MobileIron ServiceNow Integrator Update Set Guide

MobileIron ServiceNow Integrator Update Set 3.1.0.0

August 27, 2019

Copyright

Contents

Overview

The MobileIron ServiceNow Integrator Update Set installs the MobileIron app which adds a subset of MobileIron functionality to ServiceNow®, an IT Service Management (ITSM) solution. ServiceNow enables IT administration of services and assets for the enterprise, and MobileIron enables enterprise mobility management of mobile devices, applications, and content.

What's new in MobileIron's ServiceNow Update Set version 3.1

This version addresses the issues described in Resolved issues.

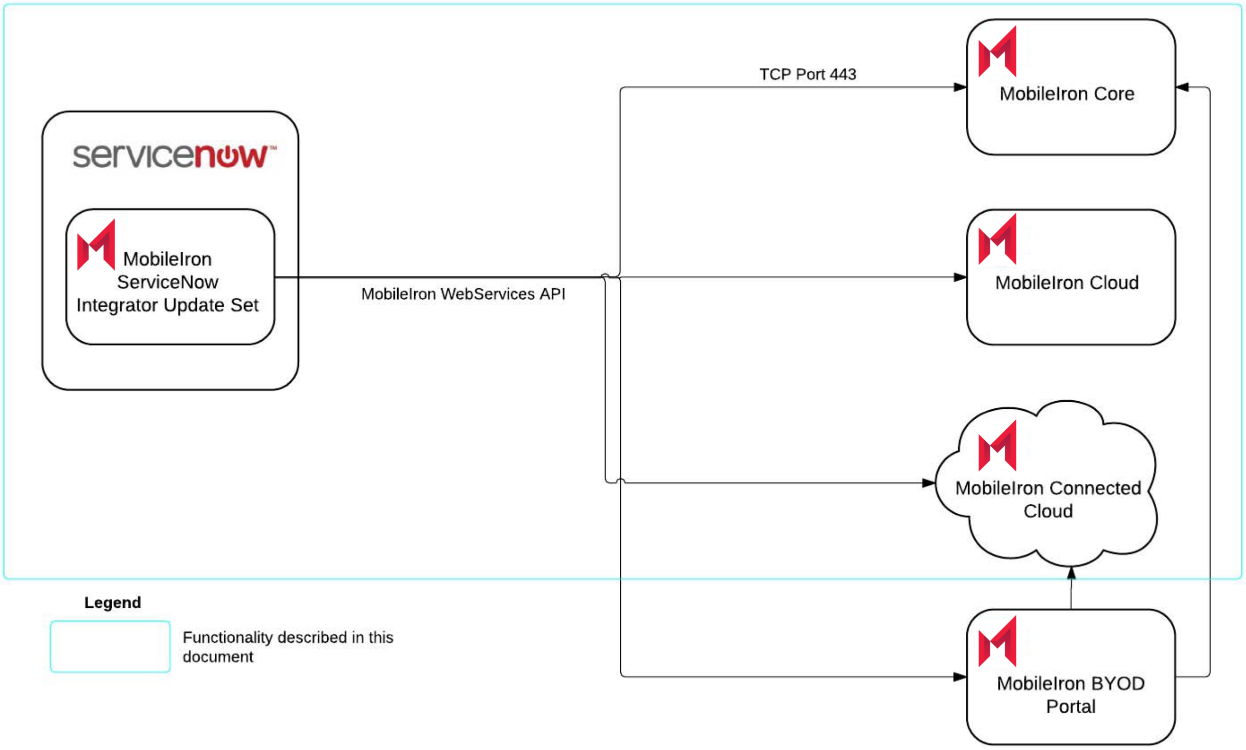
Feature overview

You can use ServiceNow with the MobileIron app installed by the MobileIron ServiceNow Integrator Update Set to:

- Import mobile device information directly from MobileIron Core, MobileIron Connected Cloud, and MobileIron Cloud into ServiceNow's Asset Catalog, and use that data to:
  - View mobile device data — For each device, you can view a configurable list of information, including, but not limited to, name, manufacturer, operating system, OS version, model number, ownership, Core name, MobileIron status, when data last updated, whether blocked, blocked reasons and home operator name.
  - Lock, unlock, wipe, and retire devices, and force devices to check in.
  - Create reports — You can create reports about all the devices in MobileIron Core, MobileIron Connected Cloud, and MobileIron Cloud.
  - Create Homepage charts summarizing any collected device data.
- Self-service
  - Support user mobile device self service **with or without** using the MobileIron BYOD Portal (BYODP) in ServiceNow's Service Catalog. This document does not describe BYODP/ServiceNow functionality. See the BYODPortal.com Product Guide for information about using BYPODP with ServiceNow.
  - Configurable Terms of Service to be presented for new device registrations requests
  - Require manager approval option for new device registrations.
  - Send reminder emails to manager for pending new device registration requests.
  - Set maximum number of devices per user. User exceptions allowed.
  - Set supported platforms for registration.
  - Set supported self service device actions available.
  - When multiple Core appliances are configured, set criteria rules to determine which Core will receive each new device registration request.
- Asset Management
  - Daily automatic purge of retired devices from ServiceNow using the **MobileIron - Delete Retired Devices** script available in ServiceNow at **Integration - MobileIron > Scheduled Import**.

- App Management
  - Import device app inventory for all apps in the App Catalog.
  - Import device app inventory for specified apps.
- Logs
  - Enhanced log messaging to assist in troubleshooting available in ServiceNow at **Integration - MobileIron > Logs**.

The following diagram illustrates ServiceNow integration with the MobileIron ServiceNow Integrator Update Set. ServiceNow interacts with MobileIron Core, MobileIron Connected Cloud, and MobileIron Cloud using the MobileIron API associated with the MobileIron product with which you are using the ServiceNow Update Set.


servicenow
MobileIron ServiceNow Integrator Update Set
MobileIron WebServices API
TCP Port 443
MobileIron Core
MobileIron Cloud
MobileIron Connected Cloud
MobileIron BYOD Portal
Legend
Functionality described in this document

Prerequisite and Setup Overview

The following sections describe the perquisite and offer an overview of the set up and configuration process.

- MobileIron ServiceNow Integrator Update Set prerequisite on page 2
- Overview of the set up and configuration process on page 3

MobileIron ServiceNow Integrator Update Set prerequisite

The MobileIron ServiceNow Integrator Update Set requires a valid SSL certificate for MobileIron Core. If you are using a self signed certificate on MobileIron Core, you will need to add MobileIron Core's certificate as a trusted server certificate in ServiceNow.

Overview of the set up and configuration process

The general process appears below and the following sections describe it in detail:

1. Ensure that you have satisfied the prerequisites described in this chapter.
2. Download the MobileIron ServiceNow Integrator Update Set, as described in Downloading the MobileIron ServiceNow Integrator Update Set on page 5.
3. Install the update set in ServiceNow, as described in Installing the update set in ServiceNow on page 5.
4. Create a MobileIron Core or MobileIron Cloud user whose credentials the ServiceNow Integrator Update set will use to connect to MobileIron, as described in Creating a MobileIron Core or MobileIron Cloud user on page 9.
5. Create a connection to at least one MobileIron Core or MobileIron Cloud, as described in Connecting to and configuring MobileIron Core or MobileIron Cloud on page 10.
6. Optionally, set up direct self-service, as described in Setting up direct self-service on page 15.
7. Optionally, configure App Inventory Tracking Settings, as described in Setting up app inventory tracking settings on page 16.
8. Optionally, specify tracked app devices, as described in Adjusting the data import schedule on page 17.
9. Optionally, add or edit the existing asset query fields to import the corresponding device data from MobileIron into ServiceNow, as described in Changing which MobileIron Core fields ServiceNow imports on page 13.
10. Optionally, adjust the data import schedule, as described in Adjusting the data import schedule on page 17 and Maintaining the update set with scripts on page 18.

Note: To set up MobileIron Connected Cloud, refer to the instructions throughout this guide for MobileIron Core.

Resolved issues

- **SCSI-252:** MobileIron ServiceNow Integrator was not importing serial numbers. The device serial number is now available in the ServiceNow mobile devices section.
- **SCSI-239:** Delete Retired Devices functionality was inconsistent between MobileIron Core and ServiceNow. This release resolves this issue by introducing two attributes, "Delete Devices That Have Been Retired More Than (days)" and "Maximum Retired Devices to Delete in Each Session." To use these new attributes after upgrading to version 3.1, you must run a full import to sync the associated values from MobileIron Core to ServiceNow. See information about the full import script in Connecting to and configuring MobileIron Core or MobileIron Cloud and Adjusting the data import schedule.
- Once Full import job is success, user is able to delete retire devices which are synced from core are deleted based on values/dates set in Instance details.
- **SCSI-238:** ServiceNow delta syncs were not fetching iOS devices if MDM profile was not installed. This release resolves this issue.
- **SCSI-237:** ServiceNow delta syncs were not fetching newly registered assets. This release resolves this issue.

- **SCSI-226:** Missing sorting functionality was creating duplicate and missing device entries. This release resolves the issue by adding sorting functionality, by registration date in ascending order, while fetching devices from MobileIron Core.

Known issues

- **SCSI-240:** "Search app inventory from devices" from "App Inventory Tracking Settings" is not working.

Limitations

- **SCSI-223:** Password length is showing 40 in the Fuji version of ServiceNow. The password length issue is resolved in later ServiceNow releases. MobileIron strongly recommends upgrading to a later ServiceNow version than Fuji.

  Workaround: Change the Password length to 255 in the table MobileIron Instances while adding instance details.

Setting Up ServiceNow Integrator Update Set

The MobileIron ServiceNow Integrator Update Set includes the updates necessary to use the MobileIron Core, MobileIron Connected Cloud, and MobileIron Cloud integration with ServiceNow; and the BYOD Portal integration with ServiceNow. You must have BYOD Portal integrated with Core to utilize the BYOD Portal integration with ServiceNow.

The BYOD Portal is a separate service from MobileIron and using it is not described in this guide. See the Compartmental Product Guide for information about using BYOD Portal with ServiceNow. If you are interested in this service, please contact your MobileIron account representative. If you do not have the BYOD Portal service from MobileIron, you can ignore the BYOD Portal content in this documentation.

Downloading the MobileIron ServiceNow Integrator Update Set

The first step is to obtain the update set. Download it at https://support.mobileiron.com/mi/servicenow-enterprise/current/. MobileIron Support sends your organization an email with credentials to download the update set from the MobileIron download page. Log in to the site with your credentials and download the update set. You upload this update set to your ServiceNow instance in a later section. You should download the fresh install update set and the upgrade install set so your users can install the update set as a fresh install or as an upgrade.

MobileIron's ServiceNow upgrade install will only upgrade from MobileIron's ServiceNow Integrator version 3.0. If you are running an earlier version, upgrade to version 3.0 before running the version 3.1 upgrade installation. To use version 3.1 enhancements after an upgrade, you must run a full import to sync from MobileIron Core to ServiceNow. See information about the full import script in Connecting to and configuring MobileIron Core or MobileIron Cloud and Adjusting the data import schedule.

Installing the update set in ServiceNow

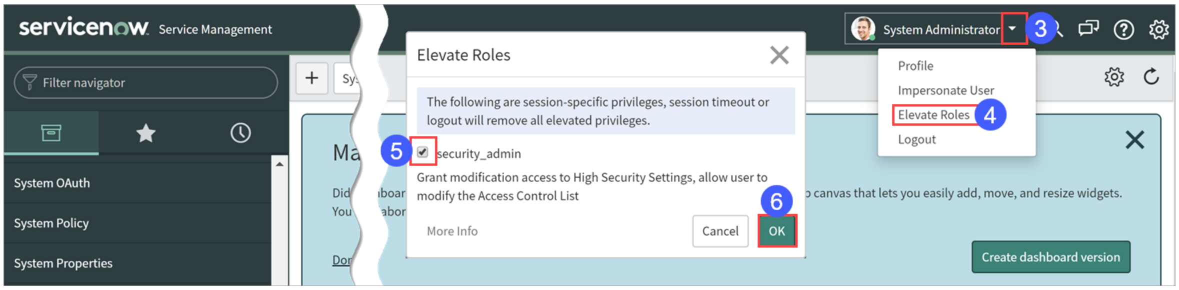
MobileIron recommends installing and testing the MobileIron update set in your development environment before committing it to your production environment. MobileIron provides fresh install update sets and upgrade update sets, therefore, ensure that you select the appropriate update set for your needs, fresh install or upgrade. See Downloading the MobileIron ServiceNow Integrator Update Set for information about obtaining the software.

You upload this update set to your ServiceNow instance in a later section. You should download the fresh install update set and the upgrade install set so your users can install the update set as a fresh install or as an upgrade.

If you are a BYOD Portal customer, the update set will create the proper ServiceNow Service Catalog categories for the BYOD Portal integration, but will not display them in your service catalog by default. You must add the newly-created items to your catalog page.

To install the 3.1 update set, or upgrade from version 3.0 to version 3.1:

1. Extract the downloaded update set on your local machine.
   Note: You need to provide the update set to ServiceNow in XML format because ServiceNow cannot process the zipped file format.
2. Log in as a user with an admin role to the target ServiceNow instance.
3. Elevate the System Administrator role to security_admin by clicking the System administrator user drop-down menu.



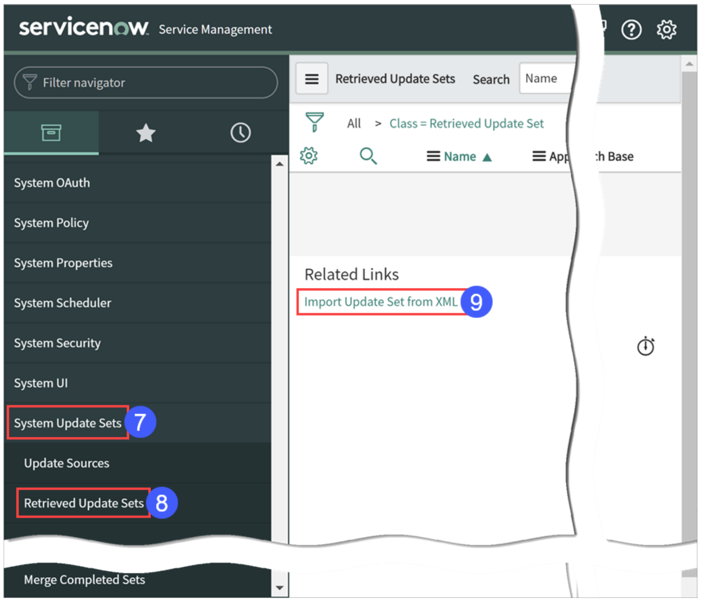
Elevate Roles

The following are session-specific privileges, session timeout or logout will remove all elevated privileges.

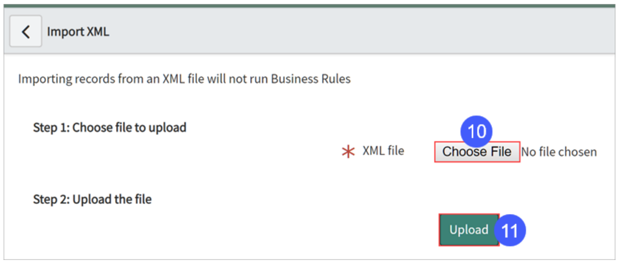
☑ security_admin

Grant modification access to High Security Settings, allow user to modify the Access Control List

4. Select **Elevate Roles**.
5. Place a check mark in the **security_admin** check box.
6. Click **OK**.
7. Navigate to **System Update Sets**.



8. Select **Retrieved Update Sets** in your ServiceNow instance.
9. Click **Import Update Set from XML** at the bottom of the list.


Import XML

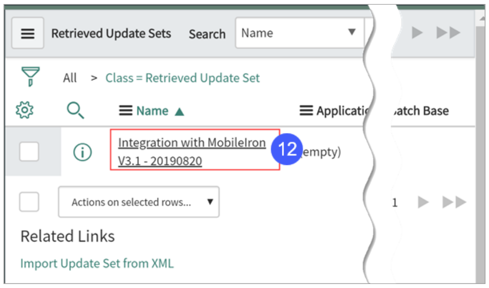
Importing records from an XML file will not run Business Rules

Step 1: Choose file to upload

* XML file Choose File No file chosen

Step 2: Upload the file

10. Click the **Choose File** button to browse to the update set XML file that you extracted in step 1.
11. Click **Upload**.
12. Once the update set uploads, click the retrieved update set to open it.

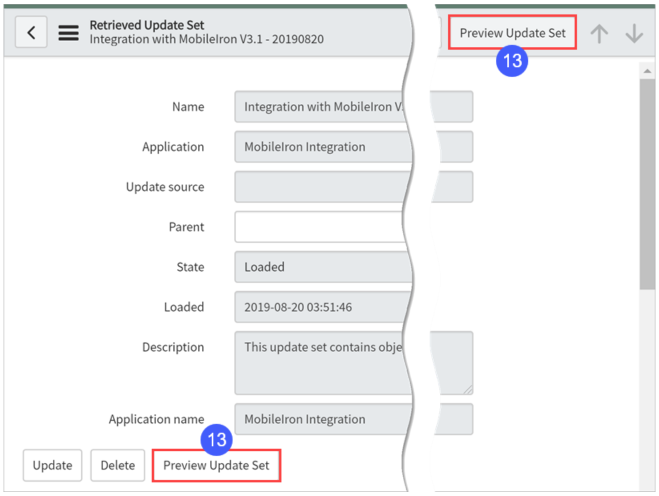


Integration with MobileIron V3.1 - 20190820

Related Links

Import Update Set from XML

13. Click **Preview Update Set**. When the Preview completes, you will see that the import has found an issue.


Retrieved Update Set
Integration with MobileIron V3.1 - 20190820

Name: Integration with MobileIron V

Application: MobileIron Integration

Update source:

Parent:

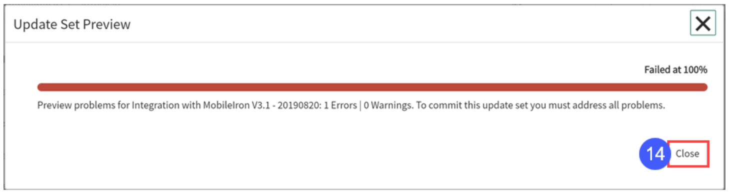
State: Loaded

Loaded: 2019-08-20 03:51:46

Description: This update set contains obje

Application name: MobileIron Integration

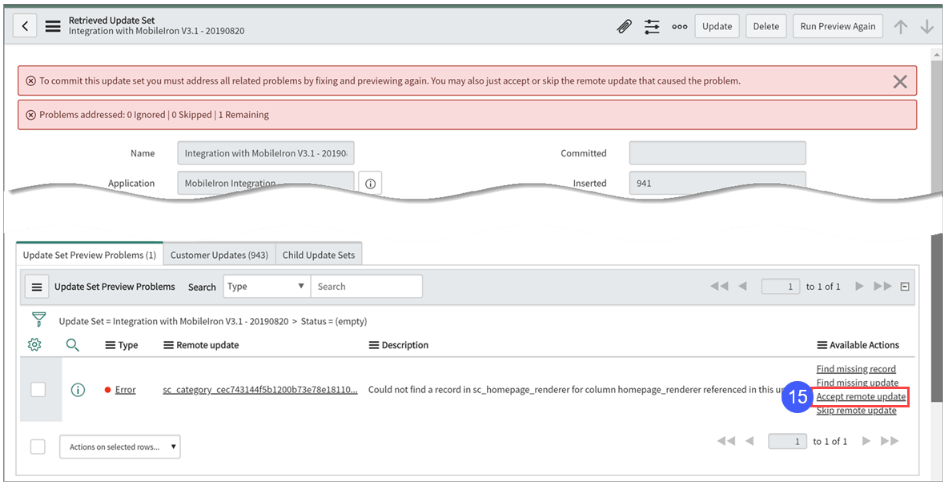
14. Click **Close** to acknowledge the issue and close the dialog box.


Update Set Preview

Failed at 100%

Preview problems for Integration with MobileIron V3.1 - 20190820: 1 Errors | 0 Warnings. To commit this update set you must address all problems.

15. Click **Accept remote update**.



Retrieved Update Set
Integration with MobileIron V3.1 - 20190820

To commit this update set you must address all related problems by fixing and previewing again. You may also just accept or skip the remote update that caused the problem.

Problems addressed: 0 Ignored | 0 Skipped | 1 Remaining

Name: Integration with MobileIron V3.1 - 20190

Application: MobileIron Integration

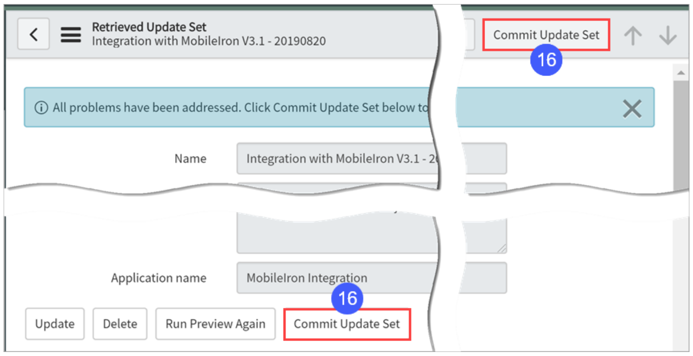
Committed:

Inserted: 941

Update Set = Integration with MobileIron V3.1 - 20190820 > Status = (empty)

Type	Remote update	Description	Available Actions
Error	sc_category_cec743144f5b1200b73e78e18110...	Could not find a record in sc_homepage_renderer for column homepage_renderer referenced in this u	Find missing record Find missing update Accept remote update Skip remote update

16. Click **Commit Update Set** to commit it to your instance. Wait for the progress window to show that the import is complete.



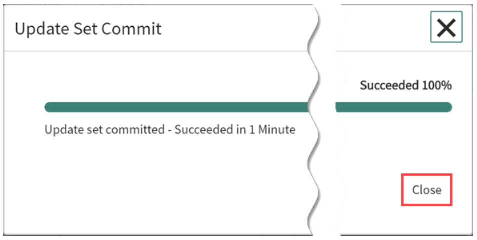
Retrieved Update Set
Integration with MobileIron V3.1 - 20190820

All problems have been addressed. Click Commit Update Set below to

Name: Integration with MobileIron V3.1 - 2

Application name: MobileIron Integration

17. Click **Close**.

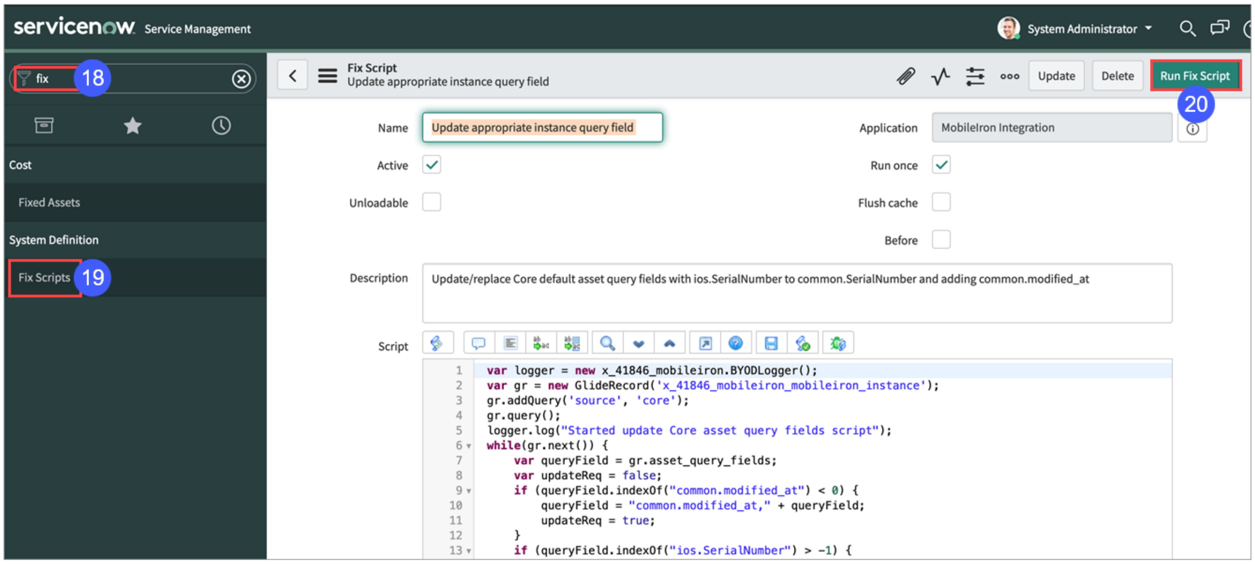


Update Set Commit

Succeeded 100%

Update set committed - Succeeded in 1 Minute

18. If you are performing a fresh install, then skip ahead to Step 21. If you are performing an upgrade installation, search for the **Fix Scripts** option.



Fix Script
Update appropriate instance query field

Name: Update appropriate instance query field

Application: MobileIron Integration

Active ☑

Run once ☑

Unloadable □

Flush cache □

Before □

Description: Update/replace Core default asset query fields with ios.SerialNumber to common.SerialNumber and adding common.modified_at

Script

```
var logger = new x_41846_mobileiron.BYODLogger();
var gr = new GlideRecord('x_41846_mobileiron_mobileiron_instance');
gr.addQuery('source', 'core');
gr.query();
logger.log("Started update Core asset query fields script");
while(gr.next()) {
    var queryField = gr.asset_query_fields;
    var updateReq = false;
    if (queryField.indexOf("common.modified_at") < 0) {
        queryField = "common.modified_at," + queryField;
        updateReq = true;
    }
    if (queryField.indexOf("ios.SerialNumber") > -1) {
```

19. Select the **Fix Scripts** option, and then search for and select the script, **Update appropriate instance query field**.
20. Click **Run Fix Script**. You must run a full sync after running this script. See information about the full import script in Connecting to and configuring MobileIron Core or MobileIron Cloud and Adjusting the data import schedule
21. Configure and use your update set as described starting with the section Creating a MobileIron Core or MobileIron Cloud user on page 9. You may need to refresh your browser to see the new **Integration – MobileIron App application in the ServiceNow application navigator.**

Creating a MobileIron Core or MobileIron Cloud user

You need to create a MobileIron Core or MobileIron Cloud user with the following characteristics:

MobileIron Core	MobileIron Cloud
Create a user in the global space with the following roles: • View dashboard, device page, device details • API	Create a user with the following roles: • Device Read Only • Device Actions

This is an admin user used to integrate to MobileIron Cloud or Core. This is the user you will use to connect to MobileIron Core or MobileIron Cloud from within ServiceNow when you complete the procedure in the section, Connecting to and configuring MobileIron Core or MobileIron Cloud on page 10.

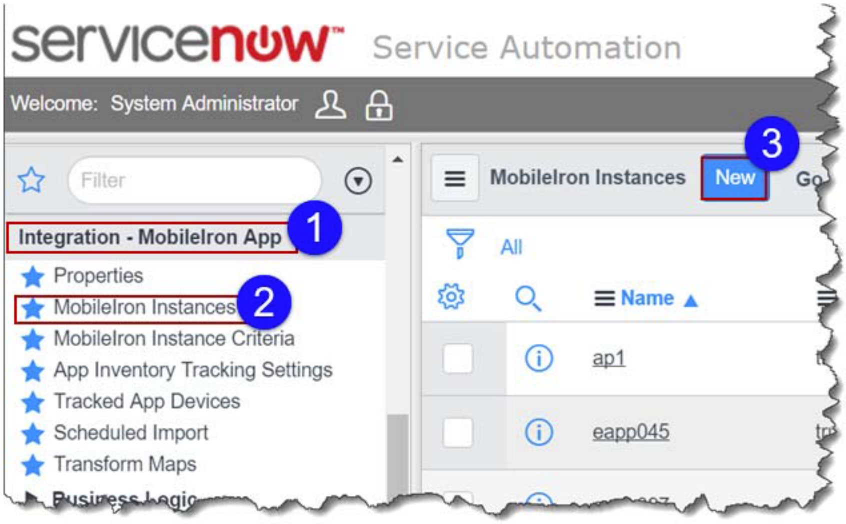
See the MobileIron Core Device Management Guide or the MobileIron Cloud Extended Help for details on how to create this user.

Connecting to and configuring MobileIron Core or MobileIron Cloud

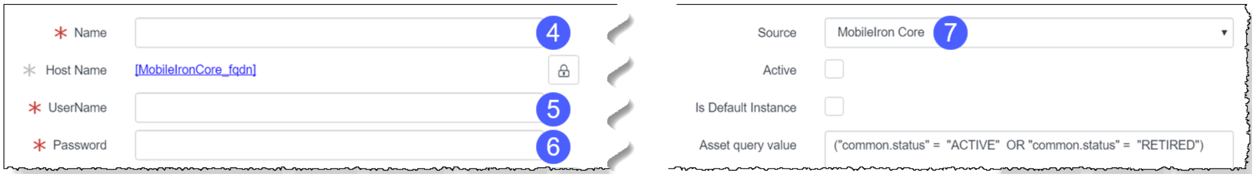
Before you can integrate MobileIron data into ServiceNow, you need to set up a connection between the MobileIron Core or MobileIron Cloud and your ServiceNow instance, so the data can be properly pulled from MobileIron by ServiceNow. The MobileIron ServiceNow integration supports connection to multiple MobileIron Core or MobileIron Cloud instances. You need to connect to at least one instance to collect data.

To create a connection to an instance:

1. In ServiceNow, navigate to **Integration – MobileIron App** in the Application Navigator.
2. Open the **MobileIron Instances** module.
3. Click **New**.



The New MobileIron Instance window appears.

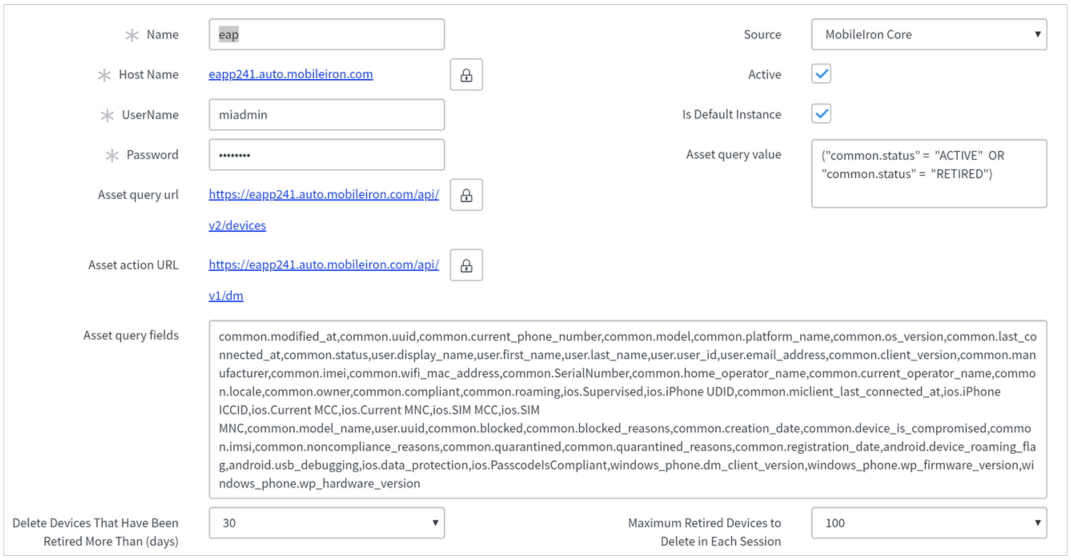


* Name	
* Host Name	[MobileIronCore_fqdn]
* UserName	
* Password	
Source	MobileIron Core
Active	□
Is Default Instance	□
Asset query value	("common.status" = "ACTIVE" OR "common.status" = "RETIRED")

4. Enter the name.
5. Enter the user name.
6. Enter the password.
7. Select the source, **MobileIron Cloud** or **MobileIron Core**.

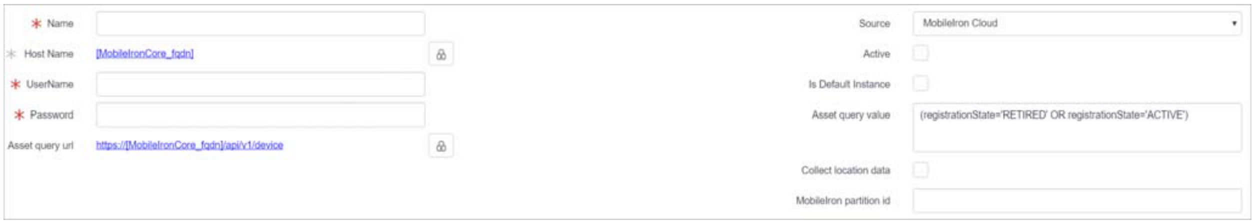
Setting	Description
Name	Create a name for the MobileIron Core so you can differentiate it from any other to which you may MobileIron Core connect. This name is used to display this MobileIron Core on several tables and screens.
Host Name	The hostname of the integrated Core or Cloud serverserver (Core/Cloud) integrated. Eg: app100.auto.mobileiron.com
Username	Enter the user name with the necessary roles that you created in the previous section, Creating a MobileIron Core or MobileIron Cloud user on page 9.
Password	Enter the password for the user with the necessary roles described in the previous section, Creating a MobileIron Core or MobileIron Cloud user on page 9.
Source	Select **MobileIron Cloud** or **MobileIron Core**. The fields on the form change accordingly.

MobileIron Core Settings



* Name	eap	Source	MobileIron Core ▾
* Host Name	eapp241.auto.mobileiron.com	Active	☑
* UserName	miadmin	Is Default Instance	☑
* Password	••••••••	Asset query value	("common.status" = "ACTIVE" OR "common.status" = "RETIRED")
Asset query url	https://eapp241.auto.mobileiron.com/api/v2/devices		
Asset action URL	https://eapp241.auto.mobileiron.com/api/v1/dm		
Asset query fields	common.modified_at,common.uuid,common.current_phone_number,common.model,common.platform_name,common.os_version,common.last_connected_at,common.status,user.display_name,user.first_name,user.last_name,user.user_id,user.email_address,common.client_version,common.manufacturer,common.imei,common.wifi_mac_address,common.SerialNumber,common.home_operator_name,common.current_operator_name,common.locale,common.owner,common.compliant,common.roaming,ios.Supervised,ios.iPhone UDID,common.miclient_last_connected_at,ios.iPhone ICCID,ios.Current MCC,ios.Current MNC,ios.SIM MCC,ios.SIM MNC,common.model_name,user.uuid,common.blocked,common.blocked_reasons,common.creation_date,common.device_is_compromised,common.imsi,common.noncompliance_reasons,common.quarantined,common.quarantined_reasons,common.registration_date,android.device_roaming_flag,android.usb_debugging,ios.data_protection,ios.PasscodeIsCompliant,windows_phone.dm_client_version,windows_phone.wp_firmware_version,windows_phone.wp_hardware_version		
Delete Devices That Have Been Retired More Than (days)	30 ▾	Maximum Retired Devices to Delete in Each Session	100 ▾

MobileIron Cloud Settings

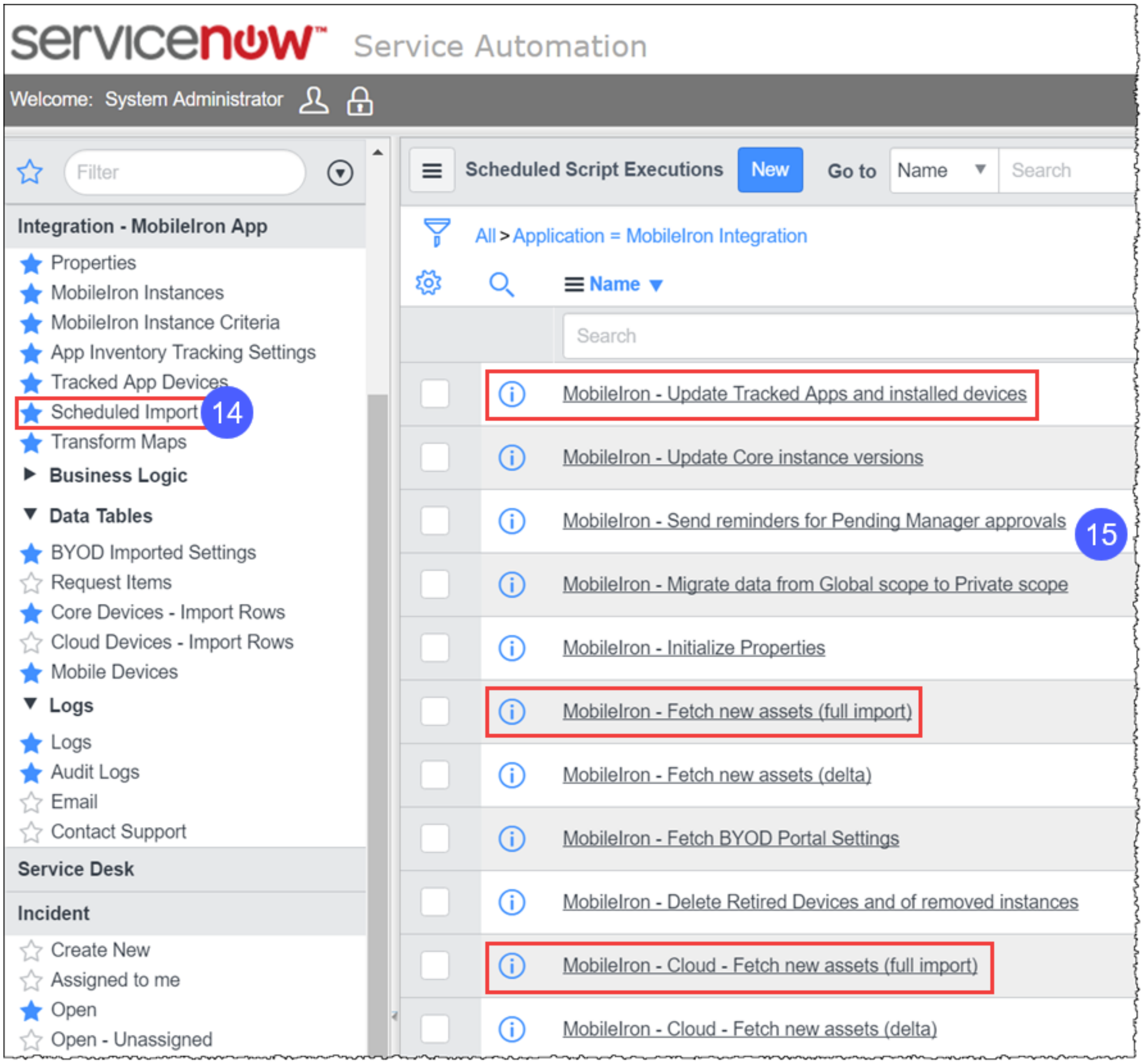


* Name		Source	MobileIron Cloud ▾
* Host Name	[MobileIronCore_fqdn]	Active	□
* UserName		Is Default Instance	□
* Password		Asset query value	(registrationState='RETIRED' OR registrationState='ACTIVE')
Asset query url	https://[MobileIronCore_fqdn]/api/v1/device		
		Collect location data	□
		MobileIron partition id	

8. Enter information into the following fields:

Setting	Description
MobileIron partition id	(MobileIron Cloud only) Auto-populated. You do not need to enter a value.
Asset query URL	(MobileIron Cloud and MobileIron Core) MobileIron Cloud: Click the lock icon to open this field, then enter the URL to your MobileIron Core between the **https://** prefix and the **/api/v1/device** suffix. For example, `na1.mobileiron.com`. This is also known as the Cluster URL. You can find instructions for determining the cluster URL here. MobileIron Core: Click the lock icon to open this field, then enter the URL to your MobileIron Core between the **https://** prefix and the **/api/v2/devices** suffix. For example, `core.company.com` or `m.mobileiron.net/customername`.
Asset query value	(MobileIron Cloud and MobileIron Core) Auto-populated. You do not need to enter a value.
Asset action URL	(MobileIron Core only) Click the lock icon to open this field, then enter the URL to your MobileIron core between the **https://** prefix and the **/api/v1/dm** suffix. For example, `core.company.com` or `m.mobileiron.net/customername`.
Delete Devices That Have Been Retired More Than (days)	(MobileIron Core only) Select the number of days that devices have been retired after which to delete devices.
Maximum Retired Devices to Delete in Each Session	(MobileIron Core only) Select the maximum number of retired devices to delete in each session.

9. If you do not want data from this MobileIron Core pulled during the scheduled job, mark the instance as inactive by removing the check mark from the **Active** check box.
10. If you are connecting multiple instances of MobileIron to ServiceNow, you can click the **Is Default Instance** box to specify the current instance as the default. This means that self-service device registration requests will default to this instance if the device does not match any of the MobileIron Appliance Criteria rules.
11. For MobileIron Cloud, check the **Collect location data** box if you want to be able to use the geo location features on devices. Note that the MobileIron Cloud instance itself may not be collecting location data, so geo location features may not be available even if you check the **Collect location data** box.
12. For MobileIron Core, optionally, configure the asset query fields as described in Changing which MobileIron Core fields ServiceNow imports on page 13.
13. Click **Submit**.
14. When you are ready for the first full import of data from MobileIron Core, select **Integration - MobileIron>Scheduled Import**.
15. Run the following scripts by clicking each one and then clicking **Execute Now** on the resultant screen:
    - **MobileIron - update Tracked Apps and Installed devices**
    - **MobileIron - Fetch new assets (full import)** script for MobileIron Core, or the **MobileIron - Cloud - Fetch new assets** for MobileIron Cloud.



- MobileIron - Update Tracked Apps and installed devices
- MobileIron - Update Core instance versions
- MobileIron - Send reminders for Pending Manager approvals
- MobileIron - Migrate data from Global scope to Private scope
- MobileIron - Initialize Properties
- MobileIron - Fetch new assets (full import)
- MobileIron - Fetch new assets (delta)
- MobileIron - Fetch BYOD Portal Settings
- MobileIron - Delete Retired Devices and of removed instances
- MobileIron - Cloud - Fetch new assets (full import)
- MobileIron - Cloud - Fetch new assets (delta)

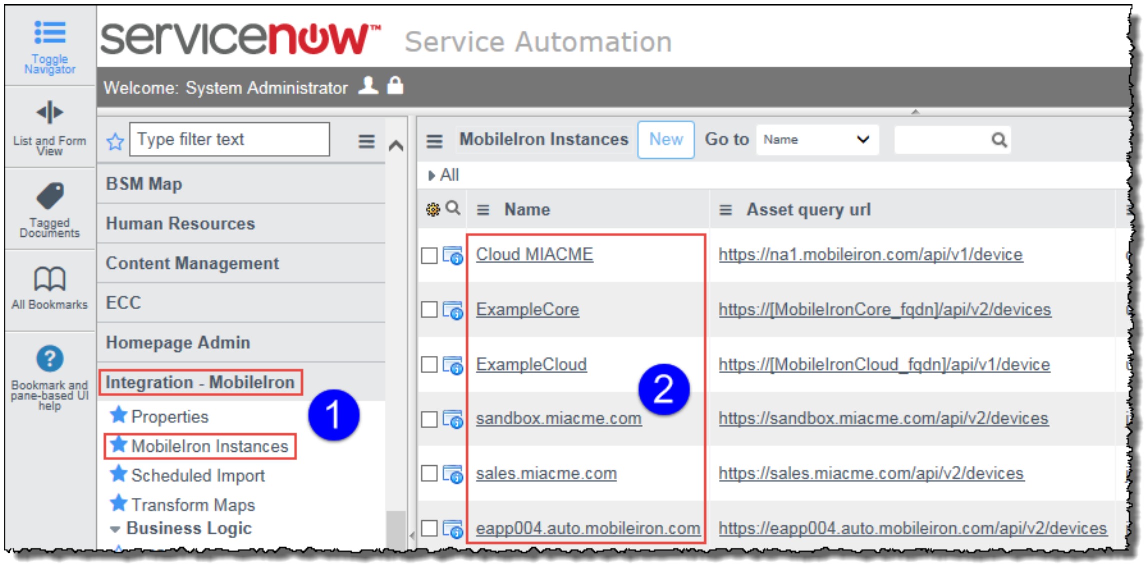
It may take several minutes for device information to appear in ServiceNow.

Changing which MobileIron Core fields ServiceNow imports

The **Asset query fields** field includes the column names imported from the MobileIron Core data. You may need to add fields, for example, if you wish to use the device location feature. This feature uses the database fields **common.location** and **common.location_last_captured_at**, and these fields are not included in the list of default fields that ServiceNow imports. See Default MobileIron Core asset query fields on page 31 for the default list and All MobileIron Core asset query fields on page 33 for a list of all possible fields.

If you need to import additional fields:

1. Click **Integration - MobileIron > MobileIron Instances**.
2. Click the MobileIron Core instance for which to configure imported fields.



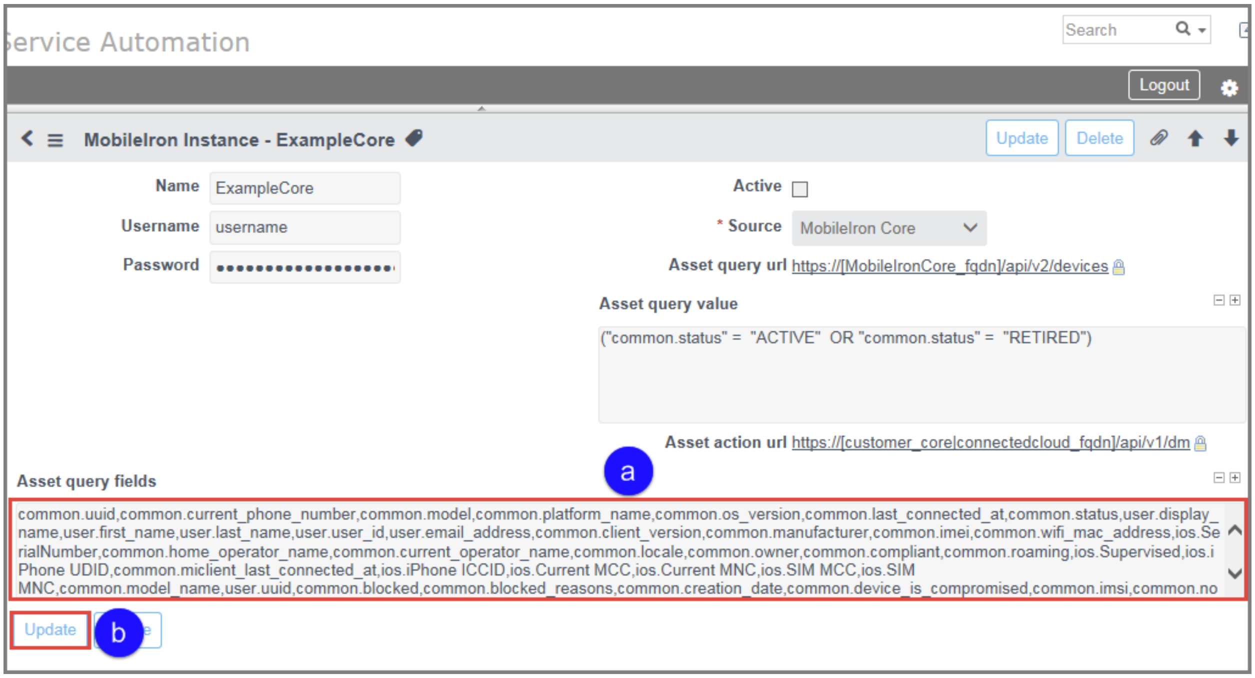
Name	Asset query url
Cloud MIACME	https://na1.mobileiron.com/api/v1/device
ExampleCore	https://[MobileIronCore_fqdn]/api/v2/devices
ExampleCloud	https://[MobileIronCloud_fqdn]/api/v1/device
sandbox.miacme.com	https://sandbox.miacme.com/api/v2/devices
sales.miacme.com	https://sales.miacme.com/api/v2/devices
eapp004.auto.mobileiron.com	https://eapp004.auto.mobileiron.com/api/v2/devices

The **Asset query fields** field appears.

3. Add the additional field names:

   a: Add the field name(s) to the **Asset query fields** field.

   b: Click **Update.**



Name: ExampleCore

Username: username

Password: ••••••••••••••••••

Active □

* Source: MobileIron Core

Asset query url https://[MobileIronCore_fqdn]/api/v2/devices

Asset query value

("common.status" = "ACTIVE" OR "common.status" = "RETIRED")

Asset action url https://[customer_core|connectedcloud_fqdn]/api/v1/dm

Asset query fields

common.uuid,common.current_phone_number,common.model,common.platform_name,common.os_version,common.last_connected_at,common.status,user.display_name,user.first_name,user.last_name,user.user_id,user.email_address,common.client_version,common.manufacturer,common.imei,common.wifi_mac_address,ios.SerialNumber,common.home_operator_name,common.current_operator_name,common.locale,common.owner,common.compliant,common.roaming,ios.Supervised,ios.iPhone UDID,common.miclient_last_connected_at,ios.iPhone ICCID,ios.Current MCC,ios.Current MNC,ios.SIM MCC,ios.SIM MNC,common.model_name,user.uuid,common.blocked,common.blocked_reasons,common.creation_date,common.device_is_compromised,common.imsi,common.no

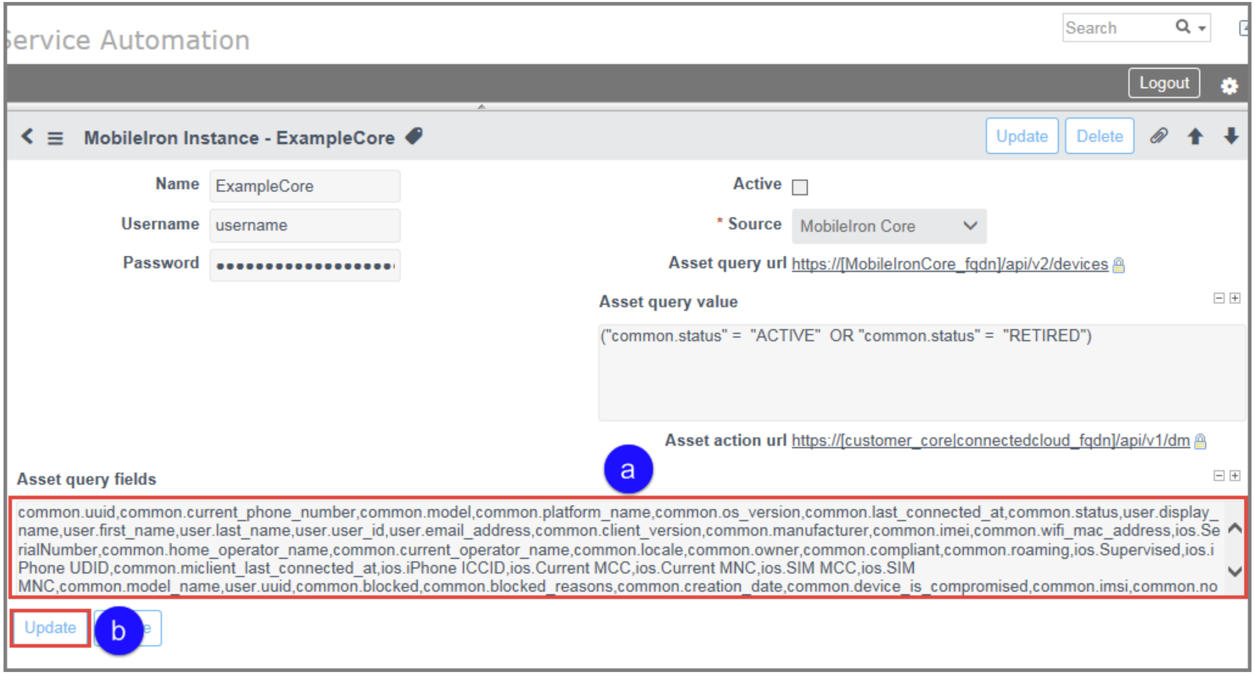
4. Add the field to the transform map so the field from MobileIron Core synchronizes with a specific field in the ServiceNow database, as described in Setting up app inventory tracking settings

Setting up direct self-service

Direct self-service allows users to manage their own MobileIron registered devices from ServiceNow or to request new mobile device registration.

To set up direct self-service:

1. Add the MobileIron instances for which to allow direct self service, as described in Connecting to and configuring MobileIron Core or MobileIron Cloud on page 10.
2. Click **Integration - MobileIron > Properties.**



MobileIron Instance - ExampleCore

Name: ExampleCore

Username: username

Password: ••••••••••••••••••

Active □

* Source: MobileIron Core

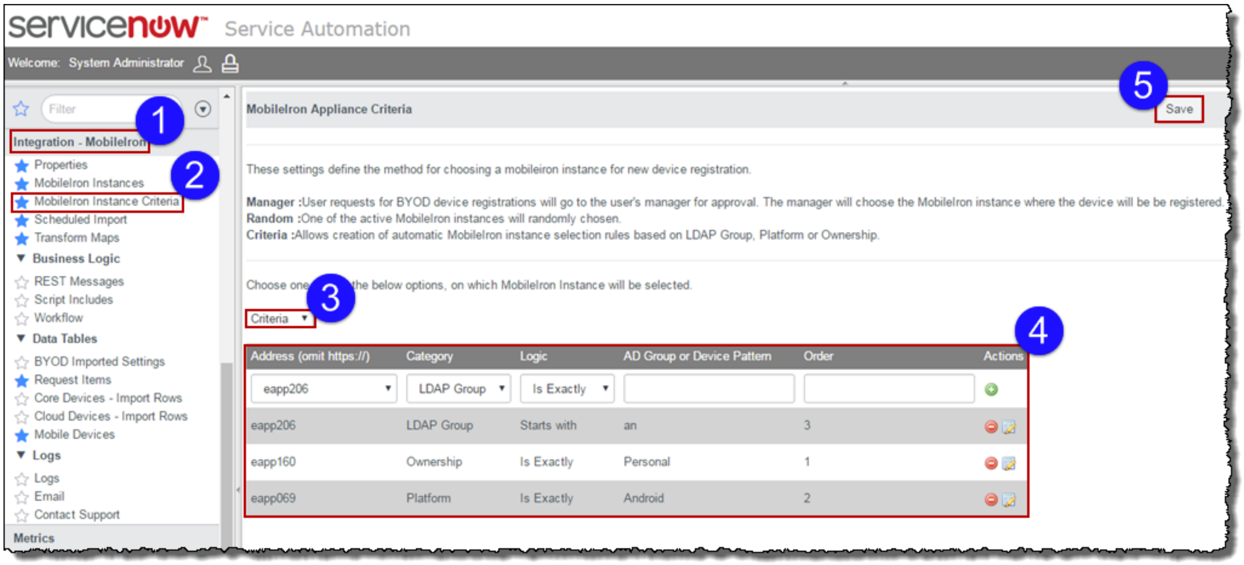
Asset query url: https://[MobileIronCore_fqdn]/api/v2/devices

Asset query value: ("common.status" = "ACTIVE" OR "common.status" = "RETIRED")

Asset action url: https://[customer_core|connectedcloud_fqdn]/api/v1/dm

Asset query fields: common.uuid,common.current_phone_number,common.model,common.platform_name,common.os_version,common.last_connected_at,common.status,user.display_name,user.first_name,user.last_name,user.user_id,user.email_address,common.client_version,common.manufacturer,common.imei,common.wifi_mac_address,ios.SerialNumber,common.home_operator_name,common.current_operator_name,common.locale,common.owner,common.compliant,common.roaming,ios.Supervised,ios.iPhone UDID,common.miclient_last_connected_at,ios.iPhone ICCID,ios.Current MCC,ios.Current MNC,ios.SIM MCC,ios.SIM MNC,common.model_name,user.uuid,common.blocked,common.blocked_reasons,common.creation_date,common.device_is_compromised,common.imsi,common.no

3. Click the **Yes** check box under **Self-Service is enabled**.
4. Click the **Direct Self-Service** radio button under **Self-Service mode**.
5. Use the **Direct Self-Service Properties:** fields to configure the direct self-service behavior.
6. Click **Save**.
7. Click **Integration - MobileIron > MobileIron Instance Criteria.**



MobileIron Appliance Criteria

These settings define the method for choosing a mobileiron instance for new device registration.

Manager :User requests for BYOD device registrations will go to the user's manager for approval. The manager will choose the MobileIron instance where the device will be be registered.
Random :One of the active MobileIron instances will randomly chosen.
Criteria :Allows creation of automatic MobileIron instance selection rules based on LDAP Group, Platform or Ownership.

Choose one of the below options, on which MobileIron Instance will be selected.

Criteria ▾

Address (omit https://)	Category	Logic	AD Group or Device Pattern	Order	Actions
eapp206 ▾	LDAP Group ▾	Is Exactly ▾			⊕
eapp206	LDAP Group	Starts with	an	3	⊖
eapp160	Ownership	Is Exactly	Personal	1	⊖
eapp069	Platform	Is Exactly	Android	2	⊖

8. Use the **Criteria** drop-down menu to configure how to select a MobileIron instance during self-service registration, by manager, randomly, or by selection criteria.
9. Use the fields that appear corresponding to your choice to configure selection criteria further.
   Note: Setting up instance criteria on the **Integration - MobileIron App > MobileIron Instance Criteria page** is only required if you selected Direct Self-Service on the **Integration - MobileIron App > Properties** page.
10. Click **Save**.

Setting up app inventory tracking settings

This feature allows administrators to find the devices on which specific apps are installed. In the settings page described below, the administrator specifies which apps to track.

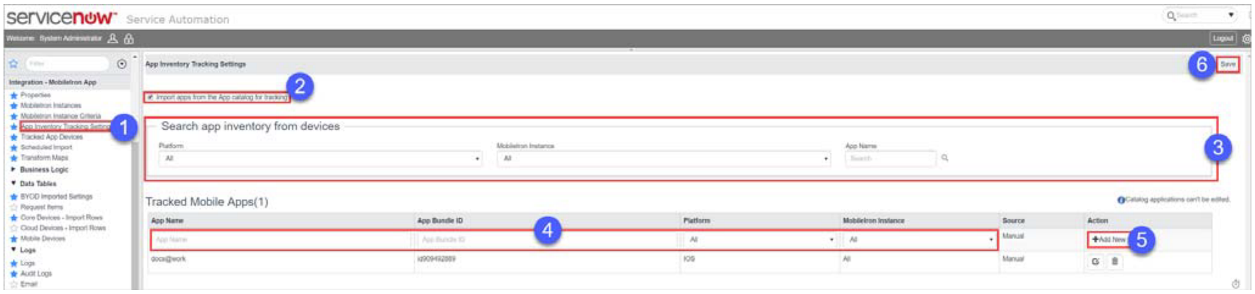
ServiceNow can track the following types of apps:

- All Apps from the app catalog
- Selected apps from the app inventory

After the administrator has selected the apps, a scheduled job fetches associated data.

To set up app inventory tracking:

1. Select **Integration - MobileIron > App Inventory Tracking Settings.**



App Inventory Tracking Settings

☑ Import apps from the App catalog for tracking

Search app inventory from devices

Platform: All
MobileIron Instance: All
App Name: Search

Tracked Mobile Apps(1)

Catalog applications can't be edited.

App Name	App Bundle ID	Platform	MobileIron Instance	Source	Action
App Name	App Bundle ID	All ▾	All ▾	Manual	+Add New
docs@work	id909492889	IOS	All	Manual	

2. Place a check mark in the **Import apps from the App catalog for tracking** check box to track all installed apps in the app catalog.
3. Use the **Search app inventory from devices** fields to specify apps to track which are installed on devices.
4. To add specific apps, use the **Tracked Mobile Apps** field to specify the app.
5. Click **+Add New**.
6. Click **Save**.

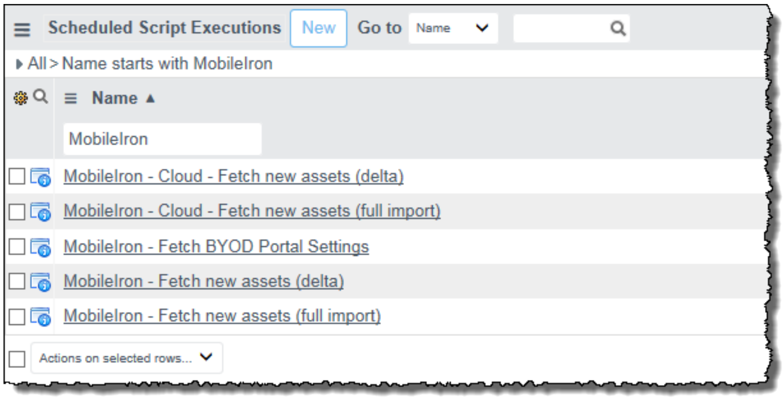
Adjusting the data import schedule

This **Scheduled Import** ServiceNow module contains the import scripts that bring mobile asset data into your ServiceNow instance. We recommend that you import asset data at least every 24 hours, but you can set these import scripts to run at your preference.

There are two different import options for assets: Full Import and Deltas. For your first import, you should import all data using the Full Import scheduled script. After the full import, your future imports can be either just the changed assets since the last import (deltas), or you can do a full import again. You may prefer to import the deltas on a daily basis, and import the full data set on a weekly or monthly basis. The default delta sync schedule is every hour.

To adjust the import schedule:

1. In ServiceNow, select **Integration – MobileIron App** in the Application Navigator, and then select **Scheduled Import**.
   The **Scheduled Script Executions** pane appears.



All > Name starts with MobileIron

Name ▲
MobileIron
MobileIron - Cloud - Fetch new assets (delta)
MobileIron - Cloud - Fetch new assets (full import)
MobileIron - Fetch BYOD Portal Settings
MobileIron - Fetch new assets (delta)
MobileIron - Fetch new assets (full import)

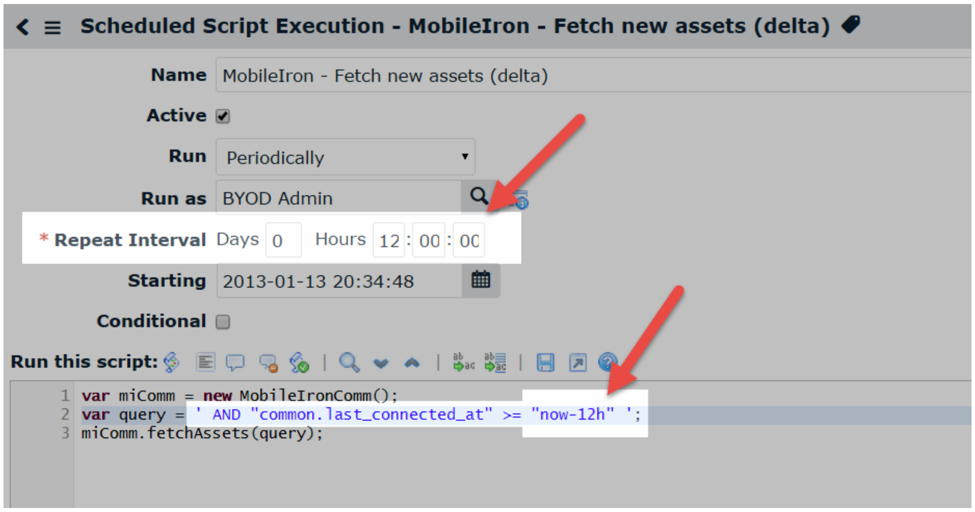
2. Select the desired import schedule to adjust:
   delta

MobileIron Cloud	MobileIron Core
Delta: **MobileIron - Cloud - Fetch new assets (delta)**	Delta: **MobileIron - Fetch new assets (delta)**
Full import: **MobileIron - Cloud - Fetch new assets (full import)**	Full import: **MobileIron - Fetch new assets (full import)**

or **full import**.

3. Change the schedule as desired both in the **Repeat Interval** fields, and also in the script, to ensure the correct interval of data import.

   For example, if you wish to change the MobileIron Core refresh interval to 12 hours for the delta script, you would select the **MobileIron - Fetch new assets (delta) script** and change both the **Repeat Interval** field, as well as the time entry in the script, as shown in the image below.



Scheduled Script Execution - MobileIron - Fetch new assets (delta)

Name: MobileIron - Fetch new assets (delta)

Active ☑

Run: Periodically

Run as: BYOD Admin

* Repeat Interval: Days 0 Hours 12 : 00 : 00

Starting: 2013-01-13 20:34:48

Conditional □

Run this script:

```
var miComm = new MobileIronComm();
var query = ' AND "common.last_connected_at" >= "now-12h" ';
miComm.fetchAssets(query);
```

4. Click **Update**.

Maintaining the update set with scripts

The **Integration – MobileIron App** > **Scheduled Import** option offers scripts that, in addition to importing MobileIron asset information, help you maintain the update set installation.

The **Integration – MobileIron App** > **Scheduled Import** option offers the following scripts:

Script	Description
MobileIron - Delete Retired Devices	The **MobileIron - Delete Retired Devices** script purges retired devices from ServiceNow every 24 hours to keep the system clean of retired device data, however, you can disable this script from running automatically, and you can run it manually if desired. The script also considers the following settings from the Core Instance configuration: • Delete Devices That Have Been Retired More Than (days) • Maximum Retired Devices to Delete in Each Session
Fetch BYOD Portal Settings	The **Fetch BYOD Portal Settings** script fetches BYOD data to sync up data between the BYOD Portal and ServiceNow.
Initialize Self-service Device OS and COPY BYOD Properties	The **Initialize Self-service Device OS and COPY BYOD Properties** script initializes the device list for Self Service.
Send Remainders for BYOD Approvals	The **Send Remainders for BYOD Approvals** script This script: • Gets all pending BYOD requests and pushes them to an array. • Gets all pending approvers from the requests and validates if the approvers have pending BYOD requests. • Prepares a list of the requests that the validated approvers need to approve and raises corresponding events containing the request details that the approvers need to approve.

To run these scripts:

1. In ServiceNow, select **Integration – MobileIron App** in the Application Navigator, and then select **Scheduled Import**.
2. Select the desired script from the resultant **Scheduled Script Executions** pane.

Using ServiceNow Integrator Update Set

You can use the MobileIron ServiceNow Integrator Update Set to:

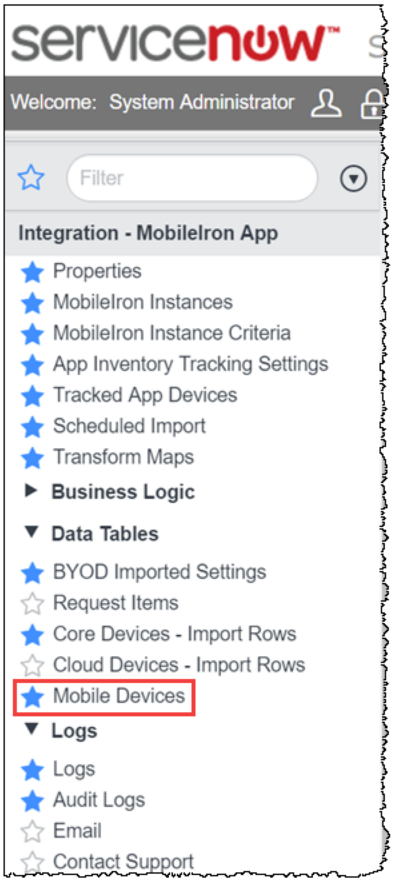
- View mobile device data.
- Take actions on mobile devices, such as lock, unlock, wipe, and retire devices, and force devices to check in.
- Create reports on mobile device data.
- Create Homepage gauges.

Viewing mobile device data

You can view a Mobile Devices grid that displays tabular information about all the devices in the system and you can view details about a specific device.

Viewing information on all mobile devices

To view information about all mobile devices in the system, click **Integration - MobileIron> Mobile Devices**.

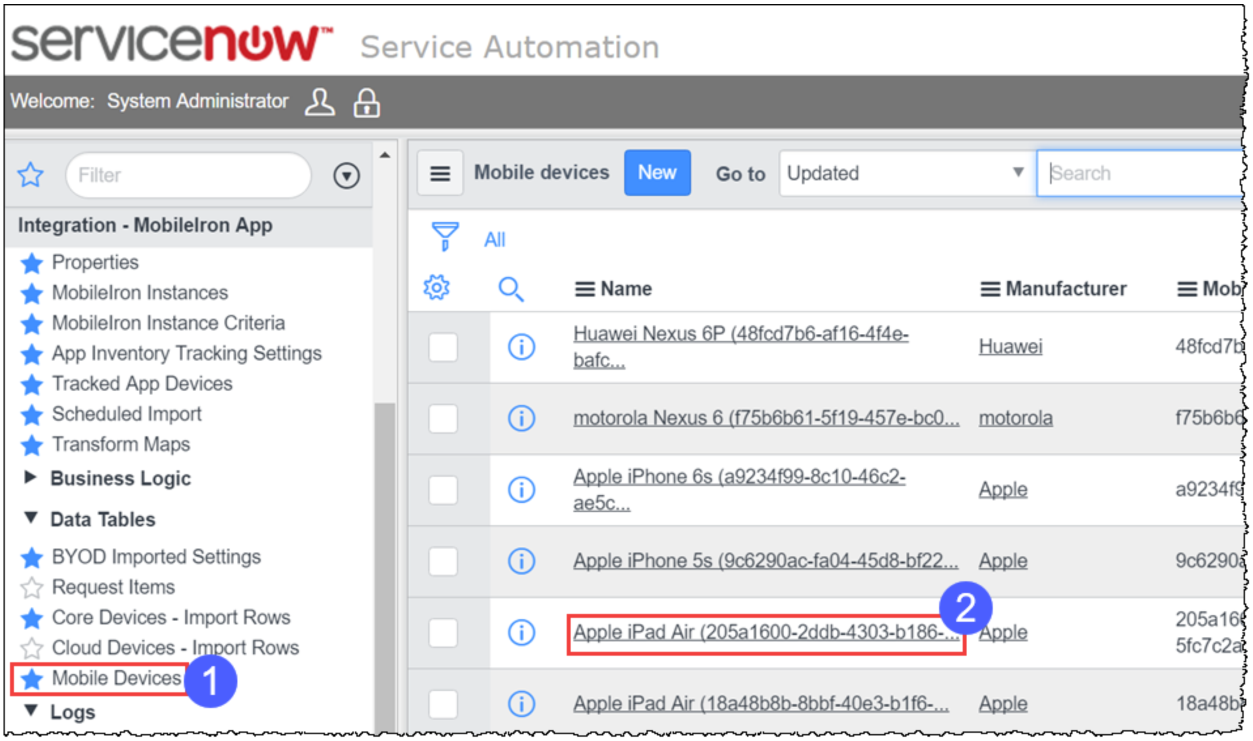


The Mobile Devices grid appears containing a paged list of all mobile devices in the system. Use the grid to sort and search the devices, just as you would with any other ServiceNow results grid.

Viewing information about a specific device

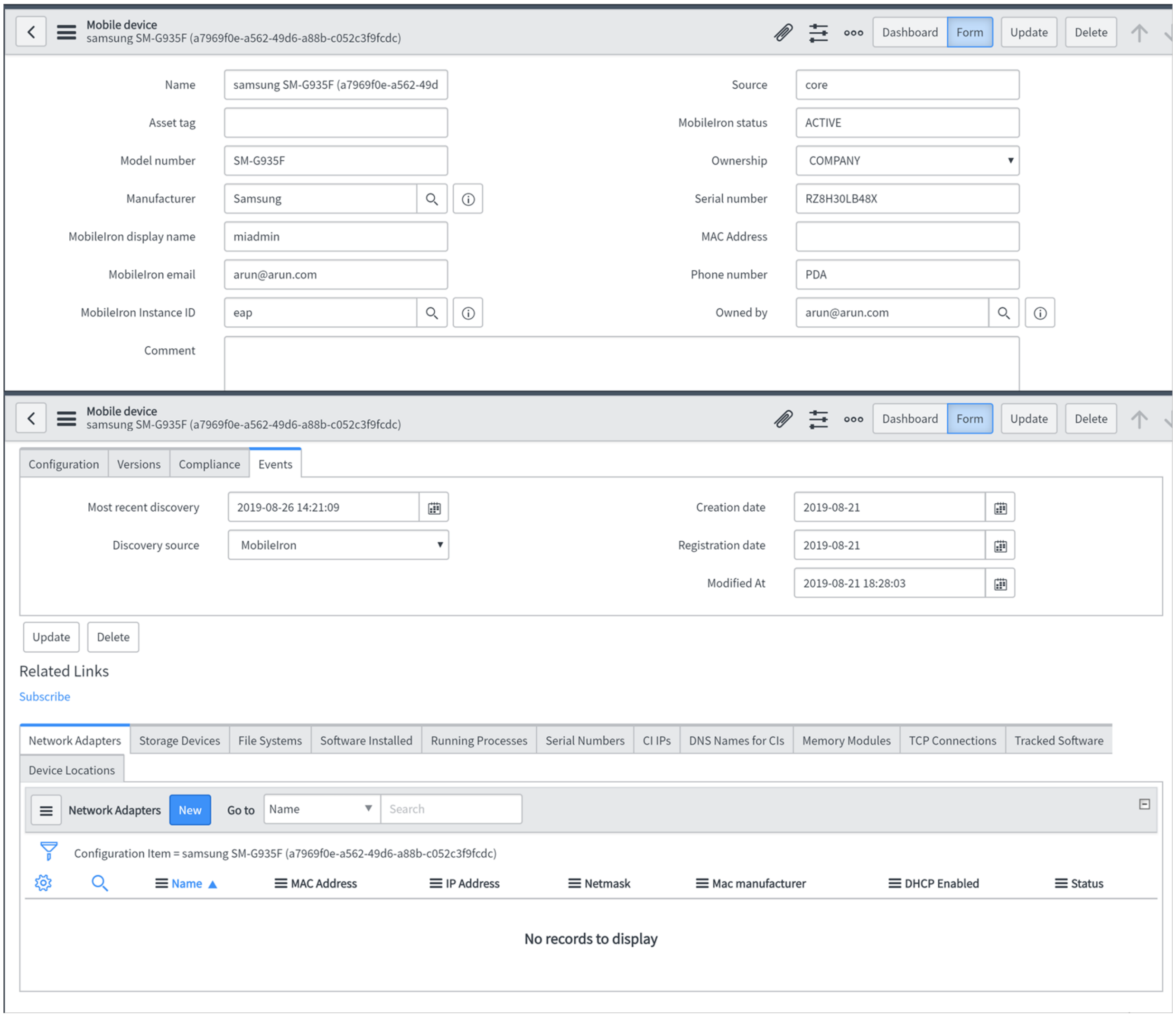
To view information about a specific device:

1. Click **Integration - MobileIron > Mobile Devices**.



Name	Manufacturer	Mob
Huawei Nexus 6P (48fcd7b6-af16-4f4e-bafc...	Huawei	48fcd7b
motorola Nexus 6 (f75b6b61-5f19-457e-bc0...	motorola	f75b6b6
Apple iPhone 6s (a9234f99-8c10-46c2-ae5c...	Apple	a9234f9
Apple iPhone 5s (9c6290ac-fa04-45d8-bf22...	Apple	9c6290a
Apple iPad Air (205a1600-2ddb-4303-b186-...	Apple	205a16 5fc7c2a
Apple iPad Air (18a48b8b-8bbf-40e3-b1f6-...	Apple	18a48b

2. Click the name of the device for which you would like to view device details.
   The Device Details pane appears. The pane displays different fields, depending on whether the device details are from MobileIron Cloud or MobileIron Core.



Mobile device
samsung SM-G935F (a7969f0e-a562-49d6-a88b-c052c3f9fcdc)

Name	samsung SM-G935F (a7969f0e-a562-49d	Source	core
Asset tag		MobileIron status	ACTIVE
Model number	SM-G935F	Ownership	COMPANY
Manufacturer	Samsung	Serial number	RZ8H30LB48X
MobileIron display name	miadmin	MAC Address	
MobileIron email	arun@arun.com	Phone number	PDA
MobileIron Instance ID	eap	Owned by	arun@arun.com
Comment			

Mobile device
samsung SM-G935F (a7969f0e-a562-49d6-a88b-c052c3f9fcdc)

Most recent discovery	2019-08-26 14:21:09	Creation date	2019-08-21
Discovery source	MobileIron	Registration date	2019-08-21
		Modified At	2019-08-21 18:28:03

Related Links
Subscribe

Network Adapters | Storage Devices | File Systems | Software Installed | Running Processes | Serial Numbers | CI IPs | DNS Names for CIs | Memory Modules | TCP Connections | Tracked Software | Device Locations

Network Adapters — Name

Configuration Item = samsung SM-G935F (a7969f0e-a562-49d6-a88b-c052c3f9fcdc)

Name ▲	MAC Address	IP Address	Netmask	Mac manufacturer	DHCP Enabled	Status

No records to display

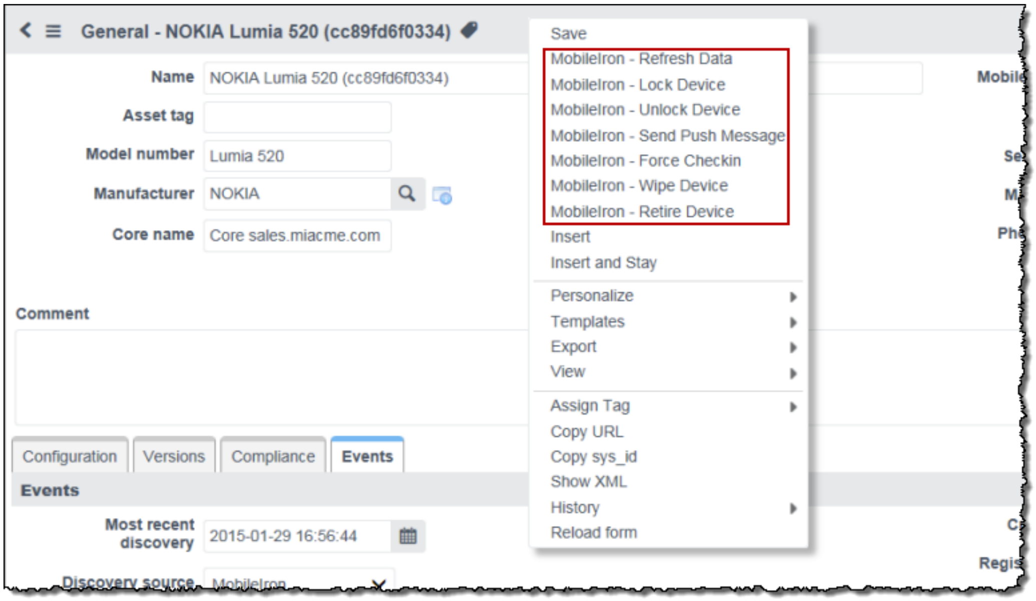
From this pane, you can see all of the data related to this device, including configuration, version, compliance and event information.

Taking actions on a mobile device

There are several MobileIron-specific actions available on the Device Details pane, allowing you to lock, unlock, wipe, retire devices, and force devices to check in.

To take an action on a device:

1. Access the Device Details pane as described in How to view information about a specific device.
2. Right-click the Device Details pane header, and select one of the MobileIron actions.



General - NOKIA Lumia 520 (cc89fd6f0334)

Name: NOKIA Lumia 520 (cc89fd6f0334)
Asset tag:
Model number: Lumia 520
Manufacturer: NOKIA
Core name: Core sales.miacme.com
Comment:
Most recent discovery: 2015-01-29 16:56:44
Discovery source: MobileIron

Refreshing data about a specific device

You can request an update of the data information from the data source, ensuring that the latest information about the device is synchronized between ServiceNow and your MobileIron data.

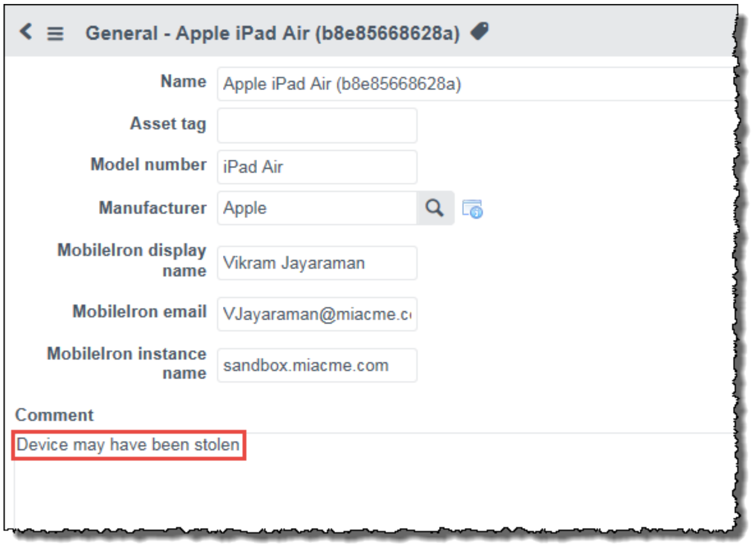
To refresh data about a specific device, follow the instructions in Taking actions on a mobile device on page 22, selecting **MobileIron - Refresh Data**.

Locking a device

You can lock a mobile device, requiring the user to enter a password or PIN to open the device. Settings and support for this feature vary depending on the hardware and OS version of the device.

To lock a specific device:

1. Enter a comment in the **Comment** field describing the reason for locking the device:



General - Apple iPad Air (b8e85668628a)

Name: Apple iPad Air (b8e85668628a)

Asset tag:

Model number: iPad Air

Manufacturer: Apple

MobileIron display name: Vikram Jayaraman

MobileIron email: VJayaraman@miacme.c

MobileIron instance name: sandbox.miacme.com

Comment: Device may have been stolen

2. Follow the instructions in Taking actions on a mobile device on page 22, selecting **MobileIron - Lock Device**.

Unlocking a device

You can unlock a mobile device so a user can continue using it. On iOS and Android devices, this action removes the passcode requirement for the device. For other platforms, unlocking the device without the user's passcode will change the passcode generated by MobileIron. The API returns the unlock passcode for the device, based on the unique device ID.

To unlock a device:

1. Enter a comment describing the reason for unlocking the device in the comment field.
2. Follow the instructions in Taking actions on a mobile device on page 22, selecting **MobileIron - Unlock Device**.

Sending a message to a device

For supported platforms, you can send a push notification that displays on the target mobile device screen. You might use this action to communicate with the authorized user (for example, to let them know that they do not have a compliant passcode), or you might use this action to communicate with somebody who has found a lost device, to give them information about how to return it to the device's owner.

To send a message to a device, follow the instructions in Taking actions on a mobile device on page 22, selecting **MobileIron - Send Push Message**.

Forcing a device to check in

You can require a device to reconnect with the MobileIron server. You might use this action, for example, if a device has been out of connection with the server for a substantial period of time. MobileIron identifies devices that require check in so you can force a check in on those devices.

To force a device to check in, follow the instructions in Taking actions on a mobile device on page 22, selecting **MobileIron - Force Checkin**.

Erasing all apps, data, and settings on a device

You can reset a mobile device and erase all apps, data, and settings on the device. This is known as wiping the device. You might use this action, for example, if a user has lost their device and does not believe they will be able to get it back.

WARNING: This option performs a factory wipe of the device. All data and apps will be erased.

To wipe a device, follow the instructions in Taking actions on a mobile device on page 22, selecting **MobileIron - Wipe Device**.

Retiring a device

You can mark a device as retired both in ServiceNow and in MobileIron, removing the security policy on the mobile device, and removing secure data and information. If a user wants to re-enable the device, they will need to re-enroll the device using the standard procedure.

You may wish to retire a device, for example, as part of your employee off-boarding process to ensure that MobileIron profiles and settings are removed from a departing employee’s personal device.

Note: Retiring a device selectively removes data, configurations and policies applied or made available to the device by MobileIron Core and then removes it from MobileIron control. This does not perform a full factory wipe/reset of the device.

To retire a device, follow the instructions in Taking actions on a mobile device on page 22, selecting **MobileIron - Retire Device**.

Trying to locate a device

The Mobile Device pane contains a **Geo locations** list that displays a history of geo locations identified for the device based on geo location data transmitted to MobileIron. The related list also has a map feature to show you a map view of the geo location point. This list is especially helpful when users report a lost or stolen device. The geo location history can show you the locations of the device during the device’s most recent connections to MobileIron.

Notes:

- This feature uses the database fields **common.location** and **common.location_last_captured_at**, and these fields are not included in the list of default fields that ServiceNow imports. To add these fields, see Changing which MobileIron Core fields ServiceNow imports on page 13.
- The ability to pull device location data into ServiceNow is dependent on MobileIron Core's ability to pull device location. If the Privacy Policy applied to the device is set to not record device location or the user is not allowing sharing of their location data, then ServiceNow will not show any location data.
- MobileIron Core 7.5 returns location data recorded for a device in reverse order, meaning that in ServiceNow, the value for latitude is actually the longitude for the device, and vice versa. This issue invalidates the built-in Google Maps link. This issue does not exist in later versions of MobileIron Core.
- For MobileIron Cloud, you must place a check mark in the **Collect location data** check box to enable device location. See Connecting to and configuring MobileIron Core or MobileIron Cloud on page 10Step 11.

To view geo locations for a specific device, access the Device Details pane as described in "How to view information about a specific device" on page 18, and then examine the Geo locations panel at the bottom of the Device Details pane.

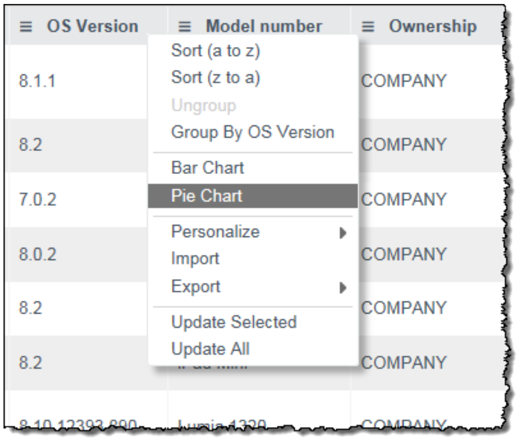
Creating mobile device reports

You can create reports to visualize different aspects of your mobile device data.

To create a mobile device report:

1. In ServiceNow, select **Reports > Create New**.
   The New report pane appears.
2. Use the **Table** field to select the table, **x_cmdb_ci_mobile_device**.
3. Use the remaining fields on the pane just as you would for any other ServiceNow report.
4. Click **Run Report**.
   The report appears.

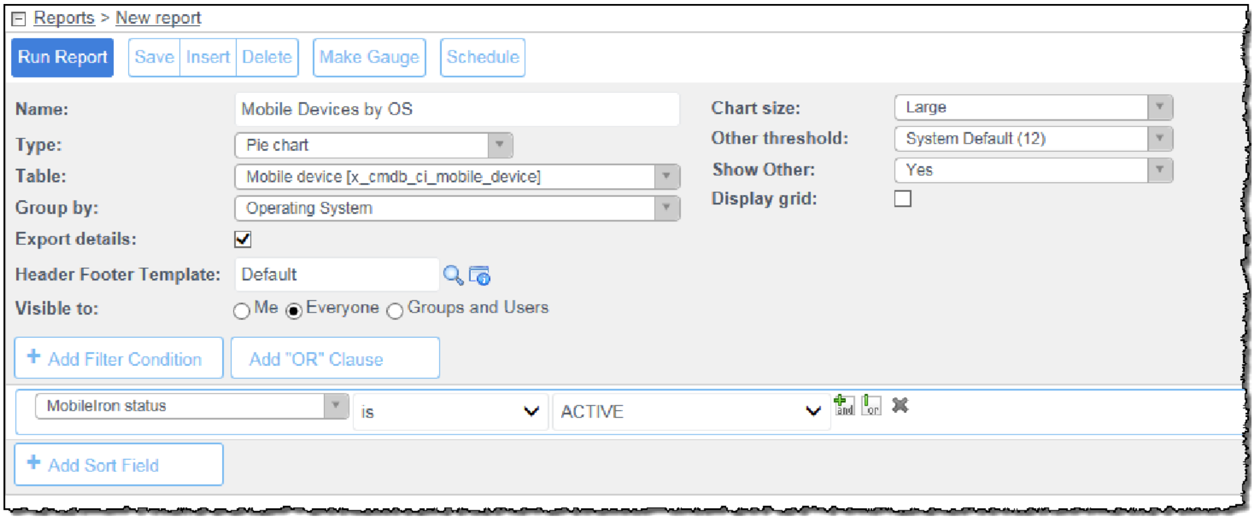
Note: To create a report, you could also right-click a column header while viewing mobile device data, as described in Viewing mobile device data on page 20, and then select Bar Chart or Pie Chart.



≡ OS Version	≡ Model number	≡ Ownership
8.1.1		COMPANY
8.2		COMPANY
7.0.2		COMPANY
8.0.2		COMPANY
8.2		COMPANY
8.2		COMPANY
8.10.12393.890	Lumia 1320	COMPANY

Creating a pie chart of mobile devices by OS

To create a pie chart of mobile devices by OS, follow the instructions in "How to create mobile device reports" on page 24, configuring the New report pane as shown below.



Reports > New report

Run Report | Save | Insert | Delete | Make Gauge | Schedule

Name: Mobile Devices by OS

Type: Pie chart

Table: Mobile device [x_cmdb_ci_mobile_device]

Group by: Operating System

Export details: ☑

Header Footer Template: Default

Visible to: ○ Me ● Everyone ○ Groups and Users

Chart size: Large

Other threshold: System Default (12)

Show Other: Yes

Display grid: □

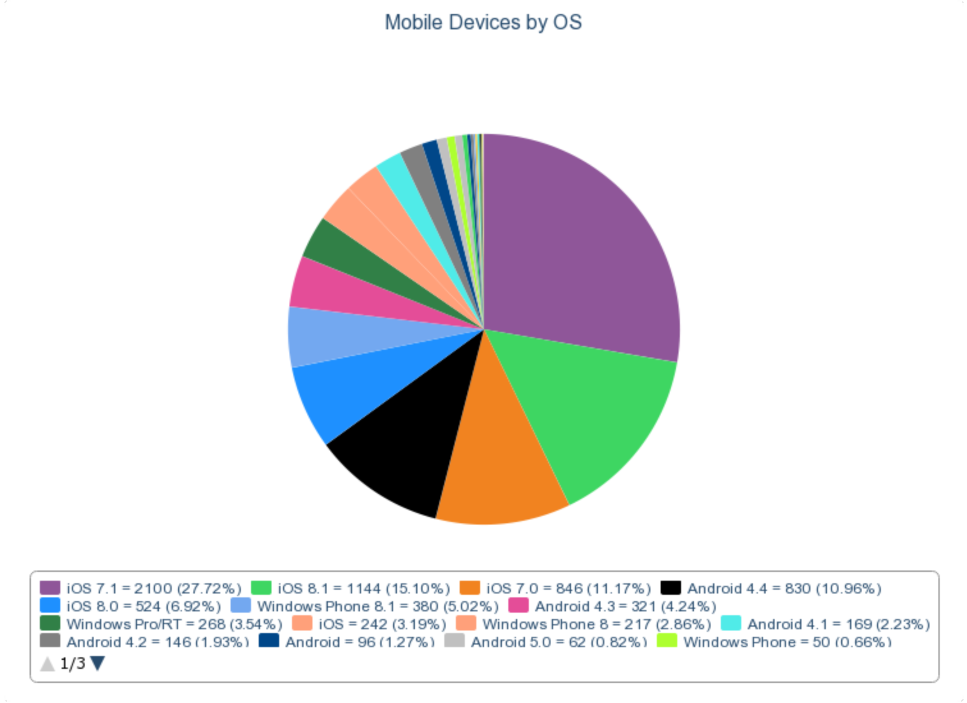
+ Add Filter Condition | Add "OR" Clause

MobileIron status | is | ACTIVE

+ Add Sort Field

Note: A filter condition is added that only includes ACTIVE devices to ensure the gauge does not include retired devices.

The resulting report would look similar to this one:



Adding mobile device gauges to the homepage

You might find it handy to add gauges, in dashboard fashion, to your ServiceNow homepage that display information about the mobile devices in the system.

To add a mobile device gauge to your homepage:

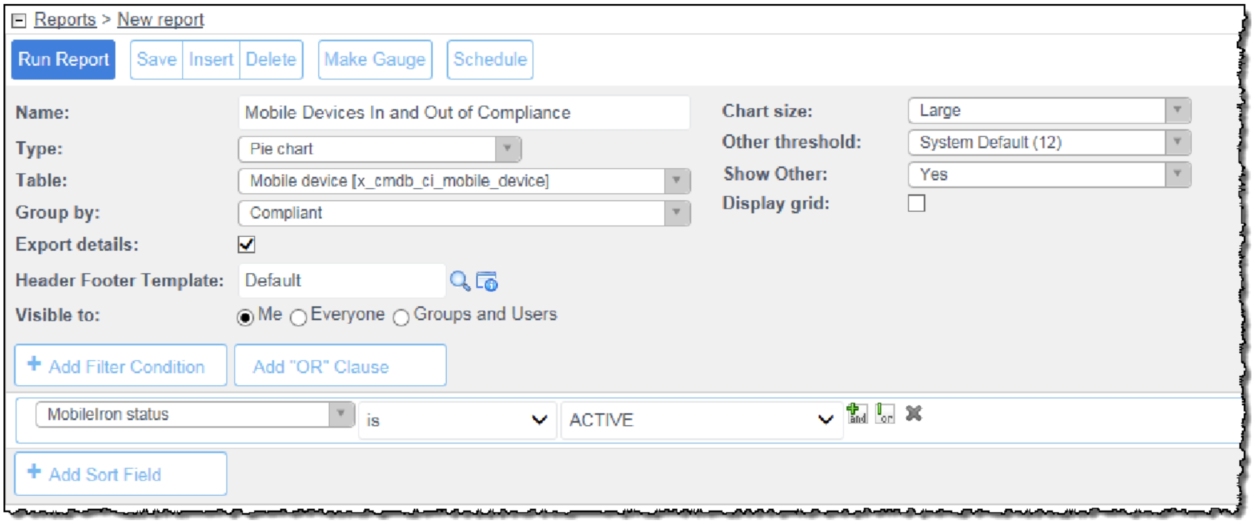
1. Create a mobile device report and preview it before adding it as a gauge to your homepage. Follow the instructions in "How to create mobile device reports" on page 24 to create this report.
2. If the report would suit your purposes as a gauge, click **Make Gauge** and then click **Add to Homepage**.

The following section describes how to add a mobile device gauge to the homepage showing a pie chart of mobile devices in and out of compliance.

Displaying a homepage gauge depicting devices in and out of compliance

To display a homepage gauge depicting devices in and out of compliance:

1. Follow the instructions in "How to create mobile device reports" on page 24, configuring the New report pane as shown below.



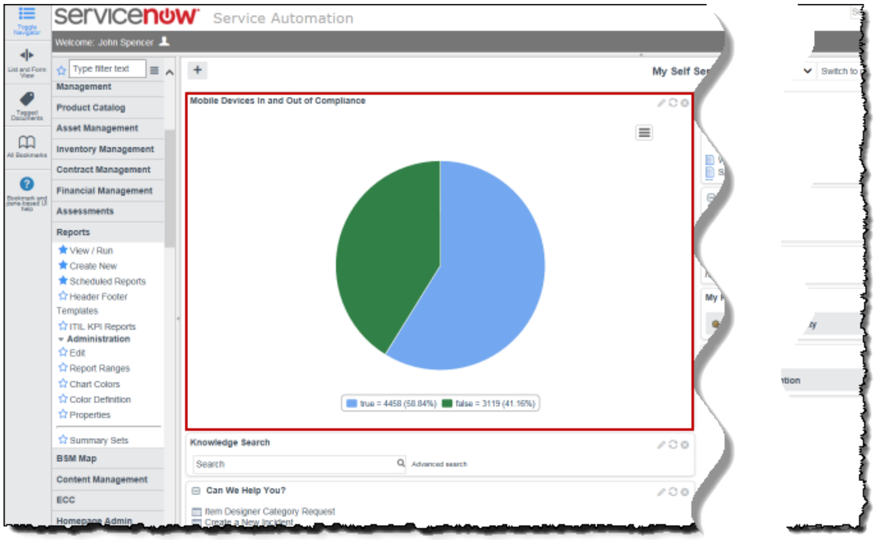
Name: Mobile Devices In and Out of Compliance
Type: Pie chart
Table: Mobile device [x_cmdb_ci_mobile_device]
Group by: Compliant
Export details: ☑
Header Footer Template: Default
Visible to: ● Me ○ Everyone ○ Groups and Users
Chart size: Large
Other threshold: System Default (12)
Show Other: Yes
Display grid: □

MobileIron status | is | ACTIVE

Note: A filter condition is added that only includes ACTIVE devices to ensure the gauge does not include retired devices.

2. Click **Make Gauge** and then click **Add to Homepage**.

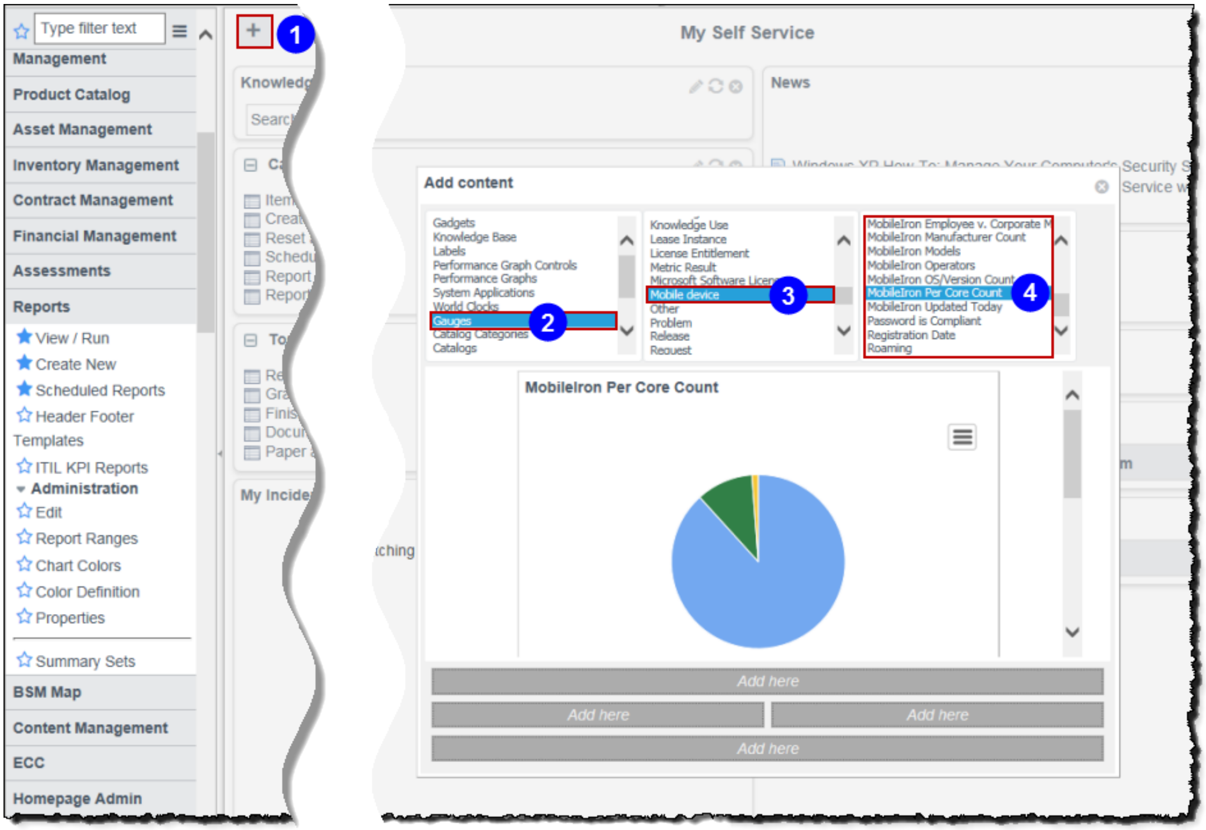
   The gauge appears on your homepage.



Find your saved mobile device gauges

To find your saved mobile device gauges:

1. Click the **Add content** icon on the homepage.



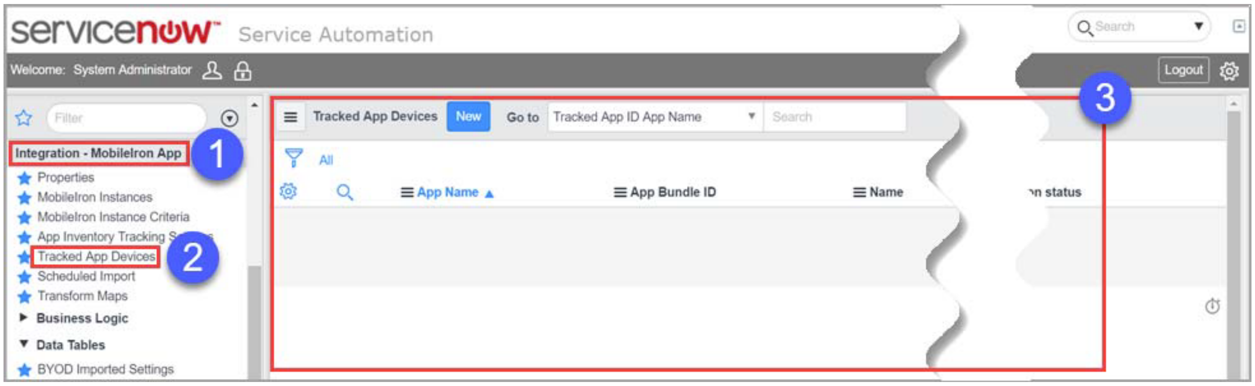
2. In the resultant Add Content window, select **Gauges**.

3. Select **Mobile device**.
4. Select a saved mobile device gauge from the list on the far right of the Add content window.

Viewing tracked apps

To view tracked app devices:

1. Navigate to **Integration - MobileIron App**



2. Select **Tracked App Devices**.
3. View the tracked app data in the pane on the right.

NOTE: The New button does not actually add new data to the app. Do not use it for this release.

Support Information

Default MobileIron Core asset query fields

This section lists the fields imported from MobileIron Core into ServiceNow by default. See Changing which MobileIron Core fields ServiceNow imports on page 13 for more information.

ServiceNow imports the following fields by default:

android.device_roaming_flag

android.samsung_dm

android.usb_debugging

common.blocked

common.blocked_reasons

common.client_version

common.compliant

common.creation_date

common.current_operator_name

common.current_phone_number

common.device_is_compromised

common.home_operator_name

common.imei

common.imsi

common.last_connected_at

common.locale

common.manufacturer

common.miclient_last_connected_at

common.model

common.model_name

common.modified_at

common.noncompliance_reasons

common.os_version

common.owner

common.platform_name

common.quarantined

common.quarantined_reasons

common.registration_date

common.roaming

common.SerialNumber

common.status

common.uuid

common.wifi_mac_address

ios.Current MCC

ios.Current MNC

ios.data_protection

ios.iPhone ICCID

ios.iPhone UDID

ios.PasscodeIsCompliant

ios.SIM MCC

ios.SIM MNC

ios.Supervised

user.display_name

user.email_address

user.first_name

user.last_name

user.user_id

user.uuid

windows_phone.dm_client_version

windows_phone.wp_firmware_version

windows_phone.wp_hardware_version

All MobileIron Core asset query fields

This section lists all the fields that ServiceNow can import from MobileIron Core. See Changing which MobileIron Core fields ServiceNow imports on page 13 for more information.

These are all the fields that can be imported from MobileIron Core by ServiceNow:

android.admin_activated

android.brand

android.device_roaming_flag

android.knox_version

android.mdm_enabled

android.media_card_capacity

android.media_card_free

android.samsung_dm

android.usb_debugging

common.blocked

common.blocked_reasons

common.client_version

common.compliant

common.creation_date

common.current_country_code

common.current_country_name

common.current_operator_name

common.current_phone_number

common.device_admin_enabled

common.device_is_compromised

common.home_operator_name

common.imei

common.imsi

common.ip_address

common.language

common.last_connected_at

common.locale

common.location

common.location_last_captured_at

common.manufacturer

common.mdm_managed

common.memory_capacity

common.memory_free

common.miclient_last_connected_at

common.model

common.model_name

common.modified_at

common.noncompliance_reasons

common.os_version

common.owner

common.platform

common.platform_name

common.quarantined

common.quarantined_reasons

common.registration_date

common.registration_imsi

common.retired

common.roaming

common.security_state

common.Serialnumber

common.status

common.uuid

common.wifi_mac_address

ios.Current MCC

ios.Current MNC

ios.data_protection

ios.iOSBackgroundStatus

ios.iPhone ICCID

ios.iPhone UDID

ios.IsDeviceLocatorServiceEnabled

ios.PasscodeIsCompliant

ios.PasscodeIsCompliantWithProfiles

ios.PasscodePresent

ios.ProductName

ios.SIM MCC

ios.SIMMNC

ios.Supervised

user.display_name

user.email_address

user.first_name

user.last_name

user.user_id

user.uuid

windows_phone.dm_client_version

windows_phone.wp_firmware_version

windows_phone.wp_hardware_version

MobileIron Cloud device fields

activationLockEnabled

cellularTechnology

clientLastCheckin

clientVersion

cloudBackupEnabled

complianceState

Core name

currentCarrierNetwork

currentMcc

currentMnc

deviceModel

displayName

easDeviceIdentifiers

emailAddress

firstName

Groups

Iccid

Update set properties and configurations

You may use the information in this section to help understand the integration settings and troubleshoot any issues.

- Tables
- Script includes
- System properties

Tables

The integration includes the following tables in ServiceNow:

- MobileIron Core [x_mobileiron_core]
- Geo Location [x_geo_location]
- Is MobileIron Devices [x_is_mobileiron_devices]
- Mobile device [x_cmdb_ci_mobile_device]
- MobileIron Instance Criteria [x_mobileiron_appliance_criteria]
- MobileIron Self-Service Properties [x_mobileiron_selfservice_properties]

MobileIron Core [x_mobileiron_core]

This new table contains the connection and query information to retrieve data from the MobileIron Core or MobileIron Cloud.

Column label	Column name	Type	Description
Active	[x_active]	True/False	When true, shows that a core is set to active, and data will be imported from the core on the set schedule
Asset action URL	[x_asset_action_url]	URL	URL for passing asset actions
Asset query fields	[x_asset_query_fields]	String	Fields to pull from the Asset record in MobileIron
Asset query URL	[x_asset_query_url]	URL	URL for getting asset information
Asset query value	[x_asset_query_value]	String	The actual query being passed to pull in the correct fields from MobileIron
Collect location data	[x_collect_location_data]	True/False	When checked, location data will be imported from the MobileIron instance if available.
Delete Devices That Have Been Retired More Than (days)	delete_retired_devices	Integer	Integer value will be considered as age of retired devices and consumed by delete retired devices script as per schedule and criteria
Maximum Retired Devices to Delete in Each Session	max_retired_devices_per_session	Integer	Integer value will be considered as devices limit per session and consumed by delete retired devices script as per schedule and criteria.
MobileIron partition id	[x_mobileiron_partition_id]	String	Identifies the MobileIron Partition ID for the cloud instance. This field will auto-populate when ServiceNow successfully authenticates to the MobileIron instance.
Name	[x_name]	String	Then name of the core, set when the core record is created

Column label	Column name	Type	Description
Password	[x_password]	Password (2-way encrypted)	The password for connecting to the core
Source	[x_api]	Choice	Identifies the integration type
Username	[x_username]	String	The username for connecting to the core

Geo Location [x_geo_location]

This custom table contains geo location data for mobile devices so you can see that information for a particular device. The fields **common.location** and **common.location_last_captured_at** must be added to the list of fields imported by ServiceNow, as described in Changing which MobileIron Core fields ServiceNow imports on page 13, for location history to be recorded.

Column label	Column name	Type	Description
Configuration Item	[x_configuration_item]	Reference	References the device for which the geo location record is being stored
Last captured	[x_last_capured]	Date/Time	The date that this geo location was most recently captured for this device
Latitude	[x_latitude]	String	The latitude recorded by the mobile device when the record was updated
Longitude	[x_longitute]	String	The longitude recorded by the mobile device when the record was updated
Map	[x_map]	URL	A URL to open the location in a mapping tool
Number	[x_number]	String	This is the unique ID value for the table and is incremented automatically when a new Geo location record is created
TempDeviceID	[x_tempdeviceid]	String	Contains Device ID, in case device is not yet imported into ServiceNow.

Is MobileIron Devices [x_is_mobileiron_devices]

This table contains a list of MobileIron devices that have been added to ServiceNow. This table extends the Import Set Row table.

This table does not contain any unique columns; all columns come from the Import Set Row table, but it is a location to store these records permanently, grouping these MobileIron devices in a single table.

Mobile device [x_cmdb_ci_mobile_device]

This table extends the Computer table. It contains the following unique fields, all of which are being imported from MobileIron:

Column label	Column name	Type
Product	[x_product]	Reference
Blocked	[x_blocked]	True/False
Blocked reasons	[x_blocked_reasons]	String
Comment	[x_comment]	Journal
Compliant	[x_compliant]	True/False
Core name	[x_core_name]	String
Creation date	[x_creation_date]	Date
Data protection	[x_data_protection]	Choice
Device is compromised	[x_device_is_compromised]	True/False
Device roaming flag	[x_device_roaming_flag]	True/False
DM client version	[x_dm_client_version]	String
Home operator name	[x_home_operator_name]	String
IMEI	[x_imei]	String
IMSI	[x_imsi]	String
Last geo capture	[x_last_geo_capture]	Date/Time
Locale	[x_locale]	String
MobileIron client version	[x_mobileiron_client_version]	String
MobileIron display name	[x_mobileiron_display_name]	String
MobileIron email	[x_mobileiron_email]	String

Column label	Column name	Type
MobileIron first name	[x_mobileiron_first_name]	String
MobileIron last name	[x_mobileiron_last_name]	String
MobileIron status	[x_mobileiron_status]	String
MobileIron user id	[x_mobileiron_user_id	String
MobileIron UUID	[x_mobileiron_uuid]	String
Modified At	modified_at	Date/Time
Noncompliance reasons	[x_noncompliance_reasons]	String
Ownership	[x_ownership]	Choice
Passcode is compliant	[x_passcode_is_compliant]	True/False
Phone number	[x_phone_number]	String
Quarantined	[x_quarantined]	String
Quarantined reasons	[x_quarantined_reasons]	String
Registration date	[x_registration_date]	Date
Roaming	[x_roaming]	True/False
USB debugging	[x_usb_debugging]	True/False
WP firmware version	[x_wp_firmware_version]	String
WP hardware version	[x_wp_hardware_version]	String

MobileIron Instance Criteria [x_mobileiron_appliance_criteria]

This table Contains MobileIron Instance selection criteria:

Column label	Column name	Type
Address	[x_address]	Reference
Category	[x_category]	Choice
Logic	[x_logic]	Choice
Order	[x_order]	Integer
Value	[x_value]	String

MobileIron Self-Service Properties [x_mobileiron_selfservice_properties]

This table Contains system properties, when self-service option is choosen. It contains the following unique fields, all of which are being imported from MobileIron:

Column label	Column name	Type
AndroidAllowed	[x_androidallowed]	True/False
iOSAllowed	[x_iosallowed]	True/False
LocateAllowed	[x_locateallowed]	True/False
LockAllowed	[x_lockallowed]	True/False
NumberOfDevices	[x_numberofdevices]	String
OSXAllowed	[x_osxallowed]	True/False
RegisterAllowed	[x_registerallowed]	True/False
RetireAllowed	[x_retireallowed]	True/False
Terms	[x_terms]	String
UnLockAllowed	[x_unlockallowed]	True/False
WindowsAllowed	[x_windowsallowed]	True/False
WipeAllowed	[x_wipeallowed]	True/False

Business rules

The following business rule is included in the update set:

Version	Business Rule	Table	Description
2.0	Populate Google Map URL	x_geo_location	This business rule creates a URL that will display the specific geo location on a Google map page
2.0	Get Cloud Partition ID	x_mobileiron_core	To get partition id for cloud instances
2.0	Set Cloud Asset action url	x_mobileiron_core	Update asset action URL for cloud instance
2.3	Ensure default instance exists	x_mobileiron_core	This rule will ensure that at least 1 default instance exists.
2.3	Ensure IsDefault is only one	x_mobileiron_core	This rule will unmark previous default instances, when another instance is marked as default

Version	Business Rule	Table	Description
2.3	Prevent change quantity for BYOD Item	Sc_req_item	To prevent changing quantity of a BYOD Request
2.3	Prevent delete mobileiron default instance	x_mobileiron_core	This rule will not allow deleting a default MobileIron instance.
2.3	Update BYOD Approval Req Prop for a Req	Sc_req	This rule will update “BYOD Approval Required” property for Catalog requests. This will be used to ensure MobileIron instance is mandatory, while manager is approving the requests.
2.3	Update Instance For BYOD Devices	Sc_req_item	This rule will populate MobileIron instance for Device Registration Requests based on VSP selection criteria.

Script includes

The following script includes are included in the update set:

Name	Description
MobileIronComm	The main communication interface to the MobileIron API
BYODLogger	Quick class to handle logging for the BYOD Portal integration
BYODParser	Class that handles the parsing both the retrieved settings as well as the device registration response
GenerateBYODToken	Generate a digested token for the BYOD integration
MobileIron Parser	Class that handles the parsing both the retrieved settings as well as the device registration response
MobileIronClientComm	Consists of various functions, which can be called from client-side
MobileIronApplianceCriteria	Consists of CRUD functions related to MobileIron Instance selection criteria
MobileIronSettings	This will initialize the settings, when the administrator chooses either Direct Self-service or through BYOD Portal.

System properties

The following system properties are either created or modified as part of the update set:

Name	Description
com.MI.byod.account	The account name that the company is registered under in the BYOD Portal cloud (that is, mycompany.byodportal.com would have an account name of mycompany)
com.MI.byod.api_password	The BYOD api password to authenticate to the basic auth protected endpoint for BYOD Portal
com.MI.byod.api_username	The BYOD api username to authenticate to the basic auth protected endpoint for BYOD Portal
com.MI.byod.debug	Enable debug logging
com.MI.byod.mobileiron_action_host	URL to the MobileIron asset action api
com.MI.byod.mobileiron_api_password	Basic authentication password for the MobileIron api
com.MI.byod.mobileiron_api_username	Basic authentication username for the MobileIron api
com.MI.byod.mobileiron_query	Query value for a MobileIron asset query
com.MI.byod.mobileiron_query_fields	The fields to return when performing a MobileIron asset query
com.MI.byod.mobileiron_query_host	URL to the MobileIron asset query api
com.MI.byod.mobileiron_query_liMIt	The limit to the number of records returned from a MobileIron asset query
com.MI.byod.secret	The shared secret used to generate the digest. This secret should be the same as the one specified in your BYOD application
com.MI.byod.url	The URL to BYOD Portal. Do NOT change unless you are using this with BYOD Portal on-premise.
com.MI.byod.user_table_field	Field from the user table from which to pull the user data
com.MI.BYODRemainders.Days	Number of days to wait, after which approval remainder will be sent to manager.
com.MI.BYODRemainders.Enabled	Indicates whether Managers should receive email, in case registration requests are not approved within configured number of days.

Name	Description
com.MI.MaxDevices.Exception	List of comma-separated email ids, for which maximum devices exception will be applied.
com.MI.selfservice.enabled	Indicates whether self-service is enabled or not.
com.MI.ManagerApproval.enabled	Indicates whether Manager approval is enabled or not.
com.mobileiron.VSPSelection	Indicates MobileIron instance selection criteria mode, Possible values are Manager,Random and Criteria.
com.MI.byod.enabled	Indicates whether self-service is enabled through BYOD Portal or not.
com.MI.byod.debug	Indicates whether Debugging is enabled or not.
com.MI.byod.url	URL of configured BYOD Portal
com.MI.byod.account	Account name used to access BYOD Portal API.
com.MI.byod.user_table_field	Column name from sys_user table in ServiceNow, which represents name of the user. Default value is "user_name"
com.MI.byod.secret	Secret of configured BYOD Portal
com.MI.byod.mobileiron_query_limit	Query Limit to use, which fetching devices through API calls.