to Google Play, which points only to the latest version on Core.

Silent install of the APK is supported only on work-managed devices. You can manually install self-hosted apps from Google Play. You can use this feature to block or allow users to show in-house app widgets on the home screen inside the Work Profile. By enabling the "Block Widget on Home Screen" and "Block Uninstall" options, you can also block or allow users from uninstalling the app. This feature applies to managed devices only.

NOTE: These apps are not available for Android enterprise devices users to install from Apps@Work.

Procedure

If you are adding a new version of an existing app, see Adding new versions of an existing Android enterprise app.

1. In the Admin Portal, select **Apps > App Catalog**.
2. Upload a new APK that becomes an in-house app.
   If you are adding a new self-hosting app:

a. Click **Add+ > In-House > Browse**.

b. Locate and select the app, then click **Next**.

c. Skip to the next step.

If you want to redefine an existing app:

a. Select the app and click **Edit**.

b. Continue with the next step.

3. Scroll down to the **ANDROID ENTERPRISE (ALL MODES)** section.
4. Select the **Install this app for Android enterprise** check box.
5. Click the **Download APK Definition** file link. The APK definition file downloads automatically.
6. Open a new browser window and log into the Google Play Developer Console site.
7. Follow Google's steps on publishing.
   - Under Distribution > Managed Google Play, make sure you have the Provately target this app to a list of organizations check box selected. Click Choose Organizations.
   - When uploading the APK file, be sure to select the "I am uploading a configuration for an APK hosted outside of Google Play" check box.
   - Go to Services > API > Licensing & in-app billing section and copy the the license key.
8. Return to the Core App Catalog broswer window and paste the key in the **App License** box provided.

   NOTE: Every version of that app uses the same License Key.
9. Select one or more of the following check boxes:
   - **Silent install for work managed devices**
   - **Block Widget on Home Screen**
   - **Block Uninstall**
10. Click **Save**.
11. Select the app in the App Catalog.
12. Click **Actions > Apply to Label.**
13. Select the appropriate labels to make this app available to device users.

Adding new versions of an existing Android enterprise app

When uploading a newer version of an app, an extra page opens to allow you to select whether to keep the app's old version information or to adopt the information from the app's new version. This feature is applicable to Android enterprise in-house / private / self-hosted apps.

Procedure

1. In the App Catalog, click the **Add+** button.
   The Add App Wizard opens.
2. Click **In-House**.

3. Click **Browse** and navigate to the in-house Android or Android enterprise app you want to upload.
4. Click **Next**.
   The An earlier version of this App exists page opens.
5. Select an option:
   - **Another version of this App was previously uploaded. Reuse its description, icon and screenshot.** If the Description, Icon or Screenshot fields of the new app are empty, then the system will populate those fields with information from the previous app version (default).
   - **Upload a new description, icon or screen shot.** Information related to the Description, Icon or Screenshot fields of the new App will be utilized. If those fields are empty, nothing will be copied from the previous app version.
6. Click **Next** and finish configuring the new version of your app (see Adding your Android enterprise private app using the app wizard in the Core Admin Portal.)
   Once finished, the new version displays in the App Catalog.

Distributing your enterprise apps in the Google Play App catalog or in Apps@Work

By default, Android enterprise apps are distributed from a managed Google Play. However, you can opt to distribute the apps from Apps@Work.

Use these steps to set up your distribution choice for your enterprise apps:

1. Make sure your device is set up for Android enterprise. See “Enabling Android enterprise for your enterprise” in the MobileIron Core Device Management Guide for Android and Android enterprise Devices.
2. On the Admin Portal, in **Services > Google**, in **Enterprise Apps Distribution**, choose either **Google Play** or **Apps@Work**.
3. If you change the setting, click **Apply**.
4. If you selected **Google Play**, in **Google Play App Catalog** section:
   Select **Yes** to use a layout based on the characteristics of apps in this instance of MobileIron Core. The apps are presented in Google Play using the categories and featured apps as you defined for each app in the App Catalog. Apps added recently to the App Catalog are presented in a “What’s New” list.
   Select **No** (the default) to use a basic layout in Google Play. In this layout, the apps are presented in alphabetical order in a single list.

   Note The Following:
   - If more than one Core instance is publishing with Google Play, you will be sending redundant (possibly conflicting) layouts to Google. This does NOT affect the distribution of apps, only the layout visible in Google Play.
   - The Google Play layout definition is based on the Android enterprise apps available on the Core that you marked as primary on help.mobileiron.com when setting up your Android enterprise enrollment. If you have multiple Cores that use the same enterprise account, the devices registered to users in each Core receive the same layout. This layout can be consistent only if one Core is set to publish the layout. If multiple Cores are marked as the primary Core, then they will attempt to publish the layout and cause the layout to become unstable.

NOTE: Updates to the Google Play App catalog may take several minutes to take effect.

Related topics

- Deploying public Android enterprise apps
- Deploying private Android enterprise apps

Distributing alternate Release Tracks for Android enterprise apps

For Android enterprise 10.4.0.0 through the latest version as supported by MobileIron, this feature works for private and public Android apps, and Android enterprise apps. Any public app that the app developer allowed Android enterprise access to their tracks will work. You can deploy numerous versions of private apps to allow rapid and flexible deployment of different builds of the same app to different groups.

In Core versions below 10.4.0.0, there were three static options (Alpha, Beta, Production) that you can select from in the list of releases (Track ID) defined by the developer who uploaded the application to Google Play. Upon upgrade to Core 10.4.0.0, Core supports as many tracks as the app devloper published and assigned to the enterprise. This list is dynamically retrieved from Google Play and displays in the release column of the Add to Label dialog box. Core uses the Track IDs to specify which track, but for administrators, Core displays the track aliases. As the list can include new and different Track aliases, during the upgrade to 10.4.0.0, Core will try to match existing Track IDs, but if there is no Track ID match, Core will assign the track to Production.

If a device is assigned to multiple Track IDs, all Track IDs will be sent to the device and Google will choose the highest available track to use. Since the tracks are set by label, it's possible for a device to belong to multiple labels getting multiple Track IDs for the same app.

Before you begin

- Select Android enterprise apps to be used in the Admin Portal.
- Identify one or more private apps administrators want to deploy to users within their organization.
- Set up separate labels to include alpha users and beta users.
- Verify that your in-house app developers have whitelisted the alpha and beta apps for distribution to your enterprise using the Google enterprise ID for Core as the target organization.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Select one of the Android enterprise-enabled apps you want to add to an alpha or beta label.
3. Click **Actions > Apply to label**.
4. Select one or more labels.
5. Go to the **Release** column and click inside the cell to enable the drop down option.
6. Select **Alpha**, **Beta**, **Production** (default) or an alternate option as per Google's dynamically-updated list.

NOTE: Core only displays the track aliases for the tracks that are possible for the app for that enterprise. It does not have to be Alpha or Beta.

7. Click **Apply**.

NOTE: At the next sync the specified track is downloaded to the designated devices. If multiple labels are applied to a device introduce conflicts, label priority applies the highest version in the following order: Alpha, Beta, then Production.

Related topics

- Deploying public Android enterprise apps
- Deploying private Android enterprise apps

Setting up Chrome with Android enterprise

You can deploy Google Chrome to Android enterprise devices if you have set up MobileIron Core for Android enterprise.

Add Chrome to the App Catalog on Core as you would any Android enterprise app. That is, in the Admin Portal, in **Apps > App Catalog**, add Gmail from Google Play. When adding it, be sure to select **Install this app for Android enterprise**.

When you add the Chrome app, its app configurations are displayed in the **Configuration Choices** section. Google documents these settings at https://www.chromium.org/administrators/policy-list-3.

NOTE: The value of the **ManagedBookmarks** configuration must be in JSON format. For example:

```
[{“toplevel_name”: “MobileIron bookmarks”}, {“url”: “http://mobileiron.com“, “name”:
“MobileIron”},
{“url”: “youtube.com”, “name”: “Youtube”}, {“name”: “Chrome links”,
“children”: [{“url”: “chromium.org”, “name”: “Chromium”},
{“url”: “dev.chromium.org”, “name”: “Chromium Developers”}]}]
```

Managing apps on Windows devices

This chapter provides topics on how to manage apps on Windows devices, including:

- Setting up certificate authentication
- Distributing apps for Windows 10 Desktop devices
- Distributing apps for Windows 8.1 Phone devices
- App inventory on Windows 10 desktop devices
- Application scheduling
- Restricting applications on Windows devices
- Working with apps
- Adding in-house apps to the App Catalog
- Adding third-party apps to the App Catalog
- Deploying apps
- Editing in-house app information
- Application dependency deployment
- Editing third-party app information
- Updating apps in the App Catalog
- Deleting apps from MobileIron Core

NOTE: MobileIron Core does not support Windows MAM-only devices.

Setting up certificate authentication

This section provides the required steps to set up a new dedicated local certification authority (local CA), provision its public certificate to Windows 10 devices (making it trusted), and configure certificate enrollment for Windows 10 devices. If Apps@Work finds a suitable device certificate to use for authentication, Apps@Work uses it instead of asking the user for a password.

Implement the work flow in the following order:

1. Add a new local certification authority
2. Create a label for all Windows 10 devices
3. Provision the CA certificate to all Windows 10 devices
4. Create a label for Windows 10 Desktop devices

5. Distribute device certificates to Windows 10 Desktop devices
6. Enable use of device certificates for Apps@Work authentication

NOTE: This cert is only used for Apps@Work and not for VPN, email, or any other profile. When the cert is used for Apps@Work the it is converted to a cert that can only be used with the app.

Add a new local certification authority

NOTE: This section supports a local CA. Other certification authorities such as Entrust, Microsoft NDES or Symantec Managed PKI are not supported.

To add a new local certification authority:

1. In the Admin Portal, go to **Services > Local CA**.
2. Select **Add > Generate Self-Signed Cert**.
3. Enter the following configuration:
   - **Local CA Name:** Contoso CA (we are using Contoso as an example in this documentation; replace Contoso with your company name)
   - **Key Type:** RSA
   - **Key Length:** 2048
   - **CSR Signature Algorithm:** SHA256
   - **Key Lifetime (in days):** 3650
   - **Issuer Name:** CN=Contoso CA
4. Click **Generate**.
5. Enter the following configuration:
   - **Hash Algorithm:** HA256
   - **Minimum Key Size Allowed:** 2048
   - **Key Lifetime (days):** 365
6. Keep other default values and click **Save**.
7. Click the **View Certificate** link.
8. Copy the base64-encoded public certificate (including the text `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----` delimiters).
9. Paste it to your text editor and save it to a file named *Contoso.cer*.
   You will use it in Provision the CA certificate to all Windows 10 devices.
10. Click **Close**.

Create a label for all Windows 10 devices

If you already have a label for all Windows 10 devices, skip this section.

To create a label for all Windows 10 devices:

1. In the Admin Portal, go to **Devices & Users > Labels**.
2. Click **Add Label**
3. Select or enter the following values:
   - Label name: *Windows 10*
   - Common fields: ***Platform Name***
   - Operator: *Equals*
4. Value: ***Windows 10***
5. Verify that the expression is valid (with a green check mark).
6. It should look like this: `"common.platform_name" = "Windows 10"`
7. Click **Save**.

Provision the CA certificate to all Windows 10 devices

After creating a new self-signed (untrusted) CA in Add a new local certification authority, you will provision its public certificate to all Windows 10 to make it trusted in this step. Without it the devices will not use the provisioned device certificates.

To provision the CA certificate to all Windows 10 devices:

1. In the Admin Portal, go to **Policies & Configs > Configurations**.
2. Click **Add New > Certificate**.
3. Enter name *Contoso CA*.
4. Click **Browse** next to **File Name**.
5. Click **Save > OK**.
6. Select the newly created **CERTIFICATE** setting and apply it to the *Windows 10* label you created earlier.
7. Click **OK** to confirm provisioning was successful.

Create a label for Windows 10 Desktop devices

If you already have a label for all Windows 10 Desktop devices, skip this section.

To create a label for Windows 10 Desktop devices:

1. In the Admin Portal, go to **Devices & Users > Labels**.
2. Click **Add Label**.
3. Select or enter the following values:
   - Label name: *Windows 10 Desktop*
   - Common fields: ***Platform Name***
   - Operator: *Equals*
   - Value: ***Windows 10***

- Phone: *False*

4. Verify that the expression is valid (with a green check mark).
   It should look like this: `"common.platform_name" starts with "Windows 10" AND "windows_phone.wp_phone" = false`
5. Click **Save**.

Distribute device certificates to Windows 10 Desktop devices

Now that the new certification authority is trusted, you can distribute device certificates to Windows 10 Desktop devices. Apps@Work for Windows 10 expects that the certificate subject is the device UUID. The device UUID value is also provisioned by MDM to Apps@Work to find the certificate.

To distribute device certificates to Windows 10 Desktop devices:

1. In the Admin Portal, go to **Policies & Configs > Configurations**.
2. Click **Add New > Certificate Enrollment > Local**.
3. Enter or select the following values for configuration:
   - **Name:** Contoso Windows Certificate Authentication
   - **Radio Button**: **Device Certificate**
   - **Local CAs:** Contoso CA
   - **Subject:** CN=$DEVICE_UUID$
   - **Key Usage:** Signing and Encryption (check both)
   - **Key Length:** 2048
   - **CSR Signature Algorithm:** SHA256
4. Click **Issue Test Certificate**.
5. Verify that the values in the test certificate are correct.
6. Click **OK > Save**.
7. Select the newly created SCEP setting and apply it to the *Windows 10 Desktop* label.

Enable use of device certificates for Apps@Work authentication

The last step is to enable use of certificates for authentication. Under the hood we are changing Apache configuration by adding the local CA created in the first paragraph to the list of accepted authorities.

To enable use of device certificates for Apps@Work authentication:

1. In the Admin Portal, go to **Settings > System Settings > Windows > Certificate Authentication**.
2. Check **Enable client certificate authentication**.
3. Select *Contoso Windows Certificate Authentication* certificate enrollment configuration.
4. Click **Save**.

Distributing apps for Windows 10 Desktop devices

Before you distribute in-house or third-party apps for Windows 10 Desktop devices, ensure that:

- apps are signed with a publicly trusted certificate issued by a CA.
- the devices are sideload enabled.

Certificates

We strongly recommend that in-house or third-party apps for Windows devices (8.1) are signed with a publicly trusted certificate issued by a Certificate Authority (CA). The CA's root certificate must be supported by the Windows OS. Signing with a publicly trusted certificate eliminates any additional steps by the device user.

We do not recommend signing apps with a self-signed certificate, as this will require the device user to perform additional steps before you can distribute the apps.

Sideloading keys

NOTE: This feature is supported for Windows Phone 8.1 only.

Typically, apps for Windows devices are signed and available only through the Windows Store. However, in-house and third-party apps can be made available through a process called sideloading. Each Window device must be sideload-enabled. You sideload-enable a device with sideload activation keys that you get directly from Microsoft.

For information about sideloading product activation keys, see
http://www.microsoft.com/licensing/activation/existing-customers/product-activation.aspx

For information about sideload enabling devices see
http://technet.microsoft.com/en-us/library/hh852635.aspx

NOTE: The previous URLs are not controlled by MobileIron and cannot be guaranteed to work or point to the correct page. They are provided here as a guide.

Pushing sideload activation keys

You can now push sideload activation keys to Windows devices (8.1) from Core Version 7.1. Sideload activation keys are required to sideload enable a Windows devices (8.1). This in turn allows you to sideload apps to the device.

Before you Begin

You must get the sideload activation key directly from Microsoft.

Configuration tasks

1. Adding the sideloading activation keys to Core.
2. Applying the sideloading activation keys configuration to a label.

Adding the sideloading activation key to Core

To add the sideloading activation keys to Core:

1. In the Admin Portal, go to **Policies & Configs > Configurations.**
2. Click **Add New > Windows > Sideloading Key**.
3. Use the following guidelines to fill the form:

Field	Description
Name	Enter a name for the configuration.
Description	Enter a description.
Sideloading key	Enter or copy and paste the sideloading key you received from Microsoft.

4. Click **Save**.

Applying the sideloading key configuration to a label

To apply the sideloading key configuration to a label:

1. In the Admin Portal, go to **Policies & Configs > Configurations**.
2. Select the sideloading key configuration.
3. Click **Actions > Apply to Label**.
4. In the **Apply to Label** dialog box, select the label.
5. Click **Apply**. The sideloading key is pushed to the devices in the label when the device checks in with Core.

Pushing the AET to Windows 8.1 Phone devices

If you are uploading third-party apps for distribution through MobileIron Core, you must also upload the AET (.aetx file) associated with the Symantec Enterprise Certificate used to sign the app. See Pushing the AET to Windows 8.1 Phone devices.

Follow these steps to push the token to Windows 8.1 Phone devices:

1. In the Admin Portal, go to **Policies & Configs > Configurations**.
2. Click **Add New > Windows > Enrollment Token (AET) (Windows Phone Only)** to open the New Application Enrollment Token dialog box.
3. Enter a **Name** and **Description** for the AET.
4. Click **Browse** to locate and select the AET file.
   This is a .aetx file.
5. Click **Save**.
6. In the **Configurations** page, select the AET.
7. Click **Actions > Apply to Label** and select the appropriate label.
   The AET is pushed to the devices to which the label is applied.

Distributing apps for Windows 8.1 Phone devices

This section describes the certificates and tokens required for distributing in-house apps for Windows Phone (8.1) devices only.

NOTE: After registration, the Windows Phone device is in Verified state. The device state changes to Active after the device checks in with MobileIron Core for the first time. This may take approximately ten seconds and up to one minute after registration. If the device user logs into the Apps@Work app before the device changes to Active state, the user will not be able to sign into Apps@Work because MobileIron Core is not yet associated with the device.

Certificates and tokens for in-house apps for Windows Phone devices

Before you distributing in-house apps for Windows Phone devices, you must do the following:

1. Review the certificates and tokens required for in-house apps for devices at:
   http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj206943.aspx
2. Create a Windows Phone Dev Center account at
   http://msdn.microsoft.com/en-us/library/windowsphone/help/jj206719.aspx
   The next step requires the Publisher ID for your company that is provided when you created the Dev Center account.
3. Get an enterprise mobile code signing certificate from Symantec at
   https://products.websecurity.symantec.com/orders/enrollment/microsoftCert.do
   Export the certificate in PFX format and be sure to export the private key with the certificate.
   You will sign your in-house app with the Symantec Enterprise Certificate. This is required for WP8 devices.
4. Generate the application enrollment token (AET) using the AETGenerator tool provided by the Windows Phone SDK 8.0.

For more information see
http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj735576.aspx

You upload the Application Enrollment Token (AET), which is an *.aetx file, to MobileIron Core. See Pushing the AET to Windows 8.1 Phone devices.

App file specifications for Windows Phone devices

The following file specifications apply to in-house apps for Windows Phone devices (WP8.1).

TABLE 16. APP FILE SPECIFICATIONS FOR WINDOWS DEVICES

Item	Format	Size	Number
App	XAP, APPX, APPXBUNDLE	100 MB maximum	—
Icon	PNG	99x99 pixels maximum	One per app.
Screen shots	PNG	480x800 pixels OR 480x854 pixels	Up to four per app.

App inventory on Windows 10 desktop devices

The inventory interval settings define how often MobileIron Core checks the app inventory on secured Windows 10 devices.

NOTE: By default, you will not see the full apps inventory list in the devices details page (**Devices & Users > Devices > Apps** tab). To see the full list, edit the Default Privacy Policy settings in **Policy & Configs > Policies > Add New > Privacy > Apps** by changing **App Catalog apps** to **All apps**.

Impact of App inventory options

App inventory, for each device, can be 20-40 MB, which can affect performance depending on how this feature is configured. When intervals do not match up with device check-in settings, the timing for checking the app inventory is deferred to the next scheduled device check-in time period to avoid excess syncs.

For example, assuming the **Device Checkin** setting is set to the default of 4 hours and the **Inventory Interval** settings are all set to 6 hours. The second app inventory sync will take place when the device check in next (in 8 hours).

NOTE: If an administrator initiates a Force Device Check-in, an inventory request is synced, reporting the installed apps in Core, independent of the Inventory Interval schedule. Every FDC resets the timer for the next inventory request.

App inventory intervals

This feature is disabled, by default. When enabled, the default time interval is 12 hours (all intervals are in hours). Use this feature to configure inventory interval settings for the following applications:

- App Store
- Non Store
- System
- Win32

How to configure an inventory intervals for apps

To configure inventory intervals for apps on Windows 10 devices:

1. In the Admin Portal, go to **Settings > Preferences**.
2. Scroll down to the **Windows 10 Inventory Configuration Settings** section.
3. Enter a number (in hours) for one or more type of app for which you want to take inventory.
   The options are:
   App Store Inventory Interval
   Non App Store Inventory Interval
   System Inventory Interval
   Win32 Inventory Interval

   NOTE: A Non Storeapp is any app that is not a system app or downloaded from the App Store (such as an in-house app).
4. Click **Save**.

How to turn on or off inventory intervals for apps

To turn on inventory intervals for apps on Windows 10 devices:

1. In the Admin Portal, go to **Policies & Configs > Policy**.
2. Select **Default Privacy Policy** and in the Policy Details pane, click **Edit** .
3. The Modify Privacy Policy dialog box opens.
4. Scroll to the **Windows 10 Inventory** group.
5. Click **Enabled** to check app inventory on the devices or **Disabled** to turn off the inventory sync.
6. Click **Save**.

NOTE: The default policy disables all app types.

How to view the app inventory

To view app inventory on Windows 10 devices:

1. In the Admin Portal, go to **Apps > Installed Apps**.
2. Select **Windows** in the left pane and then click **Search** to see a list of installed apps based on the configurations you set for secure Windows 10 devices.
3. If you have configured an inventory interval and turned on the feature, you will see a list of apps in the window.
   App inventory is pulled from the device the first time the device is enrolled.

Application scheduling

Windows 10 Desktop applications can be large, adding extra and extended load on networks and servers during key use times for the enterprise. This feature allows you to schedule a time to install applications, especially large applications, on devices during a time you choose. You can schedule the following types of applications:

- UWP
- MSI Wrapped Win32
- Win32
- Store
- Appx/appx bundle/AET token
- .EXE

Note The Following:

- BSP apps sync to Core only after 5-6 hours. Then it follows a manual sync or follows the configured BSP Sync Interval.
- If a new application is purchased on BSP portal, it is synced on to Core only after 24hrs. You can update the installation schedule for these BSP applications.
- Devices should be AAD enrolled for BSP applications and Store applications.
- The following procedure describes how to schedule application deployment when adding an application to the App Catalog. If you want to schedule deployment for an application already in the App Catalog, open the application, click **Edit**, then go directly to Step

Procedure

1. In the Admin Portal, select **Apps > App Catalog > Add+ > In-House**.
2. Browse for and upload one of the following types of applications:
   - UWP
   - MSI Wrapped Win32
   - Win32
   - Store

- Appx/appx bundle/AET token
- .EXE

3. Complete the wizard to add the application to the App Catalog.
4. Uncheck the **Feature this App in the Apps@Work catalog** option.
5. Check **Silent Upgrade/Install** and **Schedule Installation**.
6. Select a **Start Time** and an **End Time**.
   All times are local to where the device is located.
7. Click **Save**.

Restricting applications on Windows devices

Core allows administrators to restrict specified applications on Windows devices using one of the following two approaches:

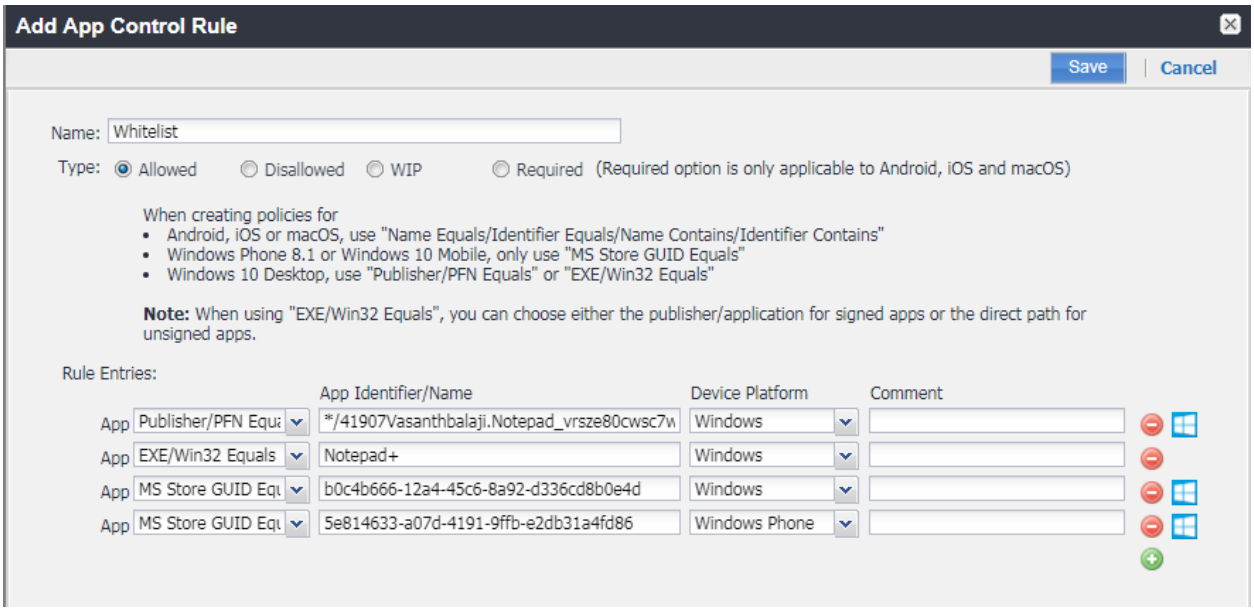
- Exclude (blacklist) - specifying applications to block, allowing all other applications on devices.
- Include (whitelist) - specifying applications and system to allow on devices, blocking all other applications not on the list.

The following topics describe how to restrict applications on Windows devices:

- Restricting applications on Windows 10 Desktop devices
- Restricting applications on Windows 10 Mobile devices
- Restricting applications on Windows Phone 8.1 devices

The figure below is an example of setting up a Whitelist App Control rule for Windows 10 Desktop, Windows 10 Mobile, and Windows Phone 8.1 devices.

FIGURE 12. SETTING UP A WHITELIST FOR WINDOWS DEVICES



Restricting applications on Windows 10 Desktop devices

The following procedures create a rule (called *Whitelist*) that allow device users to use only the specified applications, and no other applications. To include or exclude apps to security policies for windows 10 Desktop devices, you can select:

- **Publisher/PFN Equals** to use the dynamic lookup feature
  PFN is the Product Family Name of the application.
- **EXE/Win32 Equals** to use the application name

This section provides information on:

- Using the dynamic lookup tool to restrict applications on devices
- Using the application name to restrict applications on devices
- Blocking applications from Windows 10 Desktop devices

Using the dynamic lookup tool to restrict applications on devices

Procedure

1. In the Admin Portal, select **Apps > App Control > Add**.
2. Enter *Whitelist* in the **Name** field as the name of the rule.
3. Select **Allowed** for the **Type** option.
   Select **Disallowed** to block an application (blacklist).

4. Select **Publisher/PFN Equals** from the **App** drop-down.
   PFN is the Product Family Name of the application.
5. Leave the **App Identifier/Name** field blank.
6. Select **Windows** from the **Device Platform** drop-down.
7. Click the Windows icon to open the **Windows Store Search** window.
   The Windows icon is next to the red minus (-) icon to the right of the **Rule Entries** list.
8. Click the **Windows 10** option at the top of the search window.
   - The **Windows 10** option searches applications from the Windows 10 store, which supports both Windows 10 Phone and Windows 10 Desktops devices.
   - The **Windows Phone** option searches applications from the Windows Phone 8.1 store.
9. Enter an application name and click **Search**.
10. Locate the application and click the **Select** button to automatically insert the PFN into the **App Identifier/Name** field in the **Add App Control Rule** window.
11. (Optional) Click the green plus (+) icon to add more apps to the rule, as necessary.
12. Click **Save**.

Using the application name to restrict applications on devices

Procedure

1. In the Admin Portal, select **Apps > App Control > Add**.
2. Enter *Whitelist* in the **Name** field as the name of the rule.
3. Select **Allowed** for the **Type** option.
   Select **Disallowed** to block an application (blacklist).
4. Select **EXE/Win32 Equals** from the **App** drop-down.
5. Enter the name of the application (Notepad+, for instance) in the **App Identifier/Name** field.
6. Select **Windows** from the **Device Platform** drop-down.
7. (Optional) Click the green plus (+) icon to add more applications to the rule, as necessary.
8. Click **Save**.

Blocking applications from Windows 10 Desktop devices

When you block an application after it is already in use and installed from the Microsoft Store, the application will continue to run until users close it. When users open a blocked application, Windows displays a message on the device informing users that the application has been blocked by their system administrator. MobileIron sends instructions to the OS to block the specified application(s).

When users try to install a blocked application from the Microsoft Store, they see a message that the application has been blocked due to company policy.

Procedure

To apply an App Control rule to a security policy:

1. Go to **Policies & Configs > Policies**.
2. Select **Default Security Policy** and in the Policy Details pane, click **Edit**.
3. In the Modify Security Policy dialog box, scroll to the **For Windows Devices** section in the **Access Control** group.
4. Select the check box next to **Application Restrictions** and select *Blacklist* from the drop-down.
5. Click **Save**.

Restricting applications on Windows 10 Mobile devices

Procedure

1. Go to **Apps > App Control**.
2. Click **Add**. The Add App Control Rule dialog box opens.
3. In the **Name** field, Enter *Whitelist* as the name of the rule.
4. In the **Type** field, select **Allowed**. Select **Disallowed** to create a Blacklist and block an application.
5. In the **App** drop-down, select **MS Store GUID Equals** .
6. Leave the **App Identifier/Name** field blank.
7. In the **Device Platform** drop-down, select **Windows**.
8. Click the Windows icon to open the **Windows Store Search** dialog box. (The Windows icon is next to the red minus (-) icon to the right of the **Rule Entries** list.)
9. Click the **Windows 10** option.
   - The Windows Phone option searches applications from the Windows Phone 8.1 store.
   - The Windows 10 option searches applications from the Windows 10 store, which supports both Windows 10 Phone and Windows 10 Desktops devices.
10. Enter an application name (Notepad+, for example) and click **Search**.
11. Locate the application and click the **Select** button to automatically insert the GUID into the **App Identifier/Name** field in the **Add App Control Rule** dialog box.
12. (Optional) Click the green plus (+) icon to add more apps to the rule, as necessary.
13. Click **Save**.

Restricting applications on Windows Phone 8.1 devices

Procedure

1. Go to **Apps > App Control**.
2. Click **Add**. The Add App Control Rule dialog box opens.
3. In the **Name** field, Enter *Whitelist* as the name of the rule.

4. In the **Type** field, select **Allowed**. Select **Disallowed** to create a Blacklist and block an application.
5. In the **App** drop-down, select **MS Store GUID Equals** .
6. Leave the **App Identifier/Name** field blank.
7. Select **Windows Phone** from the **Device Platform** drop-down.
8. Click the Windows icon to open the **Windows Store Search** dialog box. (The Windows icon is next to the red minus (-) icon to the right of the **Rule Entries** list.)
9. Click the **Windows Phone** option.
10. Enter an application name and click **Search**.
11. Locate the application and click the **Select** button to automatically insert the PFN into the **App Identifier/Name** field in the **Add App Control Rule** dialog box.
12. (Optional) Click the green plus (+) icon to add more apps to the rule, as necessary.
13. Click **Save**.

Upgrading from Windows Phone 8.1 devices to Windows 10 Mobile devices

IMPORTANT: When using the newer API, not all applications will appear in the store. The applications called Settings Apps and Inbox or those applications that default applications on the device, will not display in the store. To look up those applications, visit https://docs.microsoft.com/en-us/windows/client-management/mdm/applocker-csp#inboxappsandcomponents .

In the link the tool Microsoft provides for golden device reviewing, not all of the GUID's in the Microsoft store point to the actual application on the device. MobileIron and Microsoft recommend you create a golden device and use that tool to review the actual GUID's needed.

For customers who are upgrading from Windows 8.1 to Windows 10, it is important to add both the Windows 10 and Windows 8.1 rules before upgrading. Failing to do so could cause the device to become unresponsive.

Take the following precautions, if you upgrade from Windows Phone 8.1 devices to Windows 10 Mobile devices and you use an application restriction rule on your Windows Phone 8.1 devices:

1. Prior to upgrading, remove your 8.1 based restriction rule.
2. After upgrading, apply an application restriction rule to the device using the new Windows 10 Mobile Rules.
3. After upgrading, manually create rules for all applications that used PFN to use GUIDs

NOTE: If you want to whitelist the Apps@Work application, you can find its GUID under the App Catalog detail page.

Working with apps

MobileIron allows you to distribute and track in-house and third-party apps to your managed devices. You can add the apps for Windows Phone devices (WP8.1) from the following sources:

- MobileIron Core (in-house apps)
- Windows Store (third-party apps)

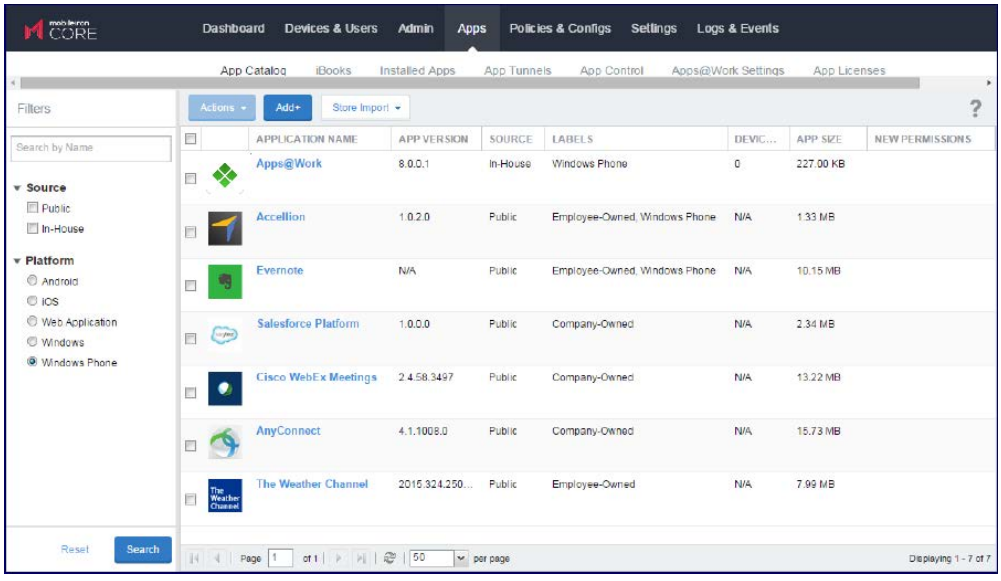
You can distribute and track in-house and third-party apps to your managed devices. These apps are listed on your managed Windows Phone devices running Apps@Work in the following screens:

- company apps (in-house apps)
- recommended (third-party apps)
- featured apps (in-house and third-party apps)

The App Catalog

The **App Catalog** is a centralized location for the apps you want to manage for your users. By importing apps into the App Catalog, you can make the apps available for users to download to their devices.

You can provide device users with links to recommended Windows apps on the Microsoft Store, or links to internally developed apps they can download from MobileIron Core using Apps@Work on their device.



Use the App Catalog to:

- add, configure, and remove managed apps
- install and uninstall managed apps to devices using labels

- group apps into categories to be displayed in Apps@Work on the device

The App Catalog also allows you to view app details at a glance, such as the app name, size, and version number, the labels to which the app is applied, the origins of the app (public or in-house), and the number of devices to which the app is installed.

Company apps

In-house apps are installed from MobileIron Core and have been developed and distributed by your company. They are called In-house apps on the Admin Portal and Company apps on your managed Windows Phone devices running Apps@Work. Upload these apps to the App Catalog from **Admin Portal > Apps > App Catalog > In-house**.

In-house apps are removed from the device when the device is un-enrolled from device management. Recommended apps are not removed.

Recommended apps

Third-party apps are installed from the Microsoft Store for Windows and Windows Phone devices and are served from public sources. They are imported to the App Catalog from the Windows Store. Import these apps to the App Catalog from **Admin Portal > Apps > App Catalog > Windows**.

The MobileIron administrator adds selected third-party apps to the App Catalog, which are made available to devices based on the labels applied. An app is downloaded by device users when they select the Apps@Work app on their devices.

Recommended apps are not removed from the device when the device is un-enrolled from device management.

If you are uploading third-party apps for distribution through MobileIron Core, you must also upload the AET (.aetx file) associated with the Symantec Enterprise Certificate used to sign the app. See Pushing the AET to Windows 8.1 Phone devices.

Featured apps

When adding apps to the App Catalog, you can designate an in-house or third-party app as a featured app. These apps are listed on the Apps@Work featured apps screen on managed devices.

Adding in-house apps to the App Catalog

Use the following steps to add apps to the App Catalog with the app wizard.

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Click the **Add+** button.

3. Click **In-House** then click **Browse** to navigate to and select the app.
   This is a .xap, a .appx or .appx bundle for WP8.1 devices.
4. Click **Next**.
5. Enter information into the **Description** section, if necessary, using the following guidelines:
   - **Application Name**: The name of the app as defined by the developer displays in Apps@Work on the device and in the Apps@Work catalog.
   - **Display Version**: The version of the app. This field is not editable.
   - **Developer**: The author of the app as defined by the developer. This field is not editable.
   - **Category**: Select one or more categories if you would like this app to be displayed in a specific group of apps on the device. Select the category from the drop-down list. The app appears under the selected category on the device. To add a new category, click the **Add New Category** link.
   - **Description**: Enter a description for the app.
6. Enter information into the **Apps@Work Catalog** section and use the following guidelines:
   - **Feature this App in the Apps@Work catalog:** Select if you want to highlight this app in the Featured apps list.
   - **Allow app downloads over insecure networks**: Select this if you are providing an Override URL (next field) that uses the HTTP URL scheme instead of HTTPS. Override URLs are intended for use behind a firewall, using a trusted and secure internal network. Before you use an HTTP URL, make sure you understand the risks of using an insecure connection.
   - **Override URL**: If you are using an alternate source for downloading in-house apps, enter that URL here. The URL must point to the in-house app in its alternate location. Override URLs are intended for use behind a firewall, using a trusted and secure internal network. Manual synchronization is required with the alternate HTTP server on which app are stored.
     See Override for in-house app URLs for the requirements for this configuration before using it.
   - **Application Enrollment Token (AET)**: The app enrollment token for Apps@Work enables companies to publish and distribute apps directly to Windows Phone devices (WP8.1) and bypassing the Windows Store. Companies can install in-house apps after they enroll their phones for app distribution from their company, then users who are enrolled for app distribution can install the company apps.
   - **Silent Upgrade/Install**: Clearing this check box will require the device user to manually install the app. Checking this box to install the app silently. The app is installed when the device checks in with Core. User action is not required.
   - **Schedule Installation**: Click the check box to schedule the installation of the application, then select a Start Time and End Time. This is especially useful for installing large applications during times that the network is not busy.
     Do not select **Feature this App in the Apps@Work catalog** if you want to set up a schedule to install an application.
7. Enter information into the **Apps@Work Catalog** section to update the information, if necessary, using the following guidelines:

- **App Icon**: Click **Browse...** to navigate and select a new graphic. Click **OK** to add the graphic. You can upload one icon per app.
- **Screenshots**: Click **Browse...** to navigate and select a new screenshot. Click **OK** to add the screenshot. You can upload up to four screenshots per app.

8. Click **Finish**.

Related topics:

App management action workflows

Apps@Work in Mobile@Work for Android

Adding third-party apps to the App Catalog

MobileIron Core allows you to add apps to the App Catalog for Windows and Windows Phone devices using the following two methods:

- **Add+ button**: Opens the app wizard. Use this wizard to add one app at a time with each wizard. After completing the wizard, it adds the app to the App Catalog.
- **Quick Import button**: Opens the **Windows Store Search** window. Use this method to search for and import one or more apps while the window is open.

Adding third-party apps using the app wizard

Use the following steps to add apps to the App Catalog with the app wizard:

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Click **Add+ > Windows**.
3. Enter a name or application type in the **Application Name** search box.
4. Select the **App Store** locale and language.
5. Enter the number of entries you want to retrieve in the **Limit** field.
   To improve search performance, the default is set to 20. You can enter a number between 20 and 50.
6. Click the name of the app in the **Name** column.
   For detailed information about the app, click the icon to open a link to the third-party web page.
7. Click **Next** to view and modify the app description as it will appear on the device and in the App Catalog.
8. Click **Finish** to complete the app wizard and add the app to the App Catalog.

Adding third-party apps using Quick Import

Use the following steps to import apps using the **Quick Import** button:

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Click **Quick Import > Windows > Windows Phone or Windows.**
3. Enter a name or application type in the **Application Name** search box.
4. Select the **App Store** locale and language.
5. Enter the number of entries you want to retrieve in the **Limit** field.
   To improve search performance, the default is set to 20. You can enter a number between 20 and 50.
6. Click **Search** to see the apps that match your criteria.
7. Click the **Import** link on the line of the app you want to import to import the app information into the **App Catalog** page.
8. Click **OK** at the **Successfully Added** window.
9. Close the **Windows Store Search** window when you have finished adding apps.

Deploying apps

Follow these steps to silently push the apps to a device using one or more labels.

1. In the Admin Portal, go to **Apps > App Catalog**.
2. From the **Filters** pane on the left, select **Platform > Windows**.or **Windows Phone**.
3. Select one or more apps.
4. Click **Actions > Apply To Label** and select one or more labels to apply.
5. Click **Apply**.
6. Apps are made available to the devices with the label. Depending on the how the label was configured, the app is silently installed (no action required by the device user) and in other cases it is available, but requires that the user install it.

NOTE: Only the latest version of the app displays in the Apps@Work app on the device. When you remove the label, the app is no longer available to devices associated with that label. The app is not deleted from MobileIron Core or from the devices on which it is already installed.

Related topics: App management action workflows

Editing in-house app information

Use the following steps to edit in-house app information, icons, and screenshots:

1. In the Admin Portal, go to **Apps > App Catalog**.
2. From the **Filters** pane on the left, select **Platform > Windows** or **Windows Phone**.
3. Click the app name link in the **Application Name** column to display the app information.
4. Click the **Edit** button to edit the following information:

Item	Description
Description	
App Name	The edited name appears in the App Catalog and Apps@Work, however, when you install the app on the device, the original name will be displayed on the device.
Display Version	The version of the app. This field is not editable.
Developer	The author of the app as defined by the developer. This field is not editable.
Category	Select one or more categories to display this app in a category tab in Apps@Work or add a new category. a. Click **Add New Category** to define new categories. b. Enter a category **Name** (up to 64 characters). c. Enter a **Description** (up to 255 characters). d. In the **Category Icon** section, click the **Replace Icon** button. e. Browse and select an icon that will represent this Category. f. Click **Save**.
Description	Edit the app description.
Apps@Work Catalog	
Feature this App in the Apps@Work catalog	By default, the check box is selected to list the app in the Featured apps list in Apps@Work. This feature does not apply to AppConnect apps.
Allow app downloads over insecure networks	Select this if you are providing an Override URL (next field) that uses the HTTP URL scheme instead of HTTPS. Override URLs are intended for use behind a firewall, using a trusted and secure internal network. Before you use an HTTP URL, make sure you understand the risks of using an insecure connection.
Override URL	If you are using an alternate source for downloading in-house apps, enter that URL here. The URL must point to the in-house app in its alternate location. Override URLs are intended for use behind a firewall, using a trusted and secure internal network. Manual synchronization is required with the alternate HTTP server on which app are stored. See Override for in-house app URLs for the requirements for this configuration before using it.

Item	Description
Application Enrollment Token (AET)	The app enrollment token for Apps@Work enables companies to publish and distribute apps directly to Windows Phone devices (WP8.1) and bypassing the Windows Store. Companies can install in-house apps after they enroll their phones for app distribution from their company, then users who are enrolled for app distribution can install the company apps.
Silent Upgrade / Install	• Clearing the check box means the device user will need to manually install the app. When the check box is cleared, the Schedule Installation field disappears. • Selecting the check box will install the app silently. The app is installed when the device checks in with Core. User action is not required. For more information, see Silent install and uninstall of mandatory apps. NOTE: Silent install is not supported for MAM-only Android devices.
Schedule Installation	Selecting the check box means the application will be installed within the specified time interval. All timings are device local time.
Screenshots	
App Icon	Click **Browse** to navigate and select a new graphic. Click **OK**to replace the existing graphic.
Screenshots	Click **Browse** to navigate and select a new screenshot. Click **OK** to replace the existing screenshot.

5. Click **Save**.

Application dependency deployment

Many Windows applications need extra programs or libraries in order to run effectively. These are commonly known as dependencies. When a Windows application comes from the Microsoft store these dependencies are packaged with the device. However, in-house applications need include dependencies at the time the application is downloaded.

As app developers do not supply dependency libraries with their apps, MobileIron Core adds the dependencies on the devices before administrators install the apps on the devices, if necessary when developers use In-House deployment. Core identifies the dependencies by name, uploads of the files, and deploys the dependencies on the device before administrators push the apps to the device.

Deploying app dependencies

When you upload an in-house app, MobileIron Core scans the app to identify dependencies. If Core finds any, it lists them in the third step of the **Add App Wizard**. For any dependency needed by an application administrators can select to upload a dependency file. However, some apps might not install without uploading the dependency file.

NOTE: Although an application needs a dependency file, Core does not require that you upload any of the files to deploy an app.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Click **Add+ > In-House** to open the Add App Wizard.
3. Click **Browse** to locate and select your in-house app.
4. Select the main binary file for the app and click **Next**.
5. Provide a description of the app and click **Next**.
   Core scans the app for dependency files and lists them in the **Application Dependencies** table.
6. Review the app information and verify that you selected the correct app.
7. Click **Upload File** next to a dependency file name.
8. Click **Browse** to locate and select a local copy of the file.
9. Click **Upload the Dependency File**.
   The administrator can choose not to associate a dependency for the app installation by clicking **No** for the **IS ASSOCIATED** column.
10. Click **Add Additional Dependency** to upload the additional dependencies, if the application requires additional dependencies for installation, but are not part of the application's manifest file (optional).
11. Repeat, as necessary, for any uploading any other dependency files.
12. Click **Finish** to complete the app upload process.
13. Apply the app to a label to deploy the app to devices.
    a. Select the app.
    b. Click **Actions > Apply to Labels**.
    c. Select one or more labels.
    d. Click **Apply**.
       The next time the devices, associated with the selected label(s), sync with Core, the app is deployed n the device along with the dependent files.

Editing third-party app information

Use the following steps to edit third-party app information:

1. In the Admin Portal, go to **Apps > App Catalog**.
2. From the **Filters** pane on the left, select **Platform > Windows**.or **Windows Phone**.
3. Click the name of the app in the **Name** column.
4. Click **Edit** and make the necessary changes in fields that are editable.
5. Information varies depending on the app you edit.
6. Click **Save**.

Updating apps in the App Catalog

Use the following steps to update apps using the **Quick Import** button:

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Click **Quick Import > Windows > Windows Phone or Windows**.
3. Enter the name of the app you want to update in the **Application Name** search box.
4. Click **Search**.
   Scroll to locate the app, if necessary.
5. Click the **Update** link next to the version number.
6. Click **OK** at the **Successfully Updated** window.
7. Close the **Windows Store Search** window when you have finished updating the app.

Deleting apps from MobileIron Core

Follow these steps to delete one or more apps:

1. In the Admin Portal, go to **Apps > App Catalog**.
2. From the **Filters** pane on the left, select **Platform > Windows** or **Windows Phone**.
3. Select one or more apps to delete.
4. Select **Action > Delete**.
5. Click **Yes** in the confirmation box.
   This action deletes the app from MobileIron Core, but does not delete it from the device.

Managing apps on MAM-only devices

MobileIron Core allows you to specify iOS and Android devices as MAM-only. Core provides Mobile App Management (MAM) to such devices, but does not provide them with Mobile Device Management (MDM).

MAM-only device overview

MobileIron Core provides both Mobile Device Management (MDM) and Mobile App Management (MAM). However, sometimes you have situations in which you want to manage apps without device management. Some examples are:

- You have contractors who need your relevant apps, but their devices are managed by another MDM system.
- You have employees who need your relevant apps on their personal devices, but your privacy or legal requirements do not allow device management.

MobileIron Core supports MAM-only devices for iOS and Android. With MAM-only devices, Apps@Work on the device presents registered device users with the apps in the App Catalog. However, features that require device management are not supported.

Related topics

- MAM-only iOS devices
- MAM-only Android devices

MAM-only iOS devices

MobileIron Core can support **only one** of the following types of registered iOS devices:

- devices that support both MAM and MDM
- devices that support only MAM (MAM-only devices)

Core cannot simultaneously support MAM-only devices and devices that support both MDM and MAM. You configure your choice by enabling or disabling iOS MDM support in the Admin Portal. You make this choice before any iOS devices register with Core. Note that your choice has no impact on Core capabilities for other device platforms, such as Android or Windows.

Whether or not you disable iOS MDM on Core, you use the App Catalog on MobileIron Core and Apps@Work on the device to make apps available to devices. Apps@Work is presented on the device either in a web clip or in Safari.

However, in the MAM-only case, Core does not send iOS devices the MDM configurations and certificates required for MDM activity on a device. These MDM configurations and certificates, as listed in the Admin Portal in **Policies & Configs > Configurations,** are:

- the System - iOS enrollment CA certificate
- the System - iOS enrollment SCEP certificate
- the System - iOS MDM configuration

Without these MDM configurations and certificates, Core does not support any MDM features, including MDM features relating to apps, such as:

- per-app VPN settings
- managed app settings
- managed app configuration settings
- requiring data protection
- displaying the apps that are installed on devices

Required Mobile@Work version for MAM-only iOS devices

MAM-only iOS device support requires Mobile@Work 9.7 for iOS through the most recently released version as supported by MobileIron.

Supported features on MAM-only iOS devices

When iOS MDM is disabled, **only the following features are supported on iOS devices**:

- In-app registration using Mobile@Work for iOS.
  No other registration methods are supported for MAM-only iOS devices.
- Pushing apps to the devices using the Apps@Work web clip.
- All types of apps are supported:
  - AppConnect apps (in-house or from the Apple App Store)
  - Non-AppConnect apps (in-house or from the Apple App Store)
  - Web applications

NOTE: The following app settings in the App Catalog are not supported for MAM-only iOS apps: per app VPN settings, managed app settings, managed app configuration settings, and requiring data protection.

- AppTunnel with HTTP/S tunneling
- AppConnect-related policies and configurations:
  - AppConnect global policy
  - AppConnect container policies
  - AppConnect app configurations
  - Web@Work settings
  - Docs@Work settings
- Standalone Sentry with ActiveSync support, using AppConnect-enabled Email+ for iOS
- The following subset of actions from the Admin Portal (**Devices & Users > Devices > Actions**):
  - Force Device Check-in
  - Send Message
  - Apply to Label
  - Remove from Label
  - Retire
  - Block AppTunnel
  - Allow AppTunnel
- Compliance actions for only the following security violations on the security policy:
  - When a device has been out of contact with Core too long
  - When the iOS version is less than a specified version
  - When the device is compromised (jailbroken)
  - When particular device models are not allowed

No other iOS features are supported. For example:

- MobileIron Core does not support applying any configurations or policies (in the Admin Portal **Policies & Configs**) that are not related to AppConnect. For example, do not apply iOS restrictions or Wi-Fi settings.
- The self-service user portal and **My Devices** in Mobile@Work are not available.
- Admin Portal MDM-related actions cannot be applied to iOS devices. These actions include wipe, lock, unlock, and locate. The Admin Portal displays an error message when you attempt to take these actions.
- iOS native email is not supported, because it requires the Exchange setting which requires MDM.
- Multi-user sign-in is not supported.
- MobileIron Tunnel (AppTunnel with TCP tunneling) is not supported.
- MobileIron Core does not display the apps installed on MAM-only iOS devices.

- Changes you make on MobileIron Core do not result in uninstalling an app from an MAM-only iOS device. For example, the app is not uninstalled if you remove an app from the App Catalog, or remove its label, or retire the device.
- The Apps@Work container app is not supported.

Device check-in on MAM-only iOS devices

The sync interval on the sync policy has no impact on MAM-only iOS devices. Therefore, automated device check-ins occur only when the AppConnect app check-in interval expires. You configure this value on the AppConnect global policy. When the AppConnect app check-in interval expires, Mobile@Work checks in with MobileIron Core, and receives updates to policies and configurations.

Device check-ins also occur when:

- When an AppConnect app launches for the first time.
- A device user taps **Check for Updates** in Mobile@Work for iOS.
- A device user brings Mobile@Work to the foreground.
- You do a Force Device Check-in from the Admin Portal (**Devices & Users > Devices > Actions**).

NOTE: This action does not update the AppConnect-related policies on the device.

Trusted certificates and MAM-only iOS devices

When you set up MobileIron Core, you provide a client TLS certificate. This certificate secures communication between the mobile device and MobileIron Core. Often the client TLS certificate is the same certificate as the Portal certificate, which secures communication between a web browser and Core.

If the client TLS certificate or Portal certificate are not ones that are trusted by iOS, on MAM-only iOS devices, unlike on MDM iOS devices, the device user must manually accept the certificates. To do this, after completing the Mobile@Work registration process, the device user must go to the device's Settings, and navigate to **Settings > General > About > Certificate Trust Settings**, and trust the certificates. Therefore, if you want to streamline the device user experience, use only certificates trusted by iOS for the client TLS certificate and the Portal certificate.

For lists of available trusted root certificates in iOS, see Apple documentation at https://support.apple.com.

Related topics

- "Types of certificates" in the *MobileIron Core Device Management Guide for iOS and macOS Devices*
- "Certificate mgmt" in the *MobileIron Core System Manager Guide*

Configurations and certificates for MAM-only iOS devices

When you use MAM-only iOS devices, MobileIron Core supports delivering only certain types of configurations and certificates to the device. These configurations belong to two categories:

- AppConnect-related configurations and policies on MAM-only iOS devices
- Other certificates and configurations that are supported with MAM-only iOS devices

NOTE: You can use the Core option to not install profiles on iOS devices to not deliver this category of certificates and configurations to devices.

AppConnect-related configurations and policies on MAM-only iOS devices

The AppConnect-related configurations and policies on MAM-only iOS devices are:

- the AppConnect global policy
- the AppConnect container policy
- the AppConnect app configuration
- the Docs@Work setting
- the Web@Work setting

Other certificates and configurations that are supported with MAM-only iOS devices

The other certificates and configurations supported with MAM-only devices, as listed in the Admin Portal in **Policies & Configs > Configurations**, are:

- the **System - iOS Enterprise AppStore** web clip
- the **System - iOS Enterprise AppStore SCEP** certificate
- the **System - TLS Trust Certificate Chain for Mobile Devices** certificate

Note the following regarding configurations and certificates when using MAM-only iOS devices:

- MobileIron Core does not receive status from the device about whether these non-AppConnect related certificates and configurations have been applied. Therefore, the status of these configurations in the device details display remains as **Sent**.
- When you retire a device, the certificates and configurations are not removed. A device user can manually remove them.

Core option to not install profiles on iOS devices

With a setting on the MobileIron Core, you can instruct Core to **not** install profiles on iOS devices. When you enable this setting, Core does not send the non-AppConnect related certificates and configurations to MAM-only iOS devices.

Not installing these profiles is useful if your device users use their own (BYOD - Bring Your Own Device) devices. Sometimes BYOD users are concerned by additional certificates and configurations on their devices that they might misconstrue as device management. However, installing these profiles allows device users to use the Apps@Work web clip, which means they can easily view and install apps without entering any further credentials. Your requirements for user convenience versus user concerns determine your choice for this setting.

The setting is the **Enable Configuration Profiles** field on the privacy policy that Core applies to the device. The field is selected by default. Because clearing this field means that Core does not push the Apps@Work web clip and certificate to the device, the device user needs another way to access Apps@Work. Therefore, when you clear this field, Mobile@Work for iOS displays an Apps@Work button on its home screen. When the device user taps that button, Apps@Work opens in Safari. The device user logs into Apps@Work with a user name and password.

Also, when you clear **Enable Configuration Profiles**:

- The Portal HTTPS certificate you configure on the MobileIron Core System Manager must be trusted by iOS if you want the device user to download in-house apps from Apps@Work. For lists of available trusted root certificates in iOS, see Apple documentation at https://support.apple.com.
- The setting has no impact on versions of Mobile@Work prior to 10.0. That is, the non-AppConnect related certificates and configurations will be installed on the device.

In-house apps and provisioning profiles for MAM-only iOS devices

In-house iOS apps require a provisioning profile. However, if you replace the provisioning profile, when MobileIron Core delivers the updated provisioning profile to the impacted iOS devices, it also resends all the non-AppConnect-related policies and configurations to the devices. Mobile@Work will prompt the device user to re-install each certificate and configuration.

The device user experience on MAM-only iOS devices

The device user experience on MAM-only iOS devices is the same as on devices that also support MDM, with these exceptions:

- Device users must register with MobileIron Core using the Mobile@Work for iOS app. (No other registration methods are available for MAM-only iOS devices). The registration process in Mobile@Work is somewhat shorter than on devices which support MDM, because the MDM configurations and certificates are not installed.
- The privacy policy that Mobile@Work presents to the device user is shorter on MAM-only devices. It tells the user only that it will not access personal content. Other statements in the policy in MDM devices, such as statements about providing some device details to the user's company, are not applicable on a MAM-only device.
- When a device user uses Apps@Work to install an app from the Apple App Store, the behavior is different on MAM-only devices than on devices with MDM.
    - On MAM-only devices: Tapping **Install** for an app in Apps@Work opens Safari to the app's entry in the Apple App Store. From there, the device user downloads and opens the app. The app is installed just as if the device user had gone directly to the Apple App Store.
    - On MDM devices: tapping **Install** for an app presents a message that MobileIron Core will install the app from the Apple App Store and manage the app. The device user enters an Apple ID, and the app is installed. If the device user had gone directly to the Apple App Store to install the app, the app would not be managed.

MAM-only Android devices

You can specify that MobileIron Core provides MAM-only features to some registered Android devices, but both MAM and MDM features to other Android devices.

NOTE: Your choice has no impact on the Core capabilities for other device platforms, such as Android enterprise devices, iOS devices, and Windows devices.

To make an Android device MAM-only, you configure an Android quick setup policy in which you disable device administration. When you apply this policy to Android devices, MobileIron Core supports app installation using Apps@Work and most policies and configurations. However, Core cannot perform any features that require the device administrator on the device. Specifically, Core cannot do the following on the MAM-only Android devices:

- Cannot enforce device password requirements from the security policy.
- Cannot enforce device encryption requirements from the security policy.

Exception: Core can enforce device log encryption from the security policy.

- Cannot enforce Android-related lockdown policies from the lockdown policy.
- Cannot apply Samsung-specific features, which include:
  - Samsung Knox features, including per app VPN
  - Samsung native email
  - Samsung-related policies: Samsung kiosk policy, Samsung general policy, Android firmware policy
  - Samsung-related configurations: Samsung APN, Samsung browser, Samsung kiosk, and Samsung Knox container
  - Samsung-related VPN configurations: OpenVPN, Samsung Knox IPsec, and MobileIron Tunnel (Samsung Knox Workspace)
  - Silent installation of apps
- Cannot apply silent installation of apps on Zebra devices
- Does not support silent installation of certificates
- The device user is always prompted to accept a certificate.
- Cannot enforce blocking smart lock or blocking fingerprint from the security policy.
- Cannot enforce common criteria mode from the security policy.
- Cannot enforce compliance actions for the following security violations on the security policy:
  - When data encryption is disabled
  - When the device administrator is deactivated
  - When Samsung Knox device attestation fails
- Cannot wipe the MAM-only device.

Note The Following:

- Mobile@Work on the device also cannot wipe the device, even if the AppConnect global policy or the security policy specify wipe as a device-initiated compliance (local compliance) action.
- When using Android Custom ROM menus, if you choose wipe as a compliance action, the device is not wiped if the security violation occurs. Instead, the device is retired.

Related topics

- MAM-only device overview
- Configuring MAM-only Android devices

Configuring MAM-only iOS devices

Configuring MAM-only iOS devices requires the following steps:

1. Disabling the MDM profile
2. Configuring the security policy for MAM-only iOS devices
3. Configuring the privacy policy for MAM-only iOS devices
4. Configuring the sync policy for MAM-only iOS devices
5. Configuring the lockdown policy for MAM-only iOS devices
6. Configuring the Apps@Work web clip for MAM-only iOS devices
7. Populating the iOS App Catalog for MAM-only iOS devices
8. Publishing iOS apps to Apps@Work on MAM-only iOS devices
9. Configuring AppConnect and AppTunnel for MAM-only iOS devices

IMPORTANT: Before configuring MobileIron Core for MAM-only iOS devices, make sure no iOS devices are registered.

Disabling the MDM profile

Disabling the MDM profile for all iOS devices is necessary for configuring MobileIron Core to support only MAM-only iOS devices.

Procedure

1. In the Admin Portal, go to **Settings > System Settings > iOS > MDM**.
2. Make sure **Enable MDM profile** is not selected.
3. Click **Save**.

Configuring the security policy for MAM-only iOS devices

Only a few fields on the security policy apply to MAM-only iOS devices. This procedure explains how to configure the default security policy. However, the same considerations apply to any security policy that you label for iOS

devices or a subset of iOS devices.

NOTE: If you are applying the default security policy or a custom security policy to both MAM-only iOS devices and to non-iOS devices, set the appropriate fields for non-iOS devices according to your requirements

Procedure

1. In the Admin Portal, go to **Policies & Configs > Policies**.
2. Select the default security policy.
3. Click **Edit**. The Modify Security Policy dialog box opens.
4. The **Password** section does not apply to MAM-only iOS devices.
5. The **Data Encryption** section does not apply to MAM-only iOS devices.
6. The **Android, Android enterprise, Windows 8.1**, and **Windows 10** sections do not apply to MAM-only iOS devices.
7. In the **Access Control** section, in **For All Platforms,** select the compliance action, if any, that you require for the security violation **when a device has not connected to Core in X days**. This security violation is the only one in this section supported for MAM-only iOS devices.
8. In the **Access Control** section, in **For iOS devices,** select the compliance action, if any, that you require for these security violations, which are the only ones in this section supported for MAM-only iOS devices:
   - **when iOS version is less than**
   - **when a compromised iOS device is detected**
   - **for the following disallowed devices**
9. Click **Save > OK**.

Related topics

- MAM-only iOS devices
- "Security policies" in *Getting Started with MobileIron Core*

Configuring the privacy policy for MAM-only iOS devices

Only a few fields on the privacy policy apply to MAM-only iOS devices. This procedure explains how to configure the default privacy policy. However, the same considerations apply to any privacy policy that you label for iOS devices or a subset of iOS devices.

NOTE: If you are applying the privacy policy or a custom privacy policy to both MAM-only iOS devices and to non-iOS devices, set the appropriate fields for non-iOS devices according to your requirements

Procedure

1. In the Admin Portal, go to **Policies & Configs > Policies**.
2. Select the default privacy policy.
3. Click **Edit**. The Modify Privacy Policy dialog box opens.
4. Set **Apps** to the appropriate value for non-iOS devices that this privacy policy applies to.
   This field has no impact to MAM-only iOS devices. It applies to iOS devices only if they are MDM enabled.
5. Set **SMS Log and Call Log** to the appropriate value for Android devices that this privacy policy applies to.
   These fields apply only to Android devices.
6. Set **iOS Location-Based Wakeups** to **Disabled**.
   Set this field to **Disabled** because you should not track the location of MAM-only devices.
7. Set **Location** to **None**.
   Set this field to **None** because you should not track the location of MAM-only devices.
8. Set **Collect Roaming Status** to the appropriate value for Android devices that this privacy policy applies to.
   This field applies only to Android devices.
9. Clear **Enable Configuration Profiles** if you do not want MobileIron Core to send non-AppConnect-related configurations and certificates to MAM-only iOS devices, including the Apps@Work web clip and certificate.
   Clearing this setting impacts only Mobile@Work 10.0 through the most recently released version as supported by MobileIron. Prior versions of Mobile@Work receive the configurations and certificates regardless of this setting.
10. Set **iOS Installed App Inventory** to **All Apps**.
    However, this field has no impact to MAM-only iOS devices. It applies to iOS devices only if they are MDM enabled.
11. The **Windows 10 Inventory** and **Android Warning Banner on the Device Reboot** sections do not apply to MAM-only iOS devices.
12. Click **Save > OK**.

Related topics

- Core option to not install profiles on iOS devices
- “Privacy policies” in *Getting Started with MobileIron Core*

Configuring the sync policy for MAM-only iOS devices

No sync policy fields apply to MAM-only iOS devices. If your MobileIron Core deployment includes only MAM-only iOS devices, you can skip this step. However, if your deployment includes other device platforms, configure the sync policy to meet your requirements for the other platforms.

Related topics

"Sync policies" in *Getting Started with MobileIron Core*

Configuring the lockdown policy for MAM-only iOS devices

The lockdown policy does not apply to iOS devices. If your MobileIron Core deployment includes only MAM-only iOS devices, you can ignore the lockdown policy. However, if your deployment includes other device platforms, configure the lockdown policy to meet your requirements.

Related topics

"Lockdown policies" in *Getting Started with MobileIron Core*

Configuring the Apps@Work web clip for MAM-only iOS devices

Configuring the Apps@Work web clip is necessary to support MAM-only iOS devices. For configuration information, see Setting up Apps@Work for iOS and macOS.

NOTE: The AppConnect container app is not supported on MAM-only iOS devices.

Populating the iOS App Catalog for MAM-only iOS devices

Populating the App Catalog on MobileIron Core with iOS apps is necessary to support MAM-only iOS devices. This task is the same as when iOS devices support MDM. However, the following features, available when adding or editing an app in the App Catalog, are not supported:

- Per App VPN settings
- Managed app settings
- Managed app configuration settings
- Requiring data protection

For configuration information, see Populating the iOS and macOS App Catalogs.

Publishing iOS apps to Apps@Work on MAM-only iOS devices

Making iOS apps available to device users in Apps@Work on MAM-only iOS devices is the same as it is with iOS devices that support MDM.

For configuration information, see Publishing iOS and macOS apps to Apps@Work.

Configuring AppConnect and AppTunnel for MAM-only iOS devices

Configuring AppConnect for MAM-only iOS devices is the same as configuring AppConnect for iOS. Configuring AppTunnel with HTTP/S tunneling is also the same. For information on configuring AppConnect for iOS, see

"Configuration overview" in the *MobileIron Core AppConnect and AppTunnel Guide*.

When configuring AppConnect for MAM-only iOS devices, consider the following:

- The app check-in interval on the AppConnect global policy determines when AppConnect apps receive updates of their AppConnect global policy, their AppConnect app configuration, and their AppConnect container policy. Because the sync interval on the sync policy has no impact on MAM-only iOS devices, the app check-in interval determines when Mobile@Work does a device check-in with MobileIron Core.
- If you configure Touch ID to access AppConnect apps, use Touch ID with fallback to AppConnect passcode. Touch ID with fallback to device code is not meaningful for MAM-only iOS devices, because you cannot enforce a strong device passcode on the security policy.

Configuring MAM-only Android devices

Configuring MAM-only Android devices requires the following steps:

- Disabling the device administrator on Android devices
- Configuring the security policy for MAM-only Android devices
- Configuring the privacy policy for MAM-only Android devices
- Configuring the sync policy for MAM-only Android devices
- Configuring the lockdown policy for MAM-only Android devices
- Making apps available to MAM-only Android devices
- Using Apps@Work on MAM-only Android devices
- Configuring AppConnect and AppTunnel for MAM-only Android devices

Disabling the device administrator on Android devices

Disabling the device administrator on Android devices is necessary for configuring MobileIron Core to support MAM-only Android devices. This setting is on the Android quick setup policy.

Procedure

1. In the Admin Portal, go to **Policies & Configs > Policies**.
2. Select **Add New > Android > Android Quick Setup**.
3. In the **Name** field, enter a descriptive name for the policy.
4. De-select **Device Administrator**.
5. Click **Save > OK**.

Related topics

"Working with Android Quick Setup policies" in the *MobileIron Core Device Management Guide for Android and Android enterprise Devices*

Configuring the security policy for MAM-only Android devices

Only some settings on the security policy apply to MAM-only Android devices. This procedure explains how to configure the default security policy. However, the same considerations apply to any security policy that you label for Android devices or a subset of Android devices.

NOTE: If you are applying the default security policy or a custom security policy to both MAM-only Android devices and to non-Android devices, including Android enterprise devices, set the appropriate fields for non-Android devices according to your requirements

Procedure

1. In the Admin Portal, go to **Policies & Configs > Policies**.
2. Select the default security policy.
3. Click **Edit**. The Modify Security Policy dialog box opens.
4. The **Password** section does not apply to MAM-only Android devices.
5. In the **Data Encryption** section, set **Device Log Encryption** to **On** if you want to encrypt the log files you email with the **Send Log** option in Mobile@Work for Android.

   NOTE: All other settings in the **Data Encryption** section do not apply to MAM-only Android devices.
6. In the **Android** section, set **Require strict TLS for Apps@Work** if you require strict TLS between Apps@Work and other services.

   NOTE: All other settings in the **Android** section do not apply to MAM-only Android devices.
7. The **Android enterprise, Windows 8.1**, and **Windows 10** sections do not apply to MAM-only Android devices.
8. In the **Access Control** section, in **For All Platforms,** select the compliance action, if any, that you require for each security violation.
9. In the **Access Control** section, in **For Android devices,** select the compliance action, if any, that you require for these security violations, which are the only ones in this section supported for MAM-only Android devices:
   - **when Android version is less than**
   - **when a compromised Android device is detected**
10. Click **Save > OK**.

NOTE: When selecting a compliance action, keep in mind that wipe is not supported for MAM-only Android devices.

Related topics

- MAM-only Android devices
- “Security policies” in *Getting Started with MobileIron Core*

- "Device Log Encryption" in the *MobileIron Core Device Management Guide for Android and Android enterprise Devices*

Configuring the privacy policy for MAM-only Android devices

All Android-related settings on the privacy policy apply to MAM-only Android devices. For information on configuring the privacy policy, see "Privacy policies" in *Getting Started with MobileIron Core*.

Related topics

MAM-only Android devices

Configuring the sync policy for MAM-only Android devices

All Android-related settings on the sync policy apply to MAM-only Android devices. For information on configuring the privacy policy, see "Sync policies" in Getting Started with MobileIron Core.

Related topics

MAM-only Android devices

Configuring the lockdown policy for MAM-only Android devices

The lockdown policy does not apply to MAM-only Android devices. If your MobileIron Core deployment includes only MAM-only Android devices, you can ignore the lockdown policy. However, if your deployment includes other device platforms, including Android enterprise, configure the lockdown policy to meet your requirements.

Related topics

- "Lockdown policies" in *Getting Started with MobileIron Core*
- MAM-only Android devices

Making apps available to MAM-only Android devices

The procedures for making apps available to MAM-only Android device is the same as when Android devices support MDM. However, the following features, available when adding or editing an app in the App Catalog, are not supported:

- Per App VPN settings
- Silent installation

For configuration information, see Adding Google Play apps for Android.

Using Apps@Work on MAM-only Android devices

Using Apps@Work on MAM-only Android devices is the same as it is with Android devices that support MDM.

For information, see Android app versions and device counts.

Configuring AppConnect and AppTunnel for MAM-only Android devices

Configuring AppConnect for MAM-only Android devices is the same as configuring AppConnect for Android. Configuring AppTunnel with HTTP/S tunneling or TCP tunneling is also the same. For information on configuring AppConnect and AppTunnel for Android, see “Configuration overview” in the *MobileIron Core AppConnect and AppTunnel Guide.*