Product Provisioning

VMware Workspace ONE UEM 2109

You can find the most up-to-date technical documentation on the VMware website at:

https://docs.vmware.com/

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

1 Product Provisioning 6

2 Relay Servers 8

Configure a Relay Server 12

Configure a Relay Server Cloud Connector 16

Relay Server Cloud Connector (RSCC) Hardware Requirements and Installation 20

Pull Service Based Relay Server Configuration 22

Configure .NET Core Pull Relay Server, PS v2.0 31

3 Device Staging 38

Stage Devices With the Enrollment Configuration Wizard 39

Generate a QR Code with the Enrollment Configuration Wizard, Android 40

Generate a Barcode Staging Package with the Enrollment Configuration Wizard 41

Generate a Sideload Staging Package with the Enrollment Configuration Wizard, Android and Windows Rugged 43

Enroll with Web Enrollment, Windows Rugged 44

Stage Devices With a Manually Created Staging Package 44

Add a Manifest to Your Staging Package 45

Wi-Fi Profiles for Staging 47

Barcode Staging 47

Generate a Barcode Staging Package 48

Enroll Zebra Devices with Stage Now Barcode, Android 49

Enroll Honeywell Devices with Staging Barcode, Android 51

How Do You Configure a Zebra Work Managed Device with Stage Now, Relay Servers, and CICO Launcher 53

Sideload Staging Packages 57

Generate a Sideload Staging Package with the Configuration Wizard 58

Enroll Zebra and Motorola Devices with Sideload Staging, Android 59

Install a Sideload Staging Package, WinRugg 59

Enroll Honeywell Devices with Sideload Staging, Android 60

Enroll Platform OEM Devices with Sideload Staging, Android 61

Use the Sideload Staging Utility, WinRugg 62

AirWatch CAB Creator, WinRugg 64

4 Products 68

Create a Product 71

Configure a CDN for Provisioning 76

Product Provisioning Profiles 80

Application Provisioning, Android 82
Product Conditions 84
Event Actions, Android and WinRugg 94
Files-Actions for Products 99
Create a Files-Actions Component 100
Files-Actions Manifest Options for Android 102
Files-Actions Manifest Options for macOS 103
Files-Actions Manifest Options for QNX 104
Files-Actions Manifest Options for Windows Desktop 105
Files-Actions Manifest Options for WinRugg 106
Editing Files-Actions 107
Delete Files-Actions 108
Import Motorola Packages in Files-Actions, Android and WinRugg 108
Create an XML Provisioning File, Android, Win7, WinRugg 108
Workspace ONE Intelligent Hub Upgrading a File-Action 110
Upload the Workspace ONE Intelligent Hub APF File, Upgrade File-Action 110
RunIntent Action, File-Action Android 111
Upgrade the OS, File-Action for Android 115
Create an OS Upgrade File-Action, Android 116
Create an OS Upgrade for Zebra Devices, Android 8.0+ 118
Upgrade the OS, File-Action for WinRugg 120
Product Sets 125
Custom Attributes 129
Custom Attributes, Android 143
Custom Attributes, macOS 144
Custom Attributes, Win7 146
Custom Attributes, WinDesk 147
Custom Attributes, WinRugg 149
What Happens If You Change a Product, Component, or Condition? 151

5 Product Management 154
Products Dashboard 156
View Information About Products in Device Details 158
Product Job Statuses 159

6 Device Management 163
Configure Settings, QNX 164
Enterprise Reset a Rugged Device, Android and WinRugg 167
Enable AirWatch Cloud Messaging, Android Rugged 169
Platform OEM Service, Android Provisioning 169
Batch (BAT) File Guidelines, Win7 and WinDesk 170

7 Lookup Values 173

Product Provisioning

1

Workspace ONE UEM powered by AirWatch enables you to create and deliver products, which contain device profiles, applications, and platform-specific installation instructions and conditions. These products follow a set of rules, schedules, and dependencies as guidelines for ensuring your devices remain up-to-date with the content they need.

Product provisioning encompasses the use of a relay server, which is an FTP(S), SFTP, or HTTPS server designed to work as a go-between for devices and the UEM console. Create these servers for each store or warehouse to store content to distribute to your devices.

Another product provisioning feature is the staging methods of enrollment. Depending on the platform and device manufacturer, you can perform sideload and barcode staging that quickly enrolls a device and downloads the Workspace ONE Intelligent Hub, Wi-Fi profile, and any other important content.

As this guide focuses on the functionality provided by product provisioning, it does not contain all the features and functionality that Workspace ONE UEM offers for managing devices of any specific platform.

Visit docs.vmware.com and search for the Platform Guide of your choice. For more information on general MDM and UEM console functionality, see the **Managing Devices Documentation** and **Console Basics Documentation** available on docs.vmware.com.

Supported Devices, OS, and Agents, Product Provisioning

The product provisioning functionality in Workspace ONE UEM supports different devices and operating systems. The available functionality changes based on the supported rugged device.

Workspace ONE UEM supports product provisioning for devices with the following operating systems.

Android

- Android Legacy devices running Android 4.4 (Kit Kat) and later with Workspace ONE Intelligent Hub.
- Android Enterprise Work Managed devices running Workspace ONE Intelligent Hub.
- macOS 10.7 Lion+ devices:

macOS

- MacBook Pro
- MacBook Air
- Mac Mini
- iMac
- Mac Pro

QNX Devices

- QNX 6.5 devices.

Windows Desktop and Windows Rugged

- Windows CE 5, 6, and 7.
- Windows Mobile 5.x/6.1/6.5 (Professional and Standard).
- Windows Embedded 6.5.
  - Motorola and Zebra Windows Rugged devices require the Rapid Deployment Client v2.0+ for barcode staging.
- Windows 10 devices with Workspace ONE Intelligent Hub installed.

Relay Servers

2

Relay servers act as a content distribution node that provides more bandwidth and data use control in Workspace ONE UEM. Relay servers act as a proxy between the Workspace ONE UEM server and the rugged device for product provisioning.

Source Server Versus Relay Server

A source server is the original location of the data, usually a database, or content repository. After the data is downloaded from the source server to the UEM console, it is then transferred to the relay server. The data is then downloaded from the relay server to devices.

Relay Server Basics

The relay server acts as an FTP / Explicit FTPS / SFTP / HTTPS server that distributes products to the device for download and installation. You can distribute to all devices without consuming all the bandwidth to the main/central MDM server. There are two ways the files are sent from the Workspace ONE UEM console, Push and Pull.

- Push Relay Servers.

  This method is typically used in on-premises deployments. The UEM console pushes content and applications contained in the product or staging to the relay server. SaaS customers interested in push relay servers must configure a public DNS to make the relay server available, and allows the Workspace ONE UEM server to open the required connection to send the files. Push servers must be FTP, Explicit FTPS, or SFTP file servers. HTTPS is supported only for Pull Relay Servers.

- Pull Relay Servers.

  Typically used in SaaS deployments, a pull service is installed on the relay server and monitors the UEM console for files to be downloaded. When it finds some, it downloads content and applications (whether provisioned or staged) to the FTP home directory through an outbound connection. This option is ideal because the pull service itself opens the HTTPS connection to the UEM console, making the need for a public connection unnecessary.

- Relay Service Cloud Connector.

A Relay Server Cloud Connector (RSCC) is a hybrid solution that pulls content from a service endpoint and distributes it to your relay servers. This design initiates an outbound connection from your network to the VMware cloud to download content for distribution. An outbound connection represents a security advantage over other designs. You can also configure RSCC as a Pull Relay Server. In such a case, RSCC is not enabled under Product Provisioning settings and the RSCC binary is configured as a Pull Relay.

FTP and FTPS servers must be compliant with RFC 959 and RFC 2228 set by the Internet Engineering Task Force.

Relay servers are required for barcode staging on Android and Windows Rugged. Otherwise, relay servers are optional but useful for pushing products to downloaded applications and content – as opposed to downloading directly from the server that hosts the Workspace ONE UEM console.

Relay servers also add redundancy through the fallback feature. If a device's relay server is down, the device falls back to the next relay server in the hierarchy system until it finds a working server or connects to the Workspace ONE UEM console server.

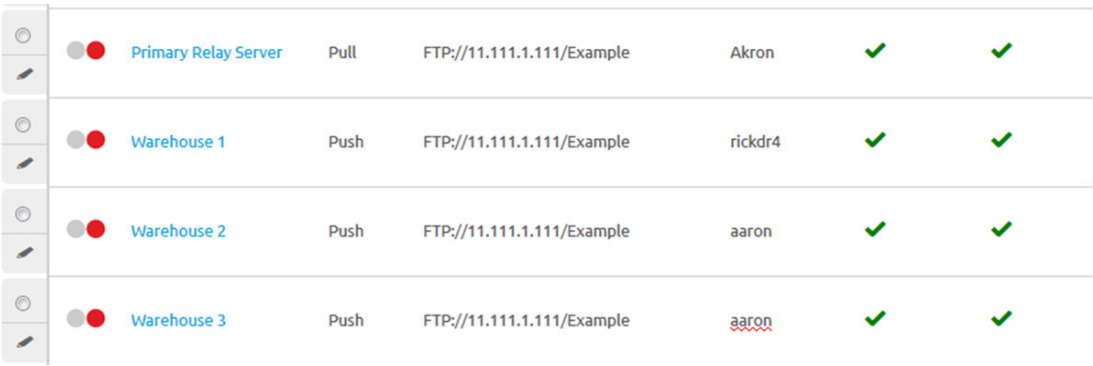
If you are not using a relay server, the device downloads applications and content directly from the UEM console server.

Relay Server and Job Status Updates

Relay servers greatly enhance the performance in a product provisioning environment, however, they complicate the task of reporting the status of a job. For more information, see Product Job Statuses.

Relay Server List View

Navigate to **Devices > Provisioning > Relay Servers > List View** to see all the push and pull relay servers in your environment. After creating a relay server, refresh the relay server list view to get the status of the connection.



The **Source Server** and **Relay Server** statuses are defined as follows:

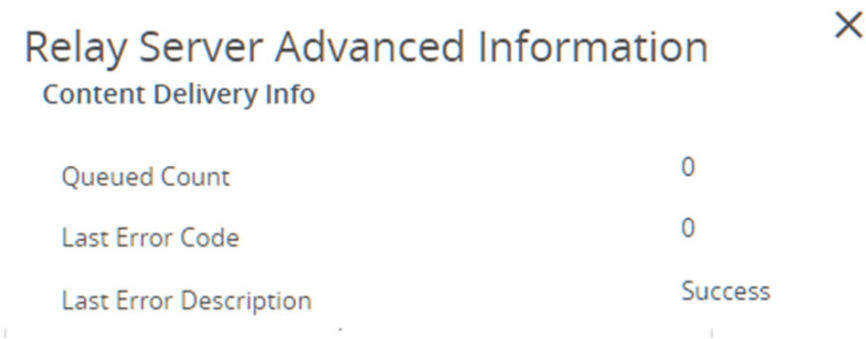
Indicator	Source Server	Relay Server
✔	Last retrieval from server succeeded.	Last file sync with server succeeded.
•••	Retrieval from a server is in progress.	File sync with a server is in progress.
⚠	Last retrieval failed.	Last file sync failed.

Once the check mark displays for both source server and relay server, the product components are available to distribute to the end-user device.

You can **Export** the Relay Servers List View to CSV (comma separated values) or XLSX file, both of which you can analyze with MS Excel. This file includes all the relevant details for each relay server in the listing.

Advanced Info

You can access the **Advanced Info** action for more detailed information pertaining to the server. This action can be found in the **More Actions** options drop-down available after selecting a relay server. The Advanced Info action displays the **Queued Count** of files, the **Last Error Code** displayed, and the **Last Error Description**.



Batch Import Relay Servers

The Batch Import feature of Workspace ONE UEM loads relay servers into your product provisioning solution in bulk, saving you time if you have many to add. The **Batch Import** screen serves two purposes, 1) download a blank relay server batch file template and 2) import a completed template.

Download a blank relay server batch file template, fill it out, then upload the completed template by taking the following steps.

1 Navigate to **Devices > Provisioning > Relay Servers > List View** then select **Add** and **Batch Import**.

The Batch Import screen displays.

2 Select the **Show Server Function Codes** and **Show Relay Server Types** links to see the codes that you need as you fill in the relay server batch import template.

3 Select the Download template link and save the template to your device.

4 Open the template with MS Excel or your favorite text editor.

   The template features two sample entries. These entries allow you to see what kinds of values and their formats the system expects to find in each column heading when you import your completed template.

5 You must associate the relay server users with an organization group (GroupID).

   The columns that feature an asterisk are required.

6 Remove the sample entries before you save your completed template.

7 Save the template in CSV format.

8 Return to the Workspace ONE UEM console and navigate once again to **Devices > Provisioning > Relay Servers > List View** then select **Add** and **Batch Import**.

9 Enter a **Batch Name**.

10 Enter a **Batch Description**.

11 Select **Choose File** to upload the completed **Batch File**.

   Batch files must be in CSV format.

12 Select **Import** to upload the file and begin the batch import process.

Results: All the relay servers in the correctly filled out template are now available to host product content in Workspace ONE UEM.

View Remote Files on Relay Server

Workspace ONE UEM lets you view files sent to a relay server through the Workspace ONE UEM console.

To use this feature, you must allow inbound communication from the Workspace ONE UEM console to the relay server over the FTP or HTTPS port used by the file server.

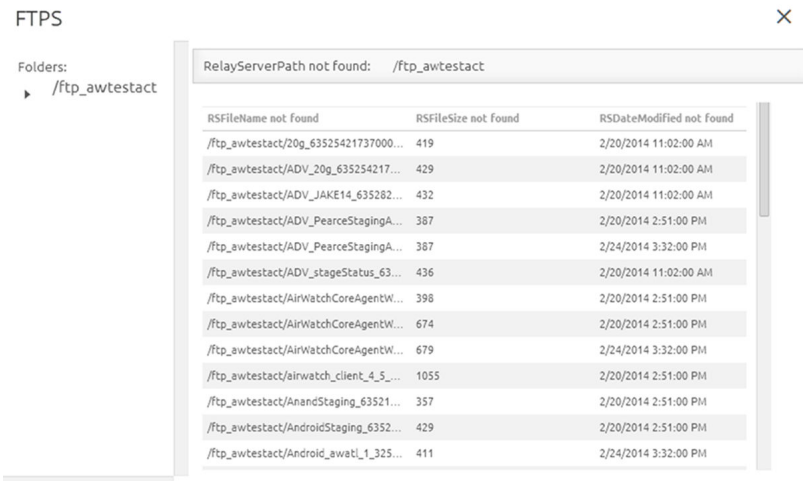
Since pull relay servers typically are used in networks where inbound communications cannot be allowed, the View Remote Files feature is typically used with push relay servers.

1 Navigate to **Devices > Provisioning > Relay Servers > List View**.

2 Select the server you are interested in viewing by clicking the radio button to the left of the Active indicator, above the Edit pencil icon.

3 Select the **More Actions** button.

4 Select **Remote File List** and open the Remote File List for your selected relay server.



This chapter includes the following topics:

- Configure a Relay Server
- Configure a Relay Server Cloud Connector
- Pull Service Based Relay Server Configuration

Configure a Relay Server

Configure a relay server for product provisioning by selecting an FTP, Explicit FTPS, Implicit FTPS (Pull only), SFTP file server, or HTTPS (pull only) protocol and integrating it with Workspace ONE UEM powered by AirWatch.

Important If you use the pull service to create a pull-based relay server, you must give the home directory full SYSTEM access. This configuration means the pull service stores and removes files from the directory.

Prerequisites

- You need an FTP, Explicit FTPS, Implicit FTPS in Pull configuration (Android only), SFTP, or HTTPS (Pull only) file server.
  - Implicit FTPS relay servers are only supported in a pull configuration and can only be used with Android devices.
  - Pull service bandwidth needs and minimum hardware requirements are negligible when compared to pushing products to devices. Such needs are entirely dependent upon 1) the number of products you are pushing, 2) how often they are pushed, and 3) the size of the products in MBs.
  - When assessing hardware and bandwidth needs for FTP servers, consider following general guidelines and adjust their specifications as your needs change.
  - General FTP Server Guidelines: 2 GHz x86 or x64 processor and 4 GB RAM.

- For FTP, FTPS, and SFTP servers, you must create an FTP user with a home directory. This user must have read/write/delete permissions for both the directory and the files used in the relay server. This FTP user must have a user name and password for authentication.
- Workspace ONE UEM supports SFTP servers for product provisioning, however, the supported staging clients, Stage Now (Android), and Rapid Deployment, do not support SFTP servers for use with barcode staging.
- If selecting an HTTPS protocol (pull configuration only), you must configure the HTTPS endpoint using the web server configuration tool of choice (for example, IIS). The root directory you opt in the web server config must be the same as the Pull Local Directory of the relay server.
- FTP and FTPS servers must be compliant with RFC 959 and RFC 2228 set by the Internet Engineering Task Force.

Data Security

Relay servers may hold sensitive data, so consider encrypting it.

- Data In Transit – FTPS, SFTP, and HTTPS relay servers use TLS/SSL or SSH protocols to secure data in transit between the relay server and Workspace ONE UEM as well as between the relay server and devices.
- Data In Storage – Consider using an OS-level disk encryption to protect your data in storage. Tools such as Bitlocker (Windows) and GnuPG (Linux) can be used to encrypt content stored on the relay servers.

Procedure

1 Navigate to **Devices > Provisioning > Relay Servers > List View** and select **Add**, followed by **Add Relay Server**.

2 Complete all applicable settings in the tabs that are displayed.

Table 2-1. General Tab

Setting	Description
Name	Enter a name for the relay server.
Description	Enter a description for the relay server.
Relay Server Type	Select either Push or Pull as the relay server method. **Push** – This method is typically used in on-premises deployments. The UEM console pushes content and applications contained in the product or staging to the relay server. **Pull** – This method is typically used in SaaS deployments. A web-based application stored in the relay server pulls content and applications contained in the product or staging from the UEM console through an outbound connection. For more information on installing a pull server, see Pull Service Based Relay Server Configuration.

Table 2-1. General Tab (continued)

Setting	Description
Log Level	This option is available only for **Pull** server types. Select the level of detail you want the log to capture as your relay server operates. **Error** to log only when things go wrong or **Debug** to capture all available detail.
Restrict Content Delivery Window	Limits content delivery to a specific time window. Provide a **Start Time** and **End Time** to restrict the delivery of content. The start time and end time of the restriction window is based on Coordinated Universal Time (UTC), which the system obtains by converting the console server time into Greenwich Mean Time (GMT). Set the system time on the console server accurately to ensure that your content is delivered on time.

Table 2-2. Assignment Tab

Setting	Description
Managed By	Select the organization group that manages the relay server.
Staging Server Assigned Organization Groups	Assign the organization groups that use the relay server as a staging server. A staging server only works for the staging process involving the supported staging clients, Stage Now (Android), and Rapid Deployment.
Production Server Assigned Organization Groups	Assign the organization groups that use the relay server as a production server. A production server works with any device with the proper Workspace ONE Intelligent Hub installed on it. **Android and Windows Rugged Only:** If you want to use the FTPS server for Barcode Enrollment only and not for Product Provisioning, remove all assigned organization groups under the Production Server section.

Table 2-3. Device Connection Tab

Setting	Description
Protocol	The information the device uses to authenticate with the FTP server when downloading applications and content. Select between **FTP**, **Explicit FTPS**, **Implicit FTPS in Pull configuration (Android only)**, **SFTP**, or **HTTPS (Pull only)** as the Protocol for the relay server. **Only Android** supports Implicit FTPS relay servers instead of Explicit FTPS relay servers and only in a pull configuration. If using FTPS or HTTPS, your server must have a valid SSL certificate. Configure the SSL certificate on the FTPS or HTTPS server. If selecting an HTTPS protocol, you must configure the HTTPS endpoint using the web server configuration tool of choice (for example, IIS).
Hostname	Enter the name of the server that hosts the device connection.
Port	Select the port established for your server. **Important** The ports you configure when you create your FTP, Explicit FTPS, Implicit FTPS in Pull configuration (Android only), SFTP, or HTTPS (Pull only) server must be the same ports you enter when creating a relay server in the Workspace ONE UEM console.

Table 2-3. Device Connection Tab (continued)

Setting	Description
User	Enter the server user name. Domain-based user names are supported, accepted formats for domain users are `username@domain` and `domain\username`. **Note** You can authenticate onto StageNow relay servers using a domain user name.
Password	Enter the server password. Passwords may not contain the colon : special character. **Note** **Regarding Zebra Stage Now Barcodes**, you are restricted from using these characters in the FTP/FTPS password. “@” | “/” | “\” | “:” | “;” | “,” | “?” | “$” | “&” | “=” | “+” | "!"
Path	Enter the path for the server. This path determines where Workspace ONE UEM content resides within the FTP/HTTP root directory. For example, if the path is set to `\WS1` and the FTP/HTTP root directory is set to `c:\ftproot`, then all content resides under `c:\ftproot\WS1`.
Passive Mode	Enable to ensure that the connection is trusted and there are no SSL errors.
Verify Server	This setting is only visible when **Protocol** is set to FTPS. Enable ensures that the connection is trusted and there are no SSL errors. If left deselected, then the certificate used to encrypt the data can be untrusted and data can still be sent.

3 For **Push** server selections made in the **General** tab, select the **Console Connection** tab and finish the settings. For **Pull** server selections, go to step 4.

The Console Connection tab contains information that the Workspace ONE UEM console uses to authenticate with the FTP(S)/SFTP server when pushing applications and content. The settings are typically identical to the **Device Connection** tab. Select the **Copy Values From Device Connection** button to save yourself from having to enter values from the Device Connection tab manually.

a Press the **Test Connection** button to test your Console Connection to the push server.

Each step of the connection is tested and the results are displayed to help with troubleshooting connection issues.

b Press the **Export** button on the **Test Relay Server Connection** page to export the data from the test as an XLSX or CSV (comma-separated values) file.

c Go directly to step 5.

4 For **Pull** server selections made in the **General** tab, select the **Pull Connection** tab and complete the settings.

Settings	Descriptions
Pull Local Directory.	Enter the local directory path for the server. The directory you enter here must be the same as the root directory you have chosen for the FTP or HTTP file server. For example, if you have configured an HTTPS endpoint and selected `c:\rootfolder` as your root directory in IIS, then you must use `c:\rootfolder` for your **Pull Local Directory**.
Pull Discovery Text.	Enter the local (not public) IP address or the MAC address of the server. IP addresses use periods as normal but MAC addresses do not use any punctuation in this form.
Pull Frequency.	Enter the frequency in minutes that the pull server should review the UEM console for changes in the product.

5 Select **Save**.

Configure a Relay Server Cloud Connector

Configure a Relay Server Cloud Connector for product provisioning by selecting an FTP, Explicit FTPS, Implicit FTPS (Pull only), SFTP file server, or HTTPS (pull only) protocol and integrating it with Workspace ONE UEM powered by AirWatch.

Important If you use the pull service to create a pull-based relay server, you must give the home directory SYSTEM full access. This configuration means the pull service stores and removes files from the directory.

Client-server applications such as Workspace ONE UEM use the transport layer security (TLS) cryptographic protocol to communicate across a network. Three transfer protocols support TLS. The file transfer protocol (FTP), the file transfer protocol over SSL (FTPS), and the SSH file transfer protocol (SFTP).

These file transfer protocols only secure those parts of the process where data is in transit between the client and the server. Because of this limitation, use an OS-level disk encryption. There are several operating system-specific tools available (for example BitLocker for Windows, GnuPG for Linux).

As an alternative to traditional push or pull relay service, Workspace ONE UEM supports the creation of a Relay Server Cloud Connection (RSCC). An RSCC is a hybrid solution that pulls content (products only) from a content service endpoint and distributes that content (products only) to your relay servers. This design can bring performance improvements over a traditional pull relay server. This Relay Server workflow includes RSCC options.

You can also configure RSCC as a Pull Relay Server. In such a case, RSCC is not enabled under Product Provisioning settings and the RSCC binary is configured as a Pull Relay

Prerequisites

- You need an FTP, Explicit FTPS, Implicit FTPS in Pull configuration (Android only), SFTP, or HTTPS (Pull only) file server.
  - Implicit FTPS relay servers are only supported in a pull configuration and can only be used with Android devices.
  - Pull service bandwidth needs and minimum hardware requirements are negligible when compared to pushing products to devices. Such needs are entirely dependent upon 1) the number of products you are pushing, 2) how often they are pushed, and 3) the size of the products in MBs.
  - When assessing hardware and bandwidth needs for FTP servers, consider following general guidelines and adjust their specifications as your needs change.
  - General FTP Server Guidelines: 2 GHz x86 or x64 processor and 4 GB RAM.
- You must create an FTP user with a home directory. This user must have read/write/delete permissions for both the directory and the files used in the relay server. This FTP user must have a user name and password for authentication.
- Workspace ONE UEM supports SFTP servers, however, the supported staging clients, Stage Now (Android), and Rapid Deployment, do not support SFTP servers for use with barcode staging.
- If selecting an HTTPS protocol (pull configuration only), you must configure the HTTPS endpoint using the web server configuration tool of choice (for example, IIS). The root directory you opt in the web server config must be the same as the Pull Local Directory of the relay server.
- FTP and FTPS servers must be compliant with RFC 959 and RFC 2228 set by the Internet Engineering Task Force.

Procedure

1 Navigate to **Groups & Settings > All Settings > Admin > Product Provisioning** and set the **Relay Server Cloud Connector** option to **Enabled**.

2 Navigate to **Devices > Provisioning > Relay Servers > List View** and select the **Add** button, followed by **Add Relay Server**.

3 Complete all applicable settings in the tabs that are displayed.

Table 2-4. General Tab

Setting	Description
Name	Enter a name for the relay server.
Description	Enter a description for the relay server.

Table 2-4. General Tab (continued)

Setting	Description
Relay Server Type	Select either Push or Cloud Relay as the relay server method. **Push** – This method is typically used in on-premises deployments. The console pushes content and applications contained in the product or staging to the relay server. **Cloud Relay** – Designed for SaaS deployments, the **Relay Server Cloud Connector (RSCC)** pulls content (products only) from a content service endpoint and distributes that content (products only) to your relay servers.
Log Level	This option is available only for **Cloud Relay** server types. Select the level of detail you want the log to capture as your relay server operates. **Error** to log only when things go wrong or **Debug** to capture all available detail.
Restrict Content Delivery Window	Enable limits the content delivery to a specific time window. Provide a **Start Time** and **End Time** to restrict the delivery of content. The start time and end time of the restriction window is based on Coordinated Universal Time (UTC), which the system obtains by converting the console server time into Greenwich Mean Time (GMT). Set the system time on the console server accurately to ensure that your content is delivered on time.

Table 2-5. Assignment Tab

Setting	Description
Managed By	Select the organization group that manages the relay server. **Android and Windows Rugged Only:** If you want to use the FTPS server for Barcode Enrollment only and not for Product Provisioning, remove all assigned organization groups under the Production Server section.
Staging Server Assigned Organization Groups	This option is available only for **Push** server types. Assign the organization groups that use the relay server as a staging server. A staging server only works for the staging process involving the supported staging clients, Stage Now (Android), and Rapid Deployment.
Production Server Assigned Organization Groups	This option is available only for **Push** server types. Assign the organization groups that use the relay server as a production server. A production server works with any device with the proper Workspace ONE Intelligent Hub installed on it.

4

- If you selected **Push** as your **Relay Server Type** in the **General** tab, then complete the **Device Connection** tab that follows.
- Otherwise, for **Relay Server** selections, skip this step and proceed directly to step 5.

Table 2-6. Device Connection Tab (only available for Push Relay Server Selections)

Setting	Description
Protocol	The information the device uses to authenticate with the FTP server when downloading applications and content. Select between **FTP**, **Explicit FTPS**, **Implicit FTPS in Pull configuration (Android only)**, **SFTP**, or **HTTPS (Pull only)** as the Protocol for the relay server. **Only Android** supports Implicit FTPS relay servers instead of Explicit FTPS relay servers and only in a pull configuration. If using Explicit FTPS, your Explicit FTPS server must have a valid SSL certificate. Configure the SSL certificate on the Explicit FTPS server. If selecting an HTTPS protocol, you must configure the HTTPS endpoint using the web server configuration tool of choice (for example, IIS).
Hostname	Enter the name of the server that hosts the device connection.
Port	Select the port established for your server. **Important** The ports you configure when you create your FTP, Explicit FTPS, Implicit FTPS in Pull configuration (Android only), SFTP, or HTTPS (Pull only) server must be the same ports you enter when creating a relay server in the Workspace ONE UEM console.
User	Enter the server user name. **Note** You can authenticate onto StageNow relay servers using a domain user name.
Password	Enter the server password. Passwords may not contain the colon : special character. **Note** **Regarding Zebra Stage Now Barcodes**, you are restricted from using these characters in the FTP/FTPS password. “@” | “/” | “\” | “:” | “;” | “,” | “?” | “$” | “&” | “=” | “+” | "!"
Path	Enter the path for the server. This path must match the home directory path of the ftp user. For example, if the ftp user's home directory is c:\ftp\home\jdoe, the path entered into this text box must be c:\ftp\home\jdoe.
Passive Mode	Passive Mode forces the server to select the data port on behalf of the device. Select Enabled for this option. Conversely, Active Mode directs both the server and the device to use pre-defined ports for transfer. Select Disabled for this option.
Verify Server	This setting is only visible when **Protocol** is set to FTPS. Enable to ensure that the connection is trusted and there are no SSL errors. If left deselected, then the certificate used to encrypt the data can be untrusted and data can still be sent.

a Next, select the **Cloud Relay Connection Tab** and complete the settings.

The **Cloud Relay Connection** tab contains information that the console uses to authenticate with the FTP(S) server when pushing applications and content. The settings are typically identical to the **Device Connection** tab. Select the **Copy Values From Device Connection** button to save yourself from having to enter values from the Device Connection tab manually.

Go directly to step 6.

5 If you selected **Cloud Relay** as your **Relay Server Type** in the **General** tab, then select the **Pull Connection Tab** and complete the settings.

Setting	Description
Pull Local Directory.	Enter the local directory path for the server. The directory you enter here must be the same as the root directory you opt when configuring an HTTPS endpoint on the webserver. For example, if you have configured an HTTPS endpoint and selected c:\rootfolder as your root directory in IIS, then you must use c:\rootfolder for your **Pull Local Directory**.
Pull Discovery Text.	Enter the IP addresses or the MAC addresses of the server. Separate each address with commas. IP addresses use periods as normal but MAC addresses do not use any punctuation in this form.
Pull Frequency.	Enter the frequency in minutes that the pull server monitors with the UEM console for changes in the product.
Max Push Connections	Modify this value to throttle the maximum number of simultaneous connections used to push content to Relay Servers. The default value is 50.
Report Status Batch Size	As content is distributed to targets Relay Servers, Cloud Relay reports the transfer status of each Relay Server to the Workspace ONE Cloud. Modify this value to throttle the number of Relay Server statuses to include in each batch. The default value is 100 statuses per batch.

6 Select **Save**.

Relay Server Cloud Connector (RSCC) Hardware Requirements and Installation

A Relay Server Cloud Connector (RSCC) is a hybrid solution that pulls content from a service endpoint and distributes it to your relay servers. This design initiates an outbound connection from your network to the VMware cloud to download content for distribution. An outbound connection represents a security advantage over other designs.

Prerequisites

Each server intended to be used as an RSCC conduit must adhere to the following hardware specifications. The quantity of servers you must configure depends upon the number of products you want to push at a time.

RSCC Server Hardware: CPU with 4 cores x 2 sockets (for a total of 8 vCPUs), 8GB RAM and 100 GB of free HD/SSD space (not including space required for the OS). The extra space is for the storage of files on RSCC.

Procedure

1 Enable the PullServiceInstaller feature flag

   a `https//{server url}/api/system/featureflag/PullServiceInstallerFeatureFlag/{global-locationgroup-guid}/true`

   b for example, `https//localhost/api/system/featureflag/PullServiceInstallerFeatureFlag/B36DE603-E05E-4E17-8DA5-B9760975BCBD/true`

2 Download the RSCC Installer

3 Generate an Installer Certificate. Navigate to **Groups & Settings > All Settings > System > Enterprise Integration > Pull Service Installers**. Select the **Generate** button.

4 Download the RSCC configuration file. If you are configuring RSCC for the first time, select the **Download Configuration** button.

5 Place the installer and the configuration file in the same folder of the RSCC server.

6 Run the RSCC Installer on each server you want to create.

   a Run the installer and choose the installation path.

   b Provide the password that was entered while downloading the certificate in step 3.

   c Enable or disable proxy.

      - If disabled, proceed to the next step.
      - If enabled, then supply the Proxy Information including the Proxy Host name, port number, auth type with username and password.

   d Enable or disable Custom Discovery Text.

      - If disabled, the Discovery Text becomes a combination of the server's IP address plus its MAC address.
      - If enabled, the Custom Discovery Text entered must be unique for each RSCC server.

   e Select **Install** and complete the installation.

7 System Code Changes (where are these located?)

   a RelayServerCloudConnectorApplicableCommandRetrievalBatchSize: 2000

   b RelayServerCloudConnectorMaxCommandQueueIteration: 1

   c RelayServerCloudConnectorBatchSize: 50

8 If you have more than one content pull endpoint, take the following steps to directly update the RSCC config file with the content pull endpoint URL.

   a Login to each RSCC server machine

   b Stop the RSCC service

   c Navigate to the folder where RSCC is installed and locate the `appsettings.json` file.

d Edit the **PullServiceURL** field with the content pull URL. For example, https://vmware.bng.com/contentpull

e Navigate to c:\windows\system32\drivers\etc and edit the `hosts` file.

f Add the IP address and host name to the `hosts` file as shown.

```
10.1.1.1 vmware.bng.com
```

g Save the hosts file.

h Restart the RSCC service.

9 Pull up the list of Windows Services and stop the Content Delivery Service (CDS).

10 Enable the RSCC feature flag **(this step to be removed once the FF is on by default).**

a `https//{server url}/api/system/featureflag/RelayServerCloudConnectorFeatureFlag/{global-locationgroup-guid}/true`

b for example, `https//localhost/api/system/featureflag/RelayServerCloudConnectorFeatureFlag/B36DE603-E05E-4E17-8DA5-B9760975BCBD/true`

11 Add each RSCC server in the console one by one using this configuration.

a While logged into Workspace ONE UEM in the Global OG, navigate to **Devices > Provisioning > Relay Servers > Undiscovered Pull Relay Servers**.

b Copy (Ctrl-C) the pull discovery text. If the pull discovery text is not provided, then you must use the IP address from one of the RSCC machines (except 127.0.0.1) to create the relay server.

c Move to the OG in which you downloaded the RSCC certificate and navigate to **Devices > Provisioning > Relay Servers > List View** and select **Add > Add Relay Server** button.

d Complete the required text boxes, making sure **Relay Server Type** is set to 'Cloud Relay'.

e Switch to the **Assignment** tab and select the appropriate OG.

f Switch to the **Pull Connection** tab and complete all required text boxes, making sure you paste the discovery text (copied earlier) into the **Pull Discovery Text** text box.

g Select Save. Repeat a. through g. until each RSCC server is added.

12 On the **Relay Servers > List View** page, activate each RSCC server by selecting its red toggle button, turning it to green.

Pull Service Based Relay Server Configuration

Pull service-based relay servers periodically contact the Workspace ONE UEM to monitor for new products, profiles, files, actions, and applications provisioned to devices under the pull

relay servers purview. Configure a pull server to provision content to devices without excessive bandwidth use.

The server creates an outbound https connection on port 443 to the UEM console and periodically polls for changes or additions. If the server finds changes or additions, then it downloads the new content onto the server before pushing it to its devices.

Pull service is preferred when using a NAT firewall or SaaS environments over on-premises hybrid environments. The reason is that SaaS customers typically do not want the service to tie up bandwidth when content is delivered from Workspace ONE UEM to the store server.

Note The IP configured in the pull connection / pull discovery must be an internal IP address for the server. The service does not configure correctly if an external IP or NAT IP address is used.

Configure .NET Core Pull Relay Server, PS v2.0

You can configure a pull service v2.0 relay server based on Microsoft's .NET Core, the open source, cross-platform successor to their .NET Framework.

The .NET Core based pull service is built using .Net core 2.1. For more information, see https://dotnet.microsoft.com/.

Hardware Requirements

Component	Requirement
Server	1
CPUs	2 (2.0 GHz Intel Processor, x86 64bit)
Memory	4 GB
Storage*	25 GB

* Required storage size for a relay server must scale with the collective size of the files and applications that will be distributed through it. A file is not deleted from the relay server until the associated File Action component is deleted from the UEM Console. Likewise, memory and CPU requirements must increase as the count of devices using the server grows.

On the client side, the pull service can use any dynamically assigned port, with no limitations, to connect to content pull on the server side.

Currently, .NET Core does not support HTTPS proxies.

Network Requirements

Component	Requirement
Outbound Traffic to Workspace ONE UEM console	Port 443
Protocol for Outbound Traffic to Workspace ONE UEM console	HTTPS

Software Requirements

Component	Requirement
Operating System	Windows 64 bit or RHEL7 64 bit or CentOS7 64 bit** or Ubuntu 18.04 64 bit**
Workspace ONE UEM	Version 1903 or later. Customers using versions 1903 through 1907 must enable the `PullServiceInstallerFeatureFlag`. Contact VMware Support for assistance in updating the database with this feature flag. The feature is enabled by default in versions 1908 and later.

Install Pull Relay Service v2.0

1 Visit https://my.workspaceone.com/products/Pull-Service and download the pull service installer for Windows or Linux platform.

- The Linux pull service installer has a hard-coded installation path of "/OPT/". You must configure this directory with sufficient write permissions and the top-most privileges. You must also have sufficient write permissions for the directory where the pull service is configured to download the contents.
- **Linux Silent Installation Option:** You have the option of running the Linux installation without being prompted. The following is an example of the code you might use to install silently the pull relay server for Linux with an NTLM proxy, basic authentication, and custom discovery text.

```
sudo ./VmwarePullserviceinstaller.bin -i silent -DPULL_SERVICE_CERT_PASSWORD=Test@1234
-DOUTBOUND_PROXY=1 -DOUTBOUND_PROXY_HOST=192.168.1.100 -DOUTBOUND_PROXY_PORT=808
-DOUTBOUND_PROXY_CREDENTIALS=1 -DOUTBOUND_PROXY_NTLM_AUTH=0 -DOUTBOUND_PROXY_DOMAIN=
-DOUTBOUND_PROXY_USERNAME=john -DOUTBOUND_PROXY_PASSWORD=hello@1234
-DIS_DISCOVERY_TEXT=1 -DDISCOVERY_TEXT=og1
```

- **Windows Silent Install Option (without custom discovery text):**

```
VMware_WS1_PullService_2.0_for_Windows.exe /s /v" /qn /l*v pullservice.log
AWUSERPROVIDEDPASS=<password>"
```

- **Windows Silent Install Option (with custom discovery text):**

```
VMware_WS1_PullService_2.0_for_Windows.exe /s /v" /qn /l*v pullservice.log
AWUSERPROVIDEDPASS=<password> AW_IS_CUSTOM_DISCOVERY=1 AWDISCOVERY_TEXT=<Disc_Text>"
```

2 On-premises customers using older versions of the console can enable the PullServiceInstallerFeatureFlag. Contact VMware Support for assistance in updating the database with this feature flag.

3 While logged into Workspace ONE UEM, you must select an organization group (OG) to install the relay service onto. It is a best practice that the devices you intend to pull content from the relay service are managed under the same OG (or lower) as that which the pull service is installed.

4 Once you have selected an OG that is equal to or above the OG that manages the devices that you want to use the pull service, navigate to **Groups & Settings > All Settings > System > Enterprise Integration > Pull Service Installers** and select the **Download Configuration** button. The **Download pull service configuration** screen displays.

5 Enter and confirm the **Certificate password** you want to use for the pull service certificate and select **Download** to save the ZIP file. Save the ZIP file to the same directory as the pull service installer you downloaded in step 1.

6 Extract the contents of the ZIP file to the directory in which you saved it.

Result: The ZIP file contains the pull service installer CONFIG file which is a file you require to run the installer. You should have the Windows EXE (or Linux BIN) Pull Service Installer file plus the CONFIG file extracted from the ZIP file together in the same directory. If you are installing the Linux variant, you might also have one of the NTLM enablers from step 2.

7 Run the Pull Service Installer, supplying the password you entered in step 5 when prompted. Continue with the remainder of the installation.

a Select whether you wish to use an outbound proxy and, if applicable, enter the proxy configuration.

b Select whether you wish to use Custom Discovery Text. The Pull Service uses Discovery Text to identify itself when communicating with Workspace ONE UEM servers. By default, Pull Service v1.0 and v2.0 use the local IP Address and MAC Address of the server but a custom value can be used instead. This is necessary to identify each Pull Server separately in rare cases where multiple Pull Servers do not have unique and distinct local IP Address and MAC Address values.

When adding Custom Discovery Text, please note that the value entered in the installer is not the full Discovery Text value. The installer will append it to a unique GUID and store it in the `appsettings.json` file. This file can be found in the install directory of the Pull Service under the `/bin` folder.

The Relay Server configuration in the Workspace ONE UEM console must use this Discovery Text value. For more information, see Configure a Relay Server.

8 ** If you are installing the Linux version and you want to configure an NTLM proxy authentication, the "gss-ntlmssp" package must be installed. Select the download link that corresponds to your Linux flavor and install it now.

- For CentOS, download the following RPM file, https://centos.pkgs.org/7/epel-x86_64/gssntlmssp-0.7.0-1.el7.x86_64.rpm.html

Using the .NET Core Pull Service, Best Practices

- Do not run multiple instances of pull service (old or new versions) on the same machine.
- To ensure that devices can access the pull service without issue, install the pull service and its CONFIG file to an OG equal to or above the OG that manages those devices.
- Once the pull service is installed, avoid selecting the **Regenerate** button on the Pull Service Installers settings page. Regenerating the certificate after the pull service is running breaks communication between the Workspace ONE UEM console and the pull service client. If you must regenerate the certificate, then you must rerun the pull service installer.
- On-premises customers must generate and download the PSinstaller.CONFIG file directly from Global OG and not from Customer OG. This configuration enables an easy relay server move from OG 1 to OG 2 without having to reinstall the pull service every time. For more information, see Move an Existing Pull Relay Server from One Organization Group to Another.

Using the .NET Core Pull Service, Starting and Stopping the Pull Service

- Start/stop service on **CentOS and RHEL**.

```
/etc/init.d/awpullservice {start|stop|status|restart)
```

  - Start the pull service.

```
service awpullservice start
```

  - Stop the pull service.

```
service awpullservice stop
```

  - Displays status of the pull service.

```
service awpullservice status
```

  - Restarts the pull service.

```
service awpullservice restart
```

- Start/stop service on **Ubuntu**. Run this SH file, which is located in the pull-service/bin folder.
  - Start the pull service.

    ```
    sh start.sh
    ```

  - Stop the pull service.

    ```
    sh stop.sh
    ```

- Start/stop service on **Windows**.
  - Navigate to **Run > services.msc**, locate the **Airwatch Pull Service**, and select start, stop, or restart.

Using the .NET Core Pull Service, Appsettings.json Tweaks

The appsettings.json file settings presented here are the default values which you can change to modify the server's default behavior. The appsettings.json file is located in the pull service\Bin folder.

If the "discoveryText" value is set as null in the appsettings.json, then the pull service assembles and posts a default discovery text. This discovery text is composed of the merging of the pull server's IP address and Mac address in place of the null value.

The pull service must be manually restarted each time you change values in appsettings.json to apply the new settings.

- "performSystemTaskIntervalMin": "720" (in minutes), the interval when the pull service logs are sent to the content pull endpoint. For more information, see Enable Pull Service Logging.
- "clientPostTimeOut": "240" (in minutes), the amount of time the connection remains active before timing out. This time frame can increase as the size of your files increase to avoid connection timeouts.
- "waitTimeOnFailure": "300" (in seconds), the amount of time you allow the pull service to rediscover the relay server if there is a connection error.
- "logMaxEntries": "10000", the maximum number of log entries in a single log file.
- "maxLogFiles": "5", the maximum number of logs generated.
- "maxLogFileSize": "2000000" (in bytes), the maximum size a log file is allowed to get.
- "enableConsoleLog": "false", for developmental purposes. Enable to activate the console log.
- "proxyBypassOnLocal": "false", enable/disable the bypass proxy server for local addresses.

Mirgating to the .NET Core Pull Service

The .NET Core pull service installer does not support an automatic migration with custom discovery text. When you reinstall the same version of the PS or upgrade the PS to a new version, the custom discovery text value is different each time. For this reason, make a note of the complete discovery text value under appsettings.json ("discoveryText") or from the Undiscovered Pull Relay servers page on the console after a fresh installation for the first time (for example: a6af8505-3d0c-4490-ab02-14db2fa3a80c_PSmachine1).

Saving the discovery text value is useful for when you accidentally uninstall the pull service or when you must move the relay server from one OG to another or from one server to another.

Follow this task when you do not want to use custom discovery text after migration and you want to continue using the same default discovery text. The default discovery text is the pull server's IP address plus Mac address.

1 Install the pull service v2.0 as described in the above section, **Install Pull Relay Service v2.0**, setting Custom Discovery Text, per your preference.

> **Note** If you want to apply Custom Discovery Text during the pull service v2.0 installation, the Relay Server configuration in the Workspace ONE UEM console must be updated after the installation.

2 From the pull server, navigate to the `/bin` folder of the installation directory, locate the `appsettings.json` file, and copy the full Discovery text (GUID_CustomText) with the Ctrl-C shortcut.

3 From the UEM console, navigate to **Devices > Provisioning > Relay Servers > List View** and select the new pull relay server.

4 Select the **Console Connection** tab and paste (Ctrl-V) the Discovery Text from the old server onto the **Discovery Text** text box of this new server.

5 **Save** and Reactivate the server.

6 Restart the AirWatch Pull Service.

Recovering a Pull Relay Server v2.0 with Custom Discovery Text

Failure to recover the uninstalled or corrupted pull relay server may cause Workspace ONE UEM console to consider it a brand new server, causing the Pull Service to re-download all files that are already on the server. These steps apply for both Linux and Windows.

1 Take note of the current Custom Discovery Text of the corrupted/uninstalled Pull Relay Server. There are two ways to find the current Custom Discovery Text. You must choose…

- …from the **UEM console**, navigate to **Devices > Provisioning > Relay Servers > List View**, select the corrupted/uninstalled pull relay server, then open the **Console Connection** tab where the Custom Discovery Text is displayed. Go directly to Step 2.

- ...from the **pull server**, navigate to the install directory of the Pull Service v2.0 and open the `appsettings.json` file located in the `/bin` folder. The Custom Discovery Text can be found listed in the JSON file under "discoveryText".

2 Navigate to **Groups & Settings > All Settings > Enterprise Integration > Pull Service** and download the latest Pull Service installer.

3 Uninstall the existing Pull Service from the Linux machine.

4 Run the Pull Service installer you downloaded in step 2. During the installation process, you are presented with the option to enter your own discovery text. Enter the custom discovery text your found in step 1.

   - If using a Linux silent install, make sure you use the `-DDISCOVERY_TEXT` switch.

5 Restart the pull service.

The server resumes functioning as it did before, using the exact same discovery text. The already queued items and new items are served to the same directory path as before.

Pull Relay Server Security

Relay Servers may hold sensitive data. Pull servers use HTTPS, which encrypts data in transit. Consider encrypting it in storage as well by using tools like Bitlocker (Windows) and GnuPG (Linux) to enable OS-level encryption on the servers.

To create a pull relay server, you must first have an FTP, Explicit FTPS, Implicit FTPS in Pull configuration (Android only), SFTP, or HTTPS (Pull only) server to function as the relay server. FTP and FTPS servers must be compliant with RFC 959 and RFC 2228 set by the Internet Engineering Task Force.

Important The ports you configure when you create your FTP, Explicit FTPS, Implicit FTPS in Pull configuration (Android only), SFTP, or HTTPS (Pull only) server must be the same ports you enter when creating a relay server in the Workspace ONE UEM console.

The process covers the installation of one server at a time. For a bulk installation, you must use a third-party application. Workspace ONE UEM supports importing servers in bulk through the Batch Import option. See Batch Import Relay Servers for more information.

Move an Existing Pull Relay Server from One Organization Group to Another

You can move an existing pull relay server installed with the Windows or new Linux installer by taking the following steps.

If you have a .NET Core pull service in an on-premises environment, you must generate and download the PullServiceInstaller.CONFIG file directly from Global OG (not from a Customer OG). This configuration allows you to move the relay server from OG 1 to OG 2 easily without reinstalling the pull service.

To ensure that devices can access the pull service without issue, select the "target" OG2 equal to or above the OG that manages those devices.

If you have a regular .NET Framework pull service or Java pull service installed, then use the following steps to move your relay server from on OG to another.

1 Delete the existing pull relay server from the original OG on the console. Once it is deleted, the pull discovery text that belongs to the pull service starts appearing on the undiscovered pull discovery page at Global OG.

2 Navigate to **Devices > Provisioning > Relay Servers > Undiscovered Pull Relay Servers** and locate the server by searching for your IP address (only available for on-premises).

3 Copy the pull discovery text that includes the IP address of your selected server.

4 Create a relay server in the new OG and activate it. After activating the relay server in the new OG, the pull service discovery text listed in the Undiscovered Pull Relay Servers page disappears.

Move an Existing Pull Service from One Relay Server to Another Relay Server

You can move the pull service from one relay server to another relay server by taking the following steps.

Such an action is advisable for when you must install the pull service onto a server with upgraded hardware and your pull service runs with custom discovery text. The old relay server was configured with this custom discovery text and as a consequence, the new relay server must also be configured with the same custom discovery text.

1 On the old relay server, locate the appsettings.json file and find the custom "discoveryText" value including the GUID. Select and copy this string to the clipboard with Ctrl-C.

2 Uninstall the pull service from the old relay server manually.

3 Using the latest version of the PS installer, install the pull service onto the new relay server.

4 From the Workspace ONE UEM console, update the FTP server details of the new relay server, located in the **Device Connection** tab.

5 In the **Pull Connection** tab, paste the custom discovery text you copied in step 1 in the **Pull Discovery Text** text box with Ctrl-V. The custom discovery text is now copied from the old machine to the new machine.

6 Save and activate the relay server from the UEM console.

7 Restart the pull service.

Results: The new pull service, using the original custom discovery text, is configured on the new relay server. The new machine serves any existing pending or new content and the device selects content from the new relay server through FTP.

Configuring Session Persistence of a Load Balancer for Multiple Pull Relay Servers

You can configure a load balancer for multiple pull relay servers (both legacy and DotNetCore) by setting session persistence.

Relay servers process requests for content. When the number of requests exceed what a single server can handle, the solution is multiple relay servers. A load balancer can be used to distribute the load across multiple pull relay servers, for both legacy and DotNetCore pull services.

You must configure the load balancer with session persistence, ensuring that the same relay server receives each endpoint call per content request. If session persistence is not configured on the load balancer, subsequent requests that are routed to another relay server can result in a File-Not-Found error.

The following endpoints are called from the pull service to retrieve content.

- **GetNextManifest**, retrieves the file metadata which the client must download.
- **GetFileInfoEx**, retrieves the file-hash which is used to determine whether the file is already downloaded.
- **GetFileData**, downloads the actual content.

Resolution: Since the file is downloaded locally to the server where Content-pull is running, you must configure the Load Balancer to have session persistence (IP based) to route all three endpoints to the same relay server.

Enable Pull Service Logging

You can enable the collection of pull service logs by taking the following steps.

Adjust how often the pull service logs are sent to the content pull endpoint by editing the number of minutes configured for the "performSystemTaskIntervalMin" setting in the appsettings.json file. This file is located in the pull service machine's \Bin folder.

1 Navigate to **Devices > Provisioning > Relay Servers > List View**.

2 Select the radio button above the **Edit** icon on the relevant Pull Server.

3 Select the **More Actions** button and select **Pull Logging**.

4 Enable the **Collect Log File** check box, enter the **Start Date** and **End Date**, then select **Save**.

Results: Pull service logging is enabled for the time period entered.

Completed logs can be found on the Workspace ONE UEM console server under `x:\Airwatch\Logs\ContentPull\RSid\` replacing 'x' for the actual drive letter of your environment. SaaS environments must contact Support for further assistance.

Configure .NET Core Pull Relay Server, PS v2.0

You can configure a pull service v2.0 relay server based on Microsoft's .NET Core, the open source, cross-platform successor to their .NET Framework.

The .NET Core based pull service is built using .Net core 2.1. For more information, see https://dotnet.microsoft.com/.

Hardware Requirements

Component	Requirement
Server	1
CPUs	2 (2.0 GHz Intel Processor, x86 64bit)
Memory	4 GB
Storage*	25 GB

* Required storage size for a relay server must scale with the collective size of the files and applications that will be distributed through it. A file is not deleted from the relay server until the associated File Action component is deleted from the UEM Console. Likewise, memory and CPU requirements must increase as the count of devices using the server grows.

On the client side, the pull service can use any dynamically assigned port, with no limitations, to connect to content pull on the server side.

Currently, .NET Core does not support HTTPS proxies.

Network Requirements

Component	Requirement
Outbound Traffic to Workspace ONE UEM console	Port 443
Protocol for Outbound Traffic to Workspace ONE UEM console	HTTPS

Software Requirements

Component	Requirement
Operating System	Windows 64 bit or RHEL7 64 bit or CentOS7 64 bit** or Ubuntu 18.04 64 bit**
Workspace ONE UEM	Version 1903 or later. Customers using versions 1903 through 1907 must enable the `PullServiceInstallerFeatureFlag`. Contact VMware Support for assistance in updating the database with this feature flag. The feature is enabled by default in versions 1908 and later.

Install Pull Relay Service v2.0

1 Visit https://my.workspaceone.com/products/Pull-Service and download the .NET Core Pull Relay Server for Windows or Linux installer.

- The Linux pull service installer has a hard-coded installation path of "/OPT/". You must configure this directory with sufficient write permissions and the top-most privileges. You must also have sufficient write permissions for the directory where the pull service is configured to download the contents.
- **Linux Silent Installation Option:** You have the option of running the Linux installation without being prompted. The following is an example of the code you might use to install silently the pull relay server for Linux with an NTLM proxy, basic authentication, and custom discovery text.

```
sudo ./VmwarePullserviceinstaller.bin -i silent -DPULL_SERVICE_CERT_PASSWORD=Test@1234
-DOUTBOUND_PROXY=1 -DOUTBOUND_PROXY_HOST=192.168.1.100 -DOUTBOUND_PROXY_PORT=808
-DOUTBOUND_PROXY_CREDENTIALS=1 -DOUTBOUND_PROXY_NTLM_AUTH=0 -DOUTBOUND_PROXY_DOMAIN=
-DOUTBOUND_PROXY_USERNAME=john -DOUTBOUND_PROXY_PASSWORD=hello@1234
-DIS_DISCOVERY_TEXT=1 -DDISCOVERY_TEXT=og1
```

- **Windows Silent Install Option (without custom discovery text):**

```
VMware_WS1_PullService_2.0_for_Windows.exe /s /v" /qn /l*v pullservice.log
AWUSERPROVIDEDPASS=<password>"
```

- **Windows Silent Install Option (with custom discovery text):**

```
VMware_WS1_PullService_2.0_for_Windows.exe /s /v" /qn /l*v pullservice.log
AWUSERPROVIDEDPASS=<password> AW_IS_CUSTOM_DISCOVERY=1 AWDISCOVERY_TEXT=<Disc_Text>"
```

2 On-premises customers using older versions of the console can enable the PullServiceInstallerFeatureFlag. Contact VMware Support for assistance in updating the database with this feature flag.

3 While logged into Workspace ONE UEM, you must select an organization group (OG) to install the relay service onto. It is a best practice that the devices you intend to pull content from the relay service are managed under the same OG (or lower) as that which the pull service is installed.

4 Once you have selected an OG that is equal to or above the OG that manages the devices that you want to use the pull service, navigate to **Groups & Settings > All Settings > System > Enterprise Integration > Pull Service Installers** and select the **Download Configuration** button. The **Download pull service configuration** screen displays.

5 Enter and confirm the **Certificate password** you want to use for the pull service certificate and select **Download** to save the ZIP file. Save the ZIP file to the same directory as the pull service installer you downloaded in step 1.

6 Extract the contents of the ZIP file to the directory in which you saved it.

Result: The ZIP file contains the pull service installer CONFIG file which is a file you require to run the installer. You should have the Windows EXE (or Linux BIN) Pull Service Installer file plus the CONFIG file extracted from the ZIP file together in the same directory. If you are installing the Linux variant, you might also have one of the NTLM enablers from step 2.

7 Run the Pull Service Installer, supplying the password you entered in step 5 when prompted. Continue with the remainder of the installation.

   a Select whether you wish to use an outbound proxy and, if applicable, enter the proxy configuration.

   b Select whether you wish to use Custom Discovery Text. The Pull Service uses Discovery Text to identify itself when communicating with Workspace ONE UEM servers. By default, Pull Service v1.0 and v2.0 use the local IP Address and MAC Address of the server but a custom value can be used instead. This is necessary to identify each Pull Server separately in rare cases where multiple Pull Servers do not have unique and distinct local IP Address and MAC Address values.

   When adding Custom Discovery Text, please note that the value entered in the installer is not the full Discovery Text value. The installer will append it to a unique GUID and store it in the `appsettings.json` file. This file can be found in the install directory of the Pull Service under the `/bin` folder.

   The Relay Server configuration in the Workspace ONE UEM console must use this Discovery Text value. For more information, see Configure a Relay Server.

8 ** If you are installing the Linux version and you want to configure an NTLM proxy authentication, the "gss-ntlmssp" package must be installed. Select the download link that corresponds to your Linux flavor and install it now.

   - For CentOS, download the following RPM file, https://centos.pkgs.org/7/epel-x86_64/gssntlmssp-0.7.0-1.el7.x86_64.rpm.html

Using the .NET Core Pull Service, Best Practices

- Do not run multiple instances of pull service (old or new versions) on the same machine.
- To ensure that devices can access the pull service without issue, install the pull service and its CONFIG file to an OG equal to or above the OG that manages those devices.
- Once the pull service is installed, avoid selecting the **Regenerate** button on the Pull Service Installers settings page. Regenerating the certificate after the pull service is running breaks communication between the Workspace ONE UEM console and the pull service client. If you must regenerate the certificate, then you must rerun the pull service installer.
- On-premises customers must generate and download the PSinstaller.CONFIG file directly from Global OG and not from Customer OG. This configuration enables an easy relay server move from OG 1 to OG 2 without having to reinstall the pull service every time. For more information, see Move an Existing Pull Relay Server from One Organization Group to Another.

Using the .NET Core Pull Service, Starting and Stopping the Pull Service

- Start/stop service on **CentOS and RHEL**.

```
/etc/init.d/awpullservice {start|stop|status|restart)
```

  - Start the pull service.

```
service awpullservice start
```

  - Stop the pull service.

```
service awpullservice stop
```

  - Displays status of the pull service.

```
service awpullservice status
```

  - Restarts the pull service.

```
service awpullservice restart
```

- Start/stop service on **Ubuntu**. Run this SH file, which is located in the pull-service/bin folder.
  - Start the pull service.

```
sh start.sh
```

  - Stop the pull service.

```
sh stop.sh
```

- Start/stop service on **Windows**.
  - Navigate to **Run > services.msc**, locate the **Airwatch Pull Service**, and select start, stop, or restart.

Using the .NET Core Pull Service, Appsettings.json Tweaks

The appsettings.json file settings presented here are the default values which you can change to modify the server's default behavior. The appsettings.json file is located in the pull service\Bin folder.

If the "discoveryText" value is set as null in the appsettings.json, then the pull service assembles and posts a default discovery text. This discovery text is composed of the merging of the pull server's IP address and Mac address in place of the null value.

The pull service must be manually restarted each time you change values in appsettings.json to apply the new settings.

- "performSystemTaskIntervalMin": "720" (in minutes), the interval when the pull service logs are sent to the content pull endpoint. For more information, see Enable Pull Service Logging.

- "clientPostTimeOut": "240" (in minutes), the amount of time the connection remains active before timing out. This time frame can increase as the size of your files increase to avoid connection timeouts.
- "waitTimeOnFailure": "300" (in seconds), the amount of time you allow the pull service to rediscover the relay server if there is a connection error.
- "logMaxEntries": "10000", the maximum number of log entries in a single log file.
- "maxLogFiles": "5", the maximum number of logs generated.
- "maxLogFileSize": "2000000" (in bytes), the maximum size a log file is allowed to get.
- "enableConsoleLog": "false", for developmental purposes. Enable to activate the console log.
- "proxyBypassOnLocal": "false", enable/disable the bypass proxy server for local addresses.

Mirgating to the .NET Core Pull Service

The .NET Core pull service installer does not support an automatic migration with custom discovery text. When you reinstall the same version of the PS or upgrade the PS to a new version, the custom discovery text value is different each time. For this reason, make a note of the complete discovery text value under appsettings.json ("discoveryText") or from the Undiscovered Pull Relay servers page on the console after a fresh installation for the first time (for example: a6af8505-3d0c-4490-ab02-14db2fa3a80c_PSmachine1).

Saving the discovery text value is useful for when you accidentally uninstall the pull service or when you must move the relay server from one OG to another or from one server to another.

Follow this task when you do not want to use custom discovery text after migration and you want to continue using the same default discovery text. The default discovery text is the pull server's IP address plus Mac address.

1 Install the pull service v2.0 as described in the above section, **Install Pull Relay Service v2.0**, setting Custom Discovery Text, per your preference.

> **Note** If you want to apply Custom Discovery Text during the pull service v2.0 installation, the Relay Server configuration in the Workspace ONE UEM console must be updated after the installation.

2 From the pull server, navigate to the `/bin` folder of the installation directory, locate the `appsettings.json` file, and copy the full Discovery text (GUID_CustomText) with the Ctrl-C shortcut.

3 From the UEM console, navigate to **Devices > Provisioning > Relay Servers > List View** and select the new pull relay server.

4 Select the **Console Connection** tab and paste (Ctrl-V) the Discovery Text from the old server onto the **Discovery Text** text box of this new server.

5 **Save** and Reactivate the server.

6 Restart the AirWatch Pull Service.

Recovering a Pull Relay Server v2.0 with Custom Discovery Text

Failure to recover the uninstalled or corrupted pull relay server may cause Workspace ONE UEM console to consider it a brand new server, causing the Pull Service to re-download all files that are already on the server. These steps apply for both Linux and Windows.

1 Take note of the current Custom Discovery Text of the corrupted/uninstalled Pull Relay Server. There are two ways to find the current Custom Discovery Text. You must choose…

   - …from the **UEM console**, navigate to **Devices > Provisioning > Relay Servers > List View**, select the corrupted/uninstalled pull relay server, then open the **Console Connection** tab where the Custom Discovery Text is displayed. Go directly to Step 2.
   - …from the **pull server**, navigate to the install directory of the Pull Service v2.0 and open the `appsettings.json` file located in the `/bin` folder. The Custom Discovery Text can be found listed in the JSON file under "discoveryText".

2 Navigate to **Groups & Settings > All Settings > Enterprise Integration > Pull Service** and download the latest Pull Service installer.

3 Uninstall the existing Pull Service from the Linux machine.

4 Run the Pull Service installer you downloaded in step 2. During the installation process, you are presented with the option to enter your own discovery text. Enter the custom discovery text your found in step 1.

   - If using a Linux silent install, make sure you use the `-DDISCOVERY_TEXT` switch.

5 Restart the pull service.

The server resumes functioning as it did before, using the exact same discovery text. The already queued items and new items are served to the same directory path as before.

Device Staging

3

Workspace ONE UEM powered by AirWatch lets you stage a device quickly to enroll and prepare it for production use. A staging package connects a device to a Wi-Fi connection, installs the Workspace ONE Intelligent Hub, and enrolls the device without end-user input.

Staging Basics

The Rugged Enrollment Configuration Wizard simplifies creating staging packages. With the wizard, everything you need for a Staging Package is created in a step-by-step process. You can also manually create Staging Packages.

Staging packages are created as part of the product provisioning process. You can include profiles, applications, and files/actions as part of the staging package depending on the device platform.

You have several methods for enrolling a rugged device through staging. Barcode Enrollment creates a staging package associated with a barcode that you scan to stage the device. The Stage Now client is exclusive to Android devices with Zebra MX version 7.1+ under Android Nougat and later. Sideloading packages are transferred to a device instead of being scanned or downloaded.

Advantages of Device Staging for Android

Using Device Staging to enroll rugged Android devices adds functionality not found in non-staging methods of enrollment, such as manually enrolling through Workspace ONE Intelligent Hub. This functionality includes the following.

- WifiConfig cannot configure Fusion settings for Motorola devices. You must push the WifiConfig.apk as an internal application after enrollment to configure the settings. Extract the WifiConfig.apk from a sideload staging bundle inside the Workspace ONE Intelligent Hub folder of a device and upload it to the Workspace ONE UEM console as an internal application.
- Product Persistence does not support the Workspace ONE Intelligent Hub enrollment method. Products marked for persistence still download to the device but an Enterprise Reset removes all products. Persisted products do not automatically reinstall following an Enterprise Reset when the device reboots.

This chapter includes the following topics:

- Stage Devices With the Enrollment Configuration Wizard

- Stage Devices With a Manually Created Staging Package
- Add a Manifest to Your Staging Package
- Wi-Fi Profiles for Staging
- Barcode Staging
- Sideload Staging Packages
- AirWatch CAB Creator, WinRugg

Stage Devices With the Enrollment Configuration Wizard

Simplify rugged device enrollment in Workspace ONE UEM powered by AirWatch through the Enrollment Configuration wizard. This wizard helps you complete each step in creating a staging package for your Android and Windows Rugged devices.

Procedure

1 Navigate to **Devices > Lifecycle > Staging** and select the **Configure Enrollment** button.

2 Select the device platform you want.

3 Select the staging enrollment type.

The settings you must configure change based on the enrollment type selected.

- Generate a QR Code with the Enrollment Configuration Wizard, Android (Android 7.0 and later only) – Create a QR Code to scan with your Android Work Managed devices to quickly stage the device. The wizard simplifies the configuration process.
- Generate a Barcode Staging Package with the Enrollment Configuration Wizard (Android 6.x and earlier only) – Create a barcode to scan with your Zebra rugged devices to quickly stage the device. The wizard simplifies the barcode configuration process.
  - Zebra devices running Android 7.0 and later must follow Generate a Barcode Staging Package.
- Generate a Sideload Staging Package with the Enrollment Configuration Wizard, Android and Windows Rugged – Download a sideload staging package to automatically configure and enroll a device. For Android devices, this method only supports Android (Legacy) enrollment. Files and commands are transferred to devices through a USB connection.
- Enroll with Web Enrollment, Windows Rugged (Windows Rugged only) – Configure a Web URL that users can browse to from a device to download a Staging Package that automatically configures and enrolls the rugged device. In this method, users are guided through an enrollment wizard.

4 Select **Configure**.

Generate a QR Code with the Enrollment Configuration Wizard, Android

After selecting QR Code enrollment in the Enrollment Configuration wizard, create a QR Code to scan with your Android 7.0 or later devices to stage the device quickly. The wizard simplifies the staging configuration process.

Procedure

1 After taking note of the prerequisites, select **Configure** to begin.

2 You can connect the device to Wi-Fi before enrollment by enabling the **Wi-Fi** toggle. Enabling this option displays the following settings.

Setting	Description
SSID	Enter the Service Set Identifier, more commonly known as the name of the Wi-Fi Network.
Password	Enter the Wi-Fi password for the entered SSID.

3 Select **Next**.

4 Select the Workspace ONE Intelligent Hub to push to devices during staging. The default selection is Use latest Workspace ONE Intelligent Hub.

To enroll devices using a specific version of Workspace ONE Intelligent Hub, select **Hosted on an external URL**. You must then download the APK for Intelligent Hub from the Workspace ONE Resources Portal and host the file in an external location. Enter the address of this location in the URL text box to instruct devices to download Intelligent Hub from this location.

5 Select **Next**.

6 Set the **Enrollment Details** settings. To use a token-based authentication, leave both commands disabled.

Setting	Description
Organization Group	Enable and select the organization group the QR Code staging package uses.
User name	Enable and configure login credentials. Enter the Workspace ONE UEM account user name.
Password	Enter the corresponding password.

7 Select **Next**.

8 The **Summary** page allows you to **Download File** of the PDF. You can also **View PDF** to see a preview of your **QR Code Format** selections.

Generate a Barcode Staging Package with the Enrollment Configuration Wizard

After selecting Barcode enrollment in the Enrollment Configuration Wizard, create a barcode to scan with your Zebra Android and Windows Rugged devices to stage the device quickly. The wizard simplifies the staging configuration process.

- This method applies to Zebra devices enrolled as Android Legacy and running Android 6.x or earlier.
- Zebra devices running Android 7.0 and later must follow Generate a Barcode Staging Package.

Procedure

1 After taking note of the prerequisites, select **Configure** to begin.

2 Stage the devices by selecting the **Relay Server**.

   a If you do not have a relay server created, select **Add Relay Server**.

   The list of relay servers populates from any relay servers created for the organization group or the parent organization groups.

3 Select **Next**.

4 (Optional) Select a **Wi-Fi Profile** that devices use to connect to the relay server and download the Workspace ONE Intelligent Hub.

   a If you do not have a Wi-Fi profile created, select **Add Wi-Fi profile**.

   You cannot create a Wi-Fi profile through the wizard that uses a certificate authentication. The Wi-Fi profile created is used for staging and remains on the device after enrollment.

5 Select **Next**.

6 Select the Workspace ONE Intelligent Hub to push to devices during staging. If you do not have an Workspace ONE Intelligent Hub added, select Add Workspace ONE Intelligent Hub to upload a Workspace ONE Intelligent Hub Package if necessary. For more information about uploading Hub agents, see Upload the Workspace ONE Intelligent Hub APF File, Upgrade File-Action.

7 Select **Next**.

8 Enter the **Stage User** credentials.

Settings	Descriptions
Name	Enter the name of the staging package.
Description	Enter a description of the staging package.
Owned By	Select the organization group that owns the staging package.

Settings	Descriptions
Enrollment User	Enter the user name of the user. If you do not have a user, select **Add User**. The user must be a basic user account. Do not use staging users or multi-user staging.
Password	Enter the password of the user.

9 Select **Next**.

10 Set the **Barcode** settings.

Setting	Description
Organization Group	Select the organization group the staging package uses.
Universal Barcode	Enable to create a universal barcode enrollment so devices can be enrolled without automatically assigning an organization group. With this setting enabled, the Workspace ONE Intelligent Hub prompts users to enter an organization group after beginning the staging process. **Android Only Note:** Since you can only enroll in Android (Legacy) mode with this barcode, take care to select an OG that is configured for Android (Legacy) enrollment.
Require Password.	Create an alphanumeric password (maximum 99 characters) used to unlock the staging package encryption on the end-user device immediately after enrollment.
Barcode Format	Select the barcode format for the devices you want to enroll.

11 Select **Save**.

What to do next

Navigate to the **Summary** page and you can **Download File** of the PDF.

You can also **View PDF** to see a preview of your **Barcode Format** selections.

Android Only Note:

After you close this window, you can still make new barcodes using the options you have selected above by following the Generate a Barcode Staging Package step.

1 Navigate to **Devices > Lifecycle > Staging > List View**.

2 From the listing displayed, locate the staging package you created with the enrollment wizard.

3 Select this staging package by clicking the radio button to the left of the staging package name. The radio button is above the **Edit** icon ( ).

4 Some new buttons now display in the buttons cluster. Select the action that applies to your staging package.

Generate a Sideload Staging Package with the Enrollment Configuration Wizard, Android and Windows Rugged

You can create a sideload staging package to configure and enroll Android Legacy or Windows Rugged devices. Files and commands are transferred to devices through a USB connection. The wizard simplifies the staging configuration process.

After selecting Sideload enrollment in the Stage Devices With the Enrollment Configuration Wizard, create a sideload staging package using the wizard.

Prerequisites

In order for you to be allowed to create a sideload package, you must be in a **Customer** type organization group.

Procedure

1. Select the Workspace ONE Intelligent Hub to push to devices during staging. If you do not have an Workspace ONE Intelligent Hub added, select Add Workspace ONE Intelligent Hub to upload a Workspace ONE Intelligent Hub Package if necessary. For more information about uploading Hub agents, see Upload the Workspace ONE Intelligent Hub APF File, Upgrade File-Action.
2. Select **Next**.
3. In the **Stage User** tab, enter the user credentials.

Settings	Descriptions
Name	Enter the name of the staging package.
Description	Enter a description of the staging package.
Owned By	Select the organization group that owns the staging package.
Enrollment User	Enter the user name of the user. The user must be a basic user account. Do not use staging users or multi-user staging.
Password	Enter the password of the user.

4. Select **Next**.
5. In the **Sideload** tab, enter the settings.

Settings	Descriptions
OG	Select the organization group the staging package uses.
Universal	You can create a universal enrollment so devices can be enrolled without automatically assigning an organization group. This option allows you to enroll devices without needing a Sideload Staging Package for each organization group. The Hub prompts you to enter an organization group after the staging process begins.

6 In the **Summary** tab, review the components and select the **Download File** button, select a save location where you can find the ZIP file, then select **Save**.

Enroll with Web Enrollment, Windows Rugged

After selecting Web enrollment in the Rugged Enrollment Configuration wizard, create a staging package to enroll devices by sending end users to a URL to enroll. The enrollment wizard simplifies the staging configuration process.

Procedure

1 Select **Configure Hub** to configure the Workspace ONE Intelligent Hub for web enrollment.

The default file path to the Workspace ONE Intelligent Hub CAB file displays.

2 If you want to use a different CAB, select **Disable** and then select the CAB file you want to use. Select the CAB file for both Windows Mobile, Windows CE, and Windows x86 CE devices.

a To upload your own CAB file, select **Add Application**.

3 Select **Next**.

4 Navigate to **Groups & Settings > All Settings > System > Advanced > Site URLs** and note the URL found in the **MDM Enrollment URL** text box. Send end users to this URL to begin enrollment.

a If you do not have access to the above nav path, use the following enrollment URL instead, replacing `<DS_Hostname>` with the device server hostname of **your** environment.

`https://<DS_Hostname>/DeviceManagement/Enrollment`

Stage Devices With a Manually Created Staging Package

Workspace ONE UEM powered by AirWatch lets you manually stage your devices to connect to Wi-Fi, download the Workspace ONE Intelligent Hub, and enroll automatically with a Staging Package. This method does not use the Rugged Enrollment wizard.

Creating a staging package manually means only specifying the user account, Organization Group, and Hub version for enrollment.

Procedure

1 Navigate to **Devices > Lifecycle > Staging** and select the **Add Staging** button.

2 Select the Platform for which you want to create a staging configuration.

The **Staging Add** screen displays.

3 Complete the required text boxes on the **General** tab.

Settings	Description
Name	Enter the name of the staging configuration.
Description	Enter the description of the staging configuration.

Settings	Description
Owned By	Select the organization group under which the staging package applies.
Enrollment User	Enter the user name of the enrollment user. You can search for and select an existing user by clicking the magnifying glass icon. You can also add a user by selecting **Add User** at the bottom of the drop-down menu.
Password	Enter the password for the enrollment user. You have the option of keeping the password redacted or displaying it as written.
Hub	Select an existing Workspace ONE Intelligent Hub package from the drop-down listing which is downloaded during staging. You can also add the Workspace ONE Intelligent Hub package by selecting Add Workspace ONE Intelligent Hub at the bottom of the drop-down menu. These agents are uploaded as the Workspace ONE Intelligent Hub Package. See Upload the Workspace ONE Intelligent Hub APF File, Upgrade File-Action for more information.

4 Select **Save**.

What to do next

You can configure additional settings by adding them to the Staging Package Manifest. Wi-Fi profiles can also be added to the manifest for sideloading or embedded in staging barcodes. Proceed to the Add a Manifest to Your Staging Package task.

Add a Manifest to Your Staging Package

You can add a **Manifest** to any staging package in Workspace ONE UEM. The manifest can contain any product provisioning components including applications, profiles, Files/Actions, and Event Actions.

While these components can be pushed through product provisioning immediately after enrollment, there are some use-cases for adding these components in the Staging Package Manifest. Consider doing so if they are critical to communication between the devices and Workspace ONE UEM servers.

For example, profiles can be added to a Windows Rugged Sideload Staging Package in order to have the device sync its date and time with an NTP server or to force traffic through an outbound proxy.

The manifest can be added by navigating to **Devices > Lifecycle > Staging**. Either select the edit icon ( ) to the left of an existing staging package in the listing and proceed to step 1 below or create a new staging package by selecting the **Add Staging** button. Select the platform then proceed to the **Manifest** tab.

Prerequisites

If you plan to add Applications, Profiles, Files/Actions or Event Actions, you must create these components before adding them to the manifest. Create these components by navigating to **Devices > Provisioning > Components**.

Procedure

1 After finishing the **General** tab of the Staging window, continue to the **Manifest** tab.

2 Select the **Add** button.

The **Add Manifest** screen displays.

3 Select the **Action(s) To Perform** during staging.

Action(s) to Perform Drop-Down Menu	Settings
Install Profile	In the **Profile** text box, select the profile to install during the staging configuration. This component must be made before adding it to the manifest.
Uninstall Profile	In the **Profile** text box, select the profile to remove during the staging configuration.
Install Application (Android Only)	In the **Application** text box, select the App to install during the staging configuration. This component must be made before adding it to the manifest.
Uninstall Application (Android Only)	In the **Application** text box, select the App to remove during the staging configuration.
Install Files/Actions	In the **Files/Actions** text box, select the Files/Actions component to install during the staging configuration. This component must be made before adding it to the manifest.
Uninstall Files/ Actions	In the **Files/Actions** text box, select the Files/Actions component to remove during the staging configuration.
Reboot (Android Only)	**Reboots** the device during the staging configuration. This action works best as the last step of the manifest.
Warm Boot/Cold Boot (WinRugg Only)	**Warm Boot** reboots the device during the staging configuration. This action works best as the last step of the manifest. **Cold Boot** shuts down the device, forcing a restart by the end user. This action works best as the last step of the manifest.
Install Event Action (Android and WinRugg Only)	In the **Event Action** text box, select the Event Action component to install during the staging configuration. This component must be made before adding it to the manifest.
Uninstall Event Action (Android and WinRugg Only)	In the **Event Action** text box, select the Event Action component to remove during the staging configuration.

4 **Android and Windows Rugged only** - Enable or disable **Persistent through enterprise reset**. Enable to keep the profile, application, files/actions, or event action on the device after an enterprise reset. For more information, see Product Persistence, Android, and WinRugg.

5 When finished with the single Manifest action, select **Save**.

6 Select **Add** again to add additional Manifest actions. You can adjust the order of manifest steps using the up and down arrows in the Manifest list view. You can also edit or delete a manifest step.

7 When you are finished adding actions, select **Save**.

What to do next

View the newly created staging profile in the List View. Take additional actions on the profile from the menus on the right.

- **Edit** your configuration.
- **Copy** your profile.
- Android and Windows Rugged only: Select **Barcode** and finish the text boxes on the **Generate Barcode** subpage.

Wi-Fi Profiles for Staging

A Wi-Fi profile in your staging package enables enrollment without device end user interaction. Devices must connect to a network before starting enrollment with Workspace ONE UEM powered by AirWatch.

You must create a Wi-Fi profile by navigating to **Devices > Provisioning > Components > Profiles** in order to add them to the manifest of a Staging Package. Navigate to the **General** settings of the profile and set the **Profile Scope**:

You can define whether a Wi-Fi profile can be used for staging by setting its Profile Scope.

- **Staging** – Can only be used for staging.
- **Production** – Can only be pushed after enrollment. This includes the option of pushing the profile through product provisioning.
- **Both** – Can be used for staging or pushed after enrollment.

Barcode Staging

Create a barcode with Workspace ONE UEM powered by AirWatch and use it to auto-enroll Android and Windows Rugged devices. For Android, this method is supported with Zebra and Honeywell devices. Barcode Staging reduces the process to a simple barcode scan which configures the device with a staging package.

You can also create universal barcode staging which does not automatically assign an organization group while enrolling the device. This generic barcode allows you to create one staging enrollment for all devices and assign the device to an organization group later, as needed.

Barcode enrollment is only available on devices running the Rapid Deployment Client or Zebra's Stage Now client. The Rapid Deployment client supports FTP and FTPS relay servers. Zebra's Stage Now client supports FTP, FTPS, and HTTPS relay servers.

Use the Rugged Enrollment wizard to simplify the creation of barcode staging packages. The wizard enables you to create all the necessary components of a staging package in one place. For more information, see Stage Devices With the Enrollment Configuration Wizard.

Barcode enrollment is only supported on the following devices.

Android

- Zebra Rugged devices with MX running Android 6.x (Marshmallow) and earlier (Rapid Deployment barcodes only).
- Zebra Rugged devices with MX version 7.1 or later running Android 7.0 (Nougat) or later and with Android Hub version 8.2 or later (Stage Now barcodes only).

You can create barcodes to enroll Honeywell Android rugged devices in the Workspace ONE UEM console by using the EZconfig utility for Android legacy devices only.

Windows Rugged with Rapid Deployment

- MC45
- MC55
- MC65/67
- MC75
- MC3090
- MC3190
- MC9090
- MC9190
- MC92N0

Generate a Barcode Staging Package

Create a barcode to scan with your Zebra or Honeywell rugged devices to stage the device quickly.

Prerequisites

You must create a staging package before you generate a barcode that uses it. See Stage Devices With a Manually Created Staging Package.

The staging user for the staging package must be a basic user account. Do not use staging users or multi-user staging.

Procedure

1 Navigate to **Devices > Lifecycle > Staging**.

2 Select the radio button to the left of the name of the staging package.

A new row of buttons displays under the **Add Staging** and **Configure Enrollment** buttons.

3 Select the **Barcode** or **Stage Now Barcode** or **Honeywell Barcode** button.

4 Select the **Staging Options**.

Settings	Descriptions
Organization Group	Select the organization group the staging package uses.
Universal Barcode	Enable to create a universal barcode enrollment so devices can be enrolled without automatically assigning an organization group. **Android Only Note:** With this setting enabled, the Workspace ONE Intelligent Hub prompts you to enter an organization group after beginning the staging process. If you are enrolling devices into Work-Managed Mode with this barcode, take care to select an OG that is configured with Android EMM Registration settings. Enabling this box repopulates the **Staging Relay Server** and **Staging Profile** with applicable options.
Staging Relay Server	Select the staging relay server that hosts the staging content.
Staging Profile	Select the staging Wi-Fi profile and apply it to the enrolled device.
Require Password.	Enable to create an alphanumeric passphrase (maximum 99 characters) to be used to unlock the staging package encryption on the end-user device.
Device Owner Mode	**Android Only.** Enable to enroll the device into Android Work Managed mode. This option can only be enabled when the **Universal Barcode** option is enabled.

5 Select the **Barcode Format** options. **Android Only Note:** Android admins can enter the optional **Barcode Instructions** to be included on the barcode PDF output page.

6 Select **View PDF**.

This option generates a preview of the barcode PDF output page for end users to scan.

7 **Save** the PDF file.

Enroll Zebra Devices with Stage Now Barcode, Android

You can enroll Zebra Android devices with a Stage Now barcode straight from the console for Workspace ONE UEM powered by AirWatch. The Stage Now staging client is Zebra's Android solution for staging Zebra devices and preparing them for production use.

Workspace ONE UEM supports Stage Now given the following conditions and limitations.

For more information on Zebra Mobility, see Zebra Mobility Extensions (MX) and Full MX Feature Matrix.

If you plan to enroll Zebra devices in Work-Managed Device Mode with a Stage Now barcode, take the following steps.

Prerequisites

- Zebra devices must be running Android 7.1 (Nougat) with MX version 7.1 or later.
  - Zebra devices running Android 6.x (Marshmallow) and earlier must continue to use Rapid Deployment as the default staging client.
- You must have Android Hub version 8.2 or later uploaded to the console as the Workspace ONE Intelligent Hub Package.
- Relay Servers
  - Stage Now supports only FTP Relay Servers set to passive mode. FTP Relay servers in active mode are not supported and do not function with the Stage Now client.
  - HTTPS endpoints for the relay server are supported so long as they are configured using the web server of your choice (for example IIS). You must also select the HTTPS protocol when you Configure a Relay Server.
- Ensure the **Stage Now URL** setting, found in **Groups & Settings > All Settings > System > Advanced > Site URLs**, is set to the appropriate URL.
  - If your on premises environment is configuring your own Stage Now server, then place your custom URL in this text box.
  - If your on premises environment is not configuring your own Stage Now server, then you simply must open your networks to allow access to the URL listed here.
  - SaaS environments do not need to change this text box.
- There must be no Google account present on the device while attempting Stage Now enrollment in Work-Managed Mode.

Procedure

1. Use the Organization Group selector to select the OG you want to configure for your Android devices.
2. Navigate to **Groups & Settings > All Settings > Devices & Users > Android > Android EMM Registration** and select the **Enrollment Restrictions** tab.

3 Complete the following settings.

Setting	Description
Current Setting	Select **Override** and affect changes to the OG you selected in step 1.
Define the enrollment method for this organization group.	This setting determines how this OG treats Android (Legacy) devices. Select from among the following settings. **Always use Android** – This setting enables the **Device Owner Mode** slider on the **Generate Stage Now Barcode** screen and makes it uneditable. This setting forces all Android (Legacy) devices that enroll in this OG to be in Device Owner Mode (or Work Managed Device Mode). **Always use Android (Legacy)** – This setting disables the **Device Owner Mode** slider from the **Generate Stage Now Barcode** screen and makes it uneditable. This setting forces all Android (Legacy) devices that enroll in this OG to be in Device Admin Mode. **Define Assignment Groups that use Android** – This setting enables the **Device Owner Mode** slider on the **Generate Stage Now Barcode** screen and makes it editable, allowing you the choice of enrolling Android devices in Device Owner Mode (Work Managed Device Mode) or enrolling them in Device Admin Mode. Select the **Smart Groups** you have assigned to the Zebra devices you want to enroll using these settings.

4 Follow the steps described in Generate a Barcode Staging Package to generate a Zebra StageNow Barcode.

5 Direct your end user to take the following steps after they take possession of the newly enrolled device.

a Start the device from a "factory settings" state.

b Ensure that there is no Google account on the device.

c Proceed through the Setup Wizard or scan the "skip setup wizard" barcode provided by Zebra.

d Open the Stage Now application.

e Scan the barcode you made in step 4.

The device is enrolled into Work-Managed mode automatically.

Enroll Honeywell Devices with Staging Barcode, Android

You can enroll Honeywell Android devices with a staging barcode straight from the console for Workspace ONE UEM powered by AirWatch. The Honeywell Enterprise Provisioner is Honeywell's Android solution for staging devices and preparing them for production use.

- Ensure the **Honeywell Enterprise Provisioner URL** setting, found in **Groups & Settings > All Settings > System > Advanced > Site URLs**, is set to the appropriate URL.
  - If your on premises environment is configuring your own Honeywell Enterprise Provisioner server, then place your custom URL in this text box.

  - If your on premises environment is not configuring your own Honeywell Enterprise Provisioner server, then you must open your networks to allow access to the URL listed here.
  - SaaS environments do not need to change this text box.
- Protocols supported: FTP, FTPS, HTTPS
- Advanced Staging Manifest is not currently supported.

Prerequisites

If you plan to enroll Honeywell devices with a staging barcode, take the following steps.

Procedure

1 Use the Organization Group selector to select the OG you want to configure for your Android devices.

2 Navigate to **Groups & Settings > All Settings > Devices & Users > Android > Android EMM Registration** and select the **Enrollment Restrictions** tab.

3 Complete the following settings.

Setting	Description
Current Setting	Select **Override** and affect changes to the OG you selected in step 1.
Define the enrollment method for this organization group.	This setting determines how this OG treats Android (Legacy) devices. Select from among the following settings. **Always use Android** – This setting enables the **Device Owner Mode** slider on the **Generate Honeywell Barcode** screen and makes it uneditable. This setting forces all Android (Legacy) devices that enroll in this OG to be in Device Owner Mode (or Work Managed Device Mode). **Always use Android (Legacy)** – This setting disables the **Device Owner Mode** slider from the **Generate Honeywell Barcode** screen and makes it uneditable. This setting forces all Android (Legacy) devices that enroll in this OG to be in Device Admin Mode. **Define Assignment Groups that use Android** – This setting enables the **Device Owner Mode** slider on the **Generate Honeywell Barcode** screen and makes it editable, allowing you the choice of enrolling Android devices in Device Owner Mode (Work Managed Device Mode) or enrolling them in Device Admin Mode. Select the **Smart Groups** you have assigned to the Honeywell devices you want to enroll using these settings.

4 Direct your end user to take the following steps after they take possession of the newly enrolled device.

  a Start the device from a "factory settings" state.

  b Ensure that there is no Google account on the device.

  c Proceed through the Setup Wizard.

d Open the Enterprise Provisioner or the EZ Config application.

e Scan the barcode.

The device is enrolled automatically.

What to do next

You can optionally continue to the next step of making a Honeywell barcode by proceeding to the next step, Generate a Barcode Staging Package with the Enrollment Configuration Wizard.

How Do You Configure a Zebra Work Managed Device with Stage Now, Relay Servers, and CICO Launcher

Workspace ONE UEM powered by AirWatch lets you provision a Zebra Rugged device with a Stage Now barcode that installs a host of useful configs including a Wi-Fi profile, Relay Server connectivity, and a Check-in/Check-out launcher.

Use Case: You want to provision a Zebra device with a way to check the device in and out making it suitable for multi-user use. These devices also need a relay server to host provisioned content, a Wi-Fi profile to connect it to the network, and a Stage Now barcode to deliver it all. This use case represents a "one-stop" provisioning solution for Zebra Android devices.

Prerequisites

- The Zebra device must begin in a factory reset state with Android 7.0 (Nougat) or later and MX version 7.1 or later.
- The Zebra device must also have VMware Workspace ONE Intelligent Hub version 8.2 or later.

Note Devices without these minimum specifications will not be included in this Stage Now barcode load out.

Procedure

1 Create a staging-based relay server. This is the server that hosts all the staging content.

a Navigate to **Devices > Provisioning > Relay Servers > List View**.

b Select **Add > Add Relay Server**.

c Configure the relay server options per your preferences and specifications. For the purpose of this use case, select the **Assignment** tab and in the **Staging Server** section, select one or more organization groups that manage your Zebra devices.

d **Save** the Staging Relay Server.

2 Create a production-based relay server. This is the server that hosts all the content to be used on a day-to-day basis **after** the conclusion of this use case.

   a While still in the **Relay Servers List View**, select **Add > Add Relay Server**.

   b Configure the relay server options per your preferences and specifications. For the purpose of this use case, select the **Assignment** tab and in the **Production Server** section, select the same organization groups you selected for the Staging Relay Server in the previous step.

   c **Save** the Production Relay Server.

3 Create the CICO Launcher profile. The launcher is the component that facilitates the check-in / check-out process.

   a Navigate to **Devices > Provisioning > Components > Profiles** and select **Add Profile > Android**.

   b For **Profile Scope**, set the drop-down menu to **Production**.

   c Select the **Launcher** payload from the list to the far-left of the window.

   d Select the **Configure** button and select the app options for the launcher per your preferences.

   e Select **Settings**, include an **Administrative Passcode**, then select **Close**.

   f **Save** the Launcher configuration.

   g Select **Save** again to save the new Android Provisioning Profile.

4 Create a Wi-Fi profile. This profile ensures that the Zebra device has Wi-Fi connectivity, which is important.

   a While still at **Devices > Provisioning > Components > Profiles** from the previous step, select **Add Profile > Android**.

   b For **Profile Scope**, set the drop-down menu to **Staging**.

   c Select the **Wi-Fi** payload from the list at the far-left of the window.

   d Select the **Configure** button and complete the Wi-Fi connection details. Make certain to select a Wi-Fi password that complies with the minimum length and complexity rules for passcodes.

   e **Save** the Wi-Fi Profile.

5 Create a smart group that contains all the Zebra devices you want to target for this use case. This smart group is how the CICO Launcher profile gets installed on your Zebra device fleet.

   a Navigate to **Groups & Settings > Groups > Assignment Groups** and select the **Add Smart Group** button.

   b Name the smart group something like "Zebra Devices CICO Launcher".

   c Under **Organization Group**, select the OG that manages your Zebra devices.

d Under **Platform and Operating System**, select **Android Greater Than or Equal To Android 7.0.0**.

e Under **Enterprise OEM Version**, select **Motorola/Zebra Greater Than or Equal To Zebra Mobility Extensions Version 7.1**.

f **Save** the smart group.

6 Create a product, include the CICO launcher, and assign it to the smart group.

a Navigate to **Devices > Provisioning > Product List View** and select the **Add Product** button.

b Select the **Android** platform.

c Enter the **Name** of the product and under **Managed By**, select the OG that manages your Zebra devices.

d Under **Smart Groups**, select the smart group you made in the previous step from the drop-down menu.

e Select the **Manifest** tab and then select the **Add** button.

f Under **Actions to Perform**, select **Install Profile**.

g Under **Profile**, select the CICO Launcher Profile you made in step 3.

h **Save** the product.

7 Configure an Android Staging Profile.

a Navigate to **Devices > Lifecycle > Staging > List View** and select **Add Staging > Android**.

b Complete all required options on the **General** tab.

- **Owned By** - Select the organization group that manages your Zebra devices.
- **Enrollment User** - You must select a device user who is configured as a staging multi-user. You can edit an existing user to be a staging multi-user by taking the following steps.

  a Navigate to **Accounts > Users > List View**.

  b Select the Edit icon ( ) to the left of the user you want to change.

  c Select the **Advanced** tab and in the **Staging** section, **Enable Device Staging** and then enable **Multi User Devices**.

  d **Save** the edit.

- **Hub** - Select version 8.2 or later of the Workspace ONE Intelligent Hub app.

c **Save** the Staging Profile.

8 Configure the Stage Now barcode.

   a Navigate to **Devices > Lifecycle > Staging > List View** and locate the Android Staging Profile you made earlier.

   b Select the radio button to the left of the Android Staging Profile. Some new buttons display under the main buttons.

   c Select the **Stage Now Barcode** button. The **Generate Stage Now Barcode** screen displays.

   d Under **Staging Relay Server**, select the staging-based relay server you made earlier.

   e Under **Staging Profile**, select the Wi-Fi profile you made earlier.

   f Select **Save** to save the barcode as a PDF file.

9 Ensure that the Stage Now URL setting is configured correctly, found in **Groups & Settings > All Settings > System > Advanced > Site URLs**. If you are operating in a SaaS environment, you can skip this step.

   - If your on-premises environment is configuring its own Stage Now server, then enter that custom URL in this text box.
   - If your on-premises environment is not configuring its own Stage Now server, then you must open your networks to allow access to the URL listed here.

10 Distribute the Zebra device and the PDF produced in a previous step to the staging individual and direct them to scan the following **Zebra Skip Setup Barcode** which runs the Stage Now application.

   a

   Figure 3-1. Zebra Skip Setup Barcode

11 Stage Now makes the request to scan a barcode.

12 Scan the Stage Now Barcode provided in PDF saved in an earlier step.

   Scanning the Stage Now Barcode triggers the following actions.

   - MDM information is obtained.
   - The Wi-Fi profile is downloaded and assigned.
   - A connection is made to the Staging Relay Server, enabling the device to pull the Staging Profile.
   - The Staging Profile contains the Workspace ONE Intelligent Hub app, ZebraMXService.apk, and credentials.bin file.

- The device installs the Workspace ONE Intelligent Hub app. The Hub opens and interprets the credentials.bin file, which in turn triggers the derived user information to be sent to the device services server for authentication and authorization.
- Device services validates the credentials.
- The Workspace ONE Intelligent Hub calls Google EMM registration service to obtain an Enterprise ID.
- Google responds with a masked Gmail account linked to the Enterprise and a token for authentication. This response allows the Workspace ONE Intelligent Hub app to continue enrolling.

13 Accept the privacy information and any other prompts presented by the Workspace ONE Intelligent Hub. For more information about Workspace ONE Mobile Applications Privacy, see https://kb.vmware.com/s/article/2960318.

The Zebra device enrolls into Workspace ONE UEM and the Workspace ONE Intelligent Hub app sends intents to the device to install additional products assigned in the UEM console including the CICO Launcher configured earlier.

14 The Zebra device downloads products from the Production Relay Server.

15 The VMware Launcher prompts the end user to log in. The device end user takes possession of the device by entering their enterprise credentials (AD/LDAP/Basic) and organization group as required.

Results

- VMware Launcher receives the credentials and passes off the authentication to the Workspace ONE Intelligent Hub app.
- The Workspace ONE Intelligent Hub app sends the user information to device services which validates the credentials.
- The Workspace ONE Intelligent Hub directs the VMware Launcher to load the configured layout and also directs the device to pull down any other products that are assigned to the end user. It pulls these products from the Production Relay Server, configured earlier.
- The device works with Google EMM to install any public applications that are assigned to the device.

What to do next

The device is successfully checked out and the end user can use the device in its full capacity.

Sideload Staging Packages

Workspace ONE UEM powered by AirWatch lets you create a sideload staging package to download and install onto your rugged devices to begin the auto-enrollment process. Sideload staging simplifies enrollment by combining all the required components into one package.

Android Only Note: sideload staging packages are only supported by Android legacy devices.

You can also create universal barcode staging to stage devices with a generic barcode that does not automatically assign an organization group when enrolling the device. This allows you to create one staging enrollment for all devices and assign the device to an organization group as needed.

Simplify creating a barcode staging package by using the Rugged Enrollment wizard. The wizard allows you to create all the necessary components of a staging package in one place. For more information, see Stage Devices With the Enrollment Configuration Wizard.

You can use the Sideload Staging Utility for Windows Rugged devices to sideload a staging package easily. The utility simplifies the process of installing a sideloading package onto the device with simple step-by-step instructions.

Generate a Sideload Staging Package with the Configuration Wizard

After selecting Sideload as the staging enrollment type in the Enrollment Configuration wizard, create a sideload staging package to download and install onto a device to configure and enroll the rugged device automatically.

Prerequisites

You must create a staging package before you create a sideload staging package. See Stage Devices With a Manually Created Staging Package.

The staging user for the staging package must be a basic user account. Do not use staging users or multi-user staging.

Following these steps for an Android staging package results in downloading a sideload staging package specifically for Zebra devices.

See also Enroll Honeywell Devices with Sideload Staging, Android and Enroll Platform OEM Devices with Sideload Staging, Android.

Procedure

1. Navigate to **Devices > Lifecycle > Staging**.
2. Select a previous staging package that you want to create a sideloaded staging package for. Select the **More** option and select **Staging Side Load** from the drop-down.
3. Select the **Organization Group** to which this staging applies.
4. (Optional for Android and Windows Rugged, not supported for QNX) Enable **Universal Barcode** to enable a universal enrollment so devices can be enrolled without automatically assigning an organization group.

   This allows you to enroll devices without needing a Sideload enrollment for each organization group.

   The Hub will prompt you to enter an organization group after beginning the staging process.
5. Select **Download** to start downloading the zip file of the staging sideload.

Enroll Zebra and Motorola Devices with Sideload Staging, Android

After creating a sideload staging package and downloading it, install it onto your Zebra Android devices to begin the enrollment process.

If you want to preconfigure a Wi-Fi connection into the sideload staging package, add a step to the manifest that installs the Wi-Fi profile. Otherwise, connect to a Wi-Fi network prior to starting enrollment.

Procedure

1. Download and install the Android Debug Bridge to the computer you want to use for staging devices.
2. Unzip the downloaded Sideload Staging ZIP file.
3. Depending upon the version of the OS that your device is running, you might need to download an updated stage.bat file from my Workspace ONE ™ Resource Portal.
   a. Visit `https://my.workspaceone.com`.
   b. Search for "VMware AirWatch Stage.bat for Zebra Sideload Staging" and download this file.
   c. Verify that this stage.bat file is in the root folder of the unzipped Sideload Staging ZIP file.
4. Establish a USB debug connection to the Android device.

   USB debugging must be enabled on the Android device. This setting is located in the device system settings under Developer Options.
5. Start the stage.bat file from the root folder of the unzipped Sideload Staging ZIP file.

   The stage.bat file copies files to the device and then uses intents to start the auto-enrollment process.

   The Workspace ONE UEM auto-enrollment screen displays on the device and shows progress.

   When auto-enrollment is complete, the Workspace ONE Intelligent Hub displays the main details screen.

Results

This script installs the MX Service and Hub then applies the Wi-Fi profile you defined in the staging manifest and any other manifest items. Once the Wi-Fi connects, the device auto-enrolls into Workspace ONE UEM.

Install a Sideload Staging Package, WinRugg

After creating a sideload staging package and downloading it, install it onto the Windows Rugged device to begin the enrollment process.

You can also use the Sideload Staging Utility for Windows Rugged devices to simplify the sideload staging process. See Use the Sideload Staging Utility, WinRugg.

Important If you want to preconfigure your Wi-Fi connection into the staging cab file, then use the advanced staging feature (Manifest tab on the staging profile) and add a step for installing a production Wi-Fi profile. If this step is not done, then the Wi-Fi profile must be manually set up on the device (preferably before running the staging cab).

Procedure

1. Unzip the file and connect your device to the staging machine through USB once the download is complete.
2. Manually create "\Program Files\AirWatch\Staging" directory on your device. You must add both the AirWatch and staging directories.
3. Copy the content of the unzipped staging file to the directory you created.

   The staging file contains several directory folders.
4. Open the "\Program Files\AirWatch\Staging\agent" directory and manually run the Workspace ONE Intelligent Hub cab file.

Results

The cab file installs the Workspace ONE Intelligent Hub, then the enrollment process completes and the Workspace ONE Intelligent Hub enrolls the device, assuming the device has network connectivity.

Enroll Honeywell Devices with Sideload Staging, Android

Enroll your Honeywell Android devices using a sideload staging package. Create a sideload staging package, download it, and install it onto your Honeywell Android devices to enroll them automatically.

Prerequisites

- You must create a staging package before you create a sideload staging package. See Stage Devices With a Manually Created Staging Package.
- Download the staging package and unzip the file to access the credentials.bin file.
- Download the latest Hub APK available on Workspace ONE UEM Resources. Contact your Workspace ONE UEM account manager for access to the APK.
- Download the latest Honeywell APK available on Workspace ONE UEM Resources. Contact your Workspace ONE UEM account manager for access to the APK.
- Download and install the Android Debug Bridge (ADB).

Procedure

1 Create a folder containing the following.

- Latest Workspace ONE Intelligent Hub for Android APK.
- Latest Honeywell OEM Service APK.
- Credentials.bin from the staging package.

2 Open a text editor such as Notepad. Copy the following chunk of text and paste it into the blank notepad.

```
adb push credentials.bin /sdcard/credentials.bin
adb install HoneywellService.apk
adb shell am start -a android.intent.action.MAIN -n
com.airwatch.admin.honeywell/.HoneywellActivity
adb install Hub.apk
adb shell am start -a android.intent.action.MAIN -n com.airwatch.androidagent/
com.airwatch.agent.ui.activity.SplashActivity -e hideui true
adb shell pm grant com.airwatch.androidagent android.permission.READ_EXTERNAL_STORAGE
adb shell am broadcast -a com.airwatch.agent.action.IMPORT_CREDENTIAL_XML -e file /sdcard/
credentials.bin --user 0
adb shell am broadcast -a com.airwatch.agent.action.AUTO_ENROLL --user 0
```

a Change the filenames and storage locations as needed.

3 Save the file as autoenroll_Honeywell.bat in the same directory as the other files.

4 Follow the steps described in Generate a Barcode Staging Package to generate a Honeywell Barcode for staging.

5 Connect a Honeywell Android device to your PC using an ADB connection. Ensure that the device is connected to Wi-Fi.

6 Run the autoenroll_Honeywell.bat file.

Enroll Platform OEM Devices with Sideload Staging, Android

You can set up a Sideload Staging bundle for devices using the Generic OEM Service. This procedure does not support Advanced Staging.

Procedure

1 Get the enrollment credentials.

a Create a Staging bundle in the console.

b Download the Sideload Staging Package.

c Unzip the Sideload Staging Package file and copy the 'credentials.bin' file inside the enrollment folder. Save this file for later.

2 Collect the necessary files for the device.

   a Get the latest Hub APK.

   b Get the OEM Service APK for your device.

   For more information, see Platform OEM Service, Android Provisioning.

   c Get the credentials.bin from the preceding step.

   d Place all these files in a folder on your PC.

3 Create your auto-enroll BAT file.

   a Using a text editor, add the following lines (change the filenames and storage locations based on your own configuration).

```
adb push credentials.bin /sdcard/credentials.bin
  adb install OEMService.apk
  adb shell am start -a android.intent.action.MAIN -n com.airwatch.admin.awoem.
[OEM_NAME]/com.airwatch.admin.awoem.PlatformOEMActivity -e hideui true
```

```
*If you are using POEM v3.2 or higher, use this intent instead:
  adb shell am start -a com.airwatch.START_AIRWATCH_SERVICE
```

```
adb install Agent.apk
  adb shell am start -a android.intent.action.MAIN -n com.airwatch.androidagent/
com.airwatch.agent.ui.activity.SplashActivity -e hideui true
  adb shell am broadcast -a com.airwatch.agent.action.IMPORT_CREDENTIAL_XML -e file /
sdcard/credentials.bin --user 0
  adb shell am broadcast -a com.airwatch.agent.action.AUTO_ENROLL --user 0
```

   b Save the file as autoenroll_OEM.bat in the same directory as the other files.

   *On Mac, it must be an SH file and run in Terminal.

4 Auto-enroll the device.

   a Connect the device to Wi-Fi.

   b Connect the device to the PC through an ADB connection.

   c Run the autoenroll_OEM.bat file.

Use the Sideload Staging Utility, WinRugg

You can easily sideload your Windows Rugged device through the AirWatch Sideload Staging Utility. The utility simplifies the process of installing a sideloading package onto the device with simple step-by-step instructions.

Procedure

1 Download the "VMware AirWatch Windows Mobile/CE Side loader Utility" from the myWorkspaceONE ™ documentation repository.

   The download URL is https://resources.workspaceone.com/view/dpzpbh2cvvnztcc9nkv3/en.

2 Install the utility after the download completes.

3 Start the AirWatch Side loader Utility after the installation finishes.

4 Connect the device to your computer.

The **Choose Side Staging** and **Stage File to Device** buttons become available for use after the utility detects the device connection.

5 Select **Choose Side Staging** and select the ZIP file you want to stage to a device.

6 Select **Stage File to Device** and begin staging the device.

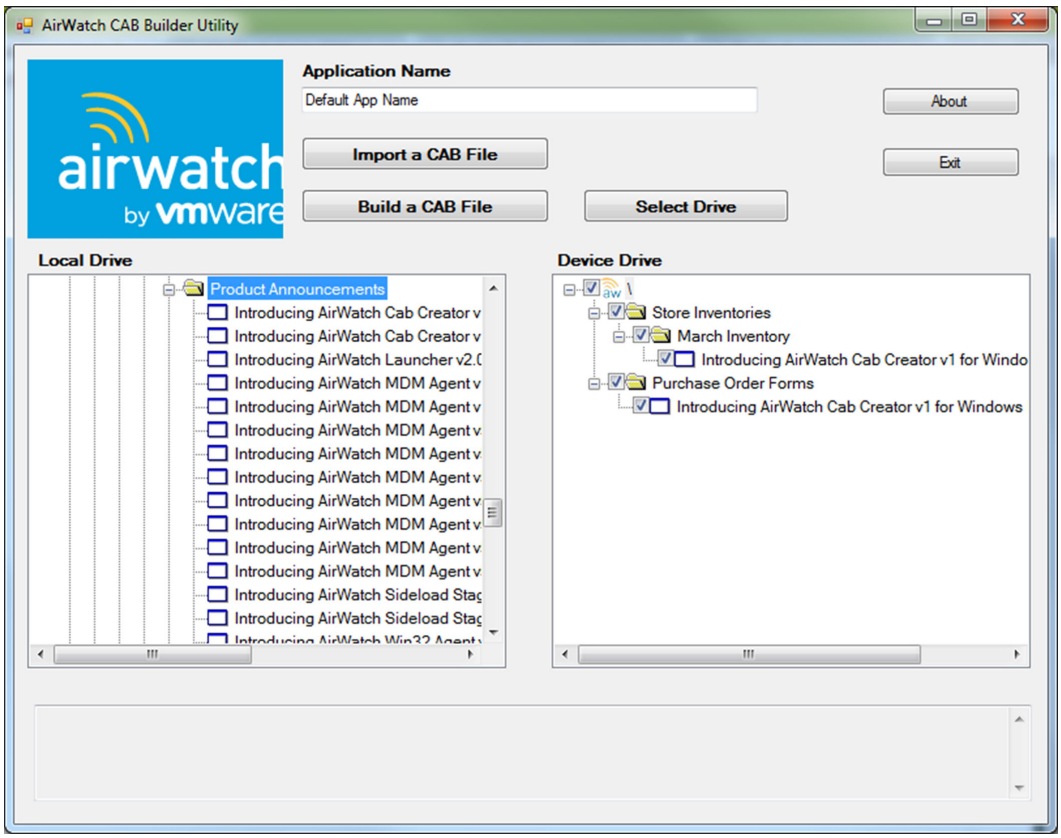
The ZIP file unzips, folders and directories are created, and the CAB file installs.

AirWatch CAB Creator, WinRugg

The AirWatch Cab Creator for Windows Rugged allows you to create custom CAB files for use on Windows Rugged devices in Workspace ONE UEM powered by AirWatch. These custom CAB files consist of files and applications you add from your computer.

Simplify the install process combining all the files and applications you want on your Windows Rugged device into a custom CAB file. You can also import other CAB files into your own *custom* CAB file.

This feature allows you to create one custom CAB file that contains all the CAB files you want to install on a device. You can also use the AirWatch Cab Creator to edit any existing CAB file on your PC. The AirWatch Cab Creator also supports importing files that are converted to the CAB file format upon saving.



Create a Custom CAB, WinRugg

Simplify installation of files onto your Windows Rugged devices by creating custom CAB files using the AirWatch Cab Creator for Windows Rugged. These custom CABs can contain your business files or the files necessary to upgrade your Windows Rugged devices.

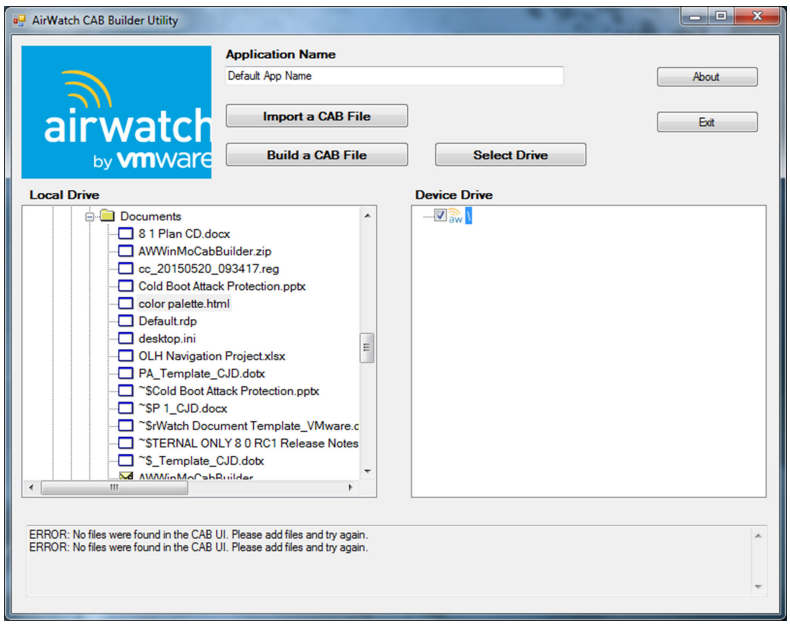
To use the AirWatch Cab Creator for Windows Rugged, you must meet the following requirements.

- A Windows device running Windows 7+
- .NET Framework 4.5

1 Download the "AirWatch Cab Creator" for Windows Rugged from the MyWorkspaceONE portal.

2 Unzip the file to your preferred directory.

3 Double-click CabBuilder.exe to start the app.



4 Navigate to a file on your **Local Drive** you want to add to the custom CAB file. You can select a different drive by selecting **Select Drive**.

5 Enter an **Application Name**. This text box is the name of the CAB file after installation.

Remember the name for use in Uninstall Manifest items.

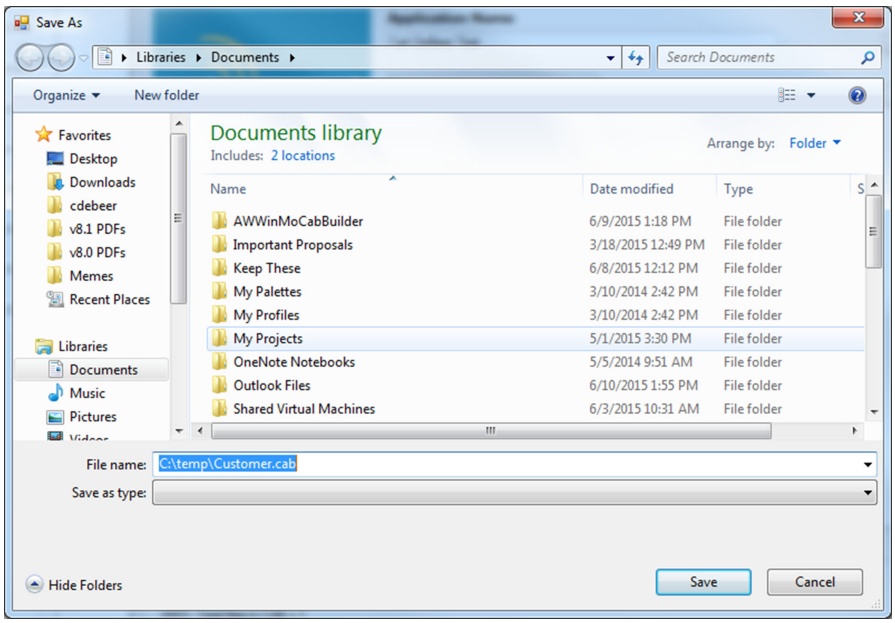
6 Select the file and drag it to the **Device Drive** pane.

To create a folder on the device drive, right-click the root drive and select **Add Folder**.

7 Repeat Step 5 for each file or application you want to add to the custom CAB file.

8 **Optional:** Add an existing CAB file to your custom CAB file by selecting **Import a CAB File**.

9 Select **Build a CAB File** to save the CAB file and select a name for the file.



10 Select **Save** to create a custom CAB file.

Products

4

Think of products as an ordered installation of device profiles, applications, and files/actions to be either published to the relay server or pushed to devices based on the conditions you create. Products are the main feature of the Product Provisioning system in Workspace ONE UEM.

Product Basics

Products are made of individual components which include Hub/Agent Packages, Applications, Conditions, Event Actions, File Servers, Files/Actions, and Profiles. Not every Product has each of these types of components. No matter how many or how few components your product has, each individual component must be created BEFORE you can make and provision a product.

Product Push to Device

After products are created and activated, they can be pushed to the device based on conditions you define. Conditions determine when a product is downloaded and when it is installed. Content in products is pushed to devices through optional relay servers or content delivery networks (CDN). Applications (Android only) provisioned by way of products can be optionally run through a tunnel.

Products are pushed to devices by way of smart group assignments. These groups control which devices get which product based on how the group is created, including software qualifiers like build version, currently installed apps, and hardware qualifiers like model number. You can also use Assignment Rules to target products to devices.

Note In addition, you can ensure the product you provision from the console or from an API call is the exact same product that gets received by the device. This product verification is built into the provisioning process. If the product status is Compliant, then the product on the device matches the product provisioned. If validation discovers a mismatch, the console pushes the content to the device again to ensure compliance between the product and the device. In this way, the product ensures that your devices remain up-to-date.

Product List View

Navigate to **Devices > Provisioning > Product List View** and see a full listing of all products in your environment.

You can download an **XLSX** or **CSV** (comma-separated values) file of the entire **Product List View**. You can then view and analyze this file with MS Excel. Select the **Export** button and choose a download location.

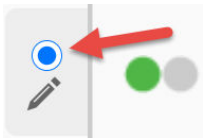
Publish Product to Relay Server

You can publish products to the relay server even if there are no devices in the organization group (OG) associated with it. Publishing content directly to the relay server is ideal for when you have a complex product and you want to stage the server before device enrollment.

1 Navigate to **Groups & Settings > All Settings > Admin > Product Provisioning** and select the **Enable** slider for **Queue Contents on Relay Servers without Assigned Devices**.

2 Move into the appropriate OG that is associated with the relay server onto which you intend to publish the product.

3 Navigate to **Devices > Provisioning > Product List View** and select the product you want to publish to the relay server. Do this by selecting the radio button above the edit icon.



4 Select the **Publish on Relay Servers** button. The **Organization Group** screen displays with a drop-down menu. The drop-down menu contains a listing of all child OGs and their parent, which you are currently in based on the move you made in step 2.

5 Select the specific OG which you intend to be the enrollment OG for devices. These devices are the future recipient of the product you are staging the server with.

6 Select the **Publish** button to publish the product to the relay server. If you select the parent OG in step 5, then any device that enrolls in any of its child OGs will receive this product upon enrollment. If you want to select individual child OGs, you must repeat steps 4 through 6 for each OG.

Active and inactive products remain on the relay server. Active products are pushed immediately to devices once they enroll in the OG selected in step 5.

You can check the relay server file listing for the product. See View Remote Files on Relay Server.

Prioritize Your Product With Expedited Deployment

You can give priority to selected products while uploading them to relay servers ahead of other products. This option moves these products to the "front of the line" when deploying to devices, ahead of other products, making it ideal for mission critical content.

Expedited Role Permission Required

Your administrator account must have a role assigned that includes the resource / permission called **Expedited Provisioning Policy Edit**.

For more information about applying this permission to a role and adding to or editing a role in an admin account, see View or Edit the Resources of an Admin Role and Assign a Role or Edit the Role Loadout of an Admin.

Expedited Product List View

You can identify expedited products in the Products List View by looking for the Expedited tag in the listing.

Product Verification

Workspace ONE UEM lets you ensure the product you provision from the console or from an API call is the exact same product that gets received by the device. This product verification is built into the product provisioning process.

Verification happens on the device Hub side but both the device end user and the administrator on the console side is made aware of the product's status.

You can review each product status by navigating to **Devices > Provisioning > Products List View**. The policy engine re-evaluates this compliance status whenever the device reports a change in the state of the applications, profiles, and other content on the device.

Product Persistence, Android, and WinRugg

Workspace ONE UEM lets you direct profiles, files-actions, and applications to remain on a device even *after* an enterprise reset. Content marked to persist is reinstalled after 1) an enterprise reset, 2) the Workspace ONE Intelligent Hub installs, and 3) the device restarts.

The help desk benefits the most from Product Persistence since it reinstalls required products and applications even after an enterprise reset.

Supported Devices

- Product Persistence for Windows Rugged only applies to Motorola, Honeywell, Psion, Pideon, and Intermec devices running Windows Mobile.
- Product Persistence for Android applies to the following.
  - Zebra devices (Android Legacy and Android Enterprise Work Managed)

    For details about enabling Workspace ONE Intelligent Hub persistence on your Zebra Android device with the correct APF file, see Upload the Workspace ONE Intelligent Hub APF File, Upgrade File-Action.
  - Honeywell devices (Android Legacy)
  - Motorola rugged devices (Android Legacy)

How Persistence Works

- A device must contain a staging configuration so that the Workspace ONE Intelligent Hub and enrollment reinstall following the enterprise reset. Staging configurations persist on a device.

- Set to persist any profiles, files-actions, or applications that you want to remain on the device after the enterprise reset.
- The device resets when the Enterprise Reset command is sent (see Chapter 5 Product Management). After resetting, the restore process begins.
- The Workspace ONE Intelligent Hub for the device reinstalls during the restore process.
- After the Workspace ONE Intelligent Hub is installed, any persisted profiles, such as Wi-Fi, reinstall.
- Any persisted files-actions or applications are reinstalled.

This chapter includes the following topics:

- Create a Product
- Configure a CDN for Provisioning
- Product Provisioning Profiles
- Application Provisioning, Android
- Product Conditions
- Event Actions, Android and WinRugg
- Files-Actions for Products
- Product Sets
- Custom Attributes
- What Happens If You Change a Product, Component, or Condition?

Create a Product

After creating the content you want to push to devices in Workspace ONE UEM, you can put all that content, including the apps, the files, and installation rules, in a product to be provisioned. Creation of the product also defines the order in which content is installed.

Prerequisites

To edit a product, the product must be deactivated in the list view first. Certain types of changes to products and their components require certain procedures. For more information, see What Happens If You Change a Product, Component, or Condition?

Procedure

1 Navigate to **Devices > Provisioning > Product List View > Add Product**.

2 Select the Platform you want to create a product for.

3 Complete the General text boxes.

Setting	Description
Name	Enter a name for the product. The name cannot be longer than 255 characters.
Description	Enter a short description for the product.
Managed By	Select the organization group that can edit the product.
Smart Groups	Enter the smart groups the product provisions. Smart groups are collections of devices that are built by identifying very specific device elements such as model, OS version, device tags, how the device was enrolled, how it's managed, and so forth. You can also build a smart group out of individual users and user groups.

4 Optionally select **Add Rules** and use **Assignment Rules** to control which devices receive the product.

Application rules can be applied to unmanaged applications installed on the device. Because the Workspace ONE UEM console does not manage ALL applications, these rules are available for certain system applications and third-party applications.

You are restricted from selecting **Assignment Rules** for organization groups of type **Partner** and **Global**.

Setting	Description
Add Rule	Select to create a rule for product provisioning. Displays the **Attribute/Application**, **Operator**, and **Value** drop-down menus.
Add Application Rule (Android Only)	**Android Only:** Select to create an application rule for product provisioning. This rule allows you to require applications to have specific versions install on the device for the rule to pass. Displays the **Attribute/Application**, **Operator**, and **Value** drop-down menus.
Add Logical Operator	Select to display a drop-down menu of logical operators such as AND, OR, NOT, and parentheses. Allows for more complex rules.
Attribute/ Application	The custom attribute or application used to designate which devices receive the product. Custom attributes are created separately. **Android Only:** Only internal applications display in the drop-down menu. You can use **Enter Manually** to enter the package ID of any application that must be present on the device. For more information, see Custom Attributes.

Setting	Description
Operator	This operator compares the **Attribute** to the **Value** to determine if the device qualifies for the product. **Note** When making an assignment rule, comparisons using the less than (<) and greater than (>) operators (and their variants) can only be used to compare numerical values including integers. The exception is when you are comparing OEM build versions, you can apply < and > operators on non-numerical ASCII strings. An example is when an OEM update filename includes hyphens, periods, and other characters together with numbers. Such assignment rules must identify a device manufacturer in the rule logic and that comparison is deemed accurate when the format on the device matches the one specified on the server.
Value	The value of the custom attribute. All values from all applicable devices are listed here for the **Attribute** selected for the rule.

5 Select **Save** and add the **Assignment Rule** to the product.

6 Select the **Manifest** tab.

7 Select **Add** and select the **Action(s) To Perform** for the **Manifest**.

Action(s) to Perform Drop-Down Menu	Settings
Install Profile	In the **Profile** text box, select the profile to install during the staging configuration. This component must be made before adding it to the manifest.
Uninstall Profile	In the **Profile** text box, select the profile to remove during the staging configuration.
Install Application (Android Only)	In the **Application** text box, select the App to install during the staging configuration. This component must be made before adding it to the manifest.
Uninstall Application (Android Only)	In the **Application** text box, select the App to remove during the staging configuration.
Install Files/Actions	In the **Files/Actions** text box, select the Files/Actions component to install during the staging configuration. This component must be made before adding it to the manifest.
Uninstall Files/ Actions	In the **Files/Actions** text box, select the Files/Actions component to remove during the staging configuration.
Reboot (Android Only)	**Reboots** the device during the staging configuration. This action works best as the last step of the manifest.
Warm Boot/Cold Boot (WinRugg Only)	**Warm Boot** reboots the device during the staging configuration. This action works best as the last step of the manifest. **Cold Boot** shuts down the device, forcing a restart by the end user. This action works best as the last step of the manifest.
Install Event Action (Android and WinRugg Only)	In the **Event Action** text box, select the Event Action component to install during the staging configuration. This component must be made before adding it to the manifest.
Uninstall Event Action (Android and WinRugg Only)	In the **Event Action** text box, select the Event Action component to remove during the staging configuration.

8 Optional step for **Android only** - Enable or disable **App Tunneling**. Application Tunneling lets you run applications through a virtual private network.

- In order for App Tunneling to function correctly, you must first create a VPN profile for your Android device.

  Create a VPN profile by navigating to **Devices > Provisioning > Components > Profiles**, select the **Add Profile** button, then select between the **Android** and **Android (Legacy)** platforms.

  For details about the platform-specific settings available for VPN profiles, see the **Platform Guides** for Android and Android Legacy, both available on https://docs.vmware.com/en/VMware-Workspace-ONE-UEM/index.html.

Platform-specific VPN profiles are supported when you enable the **App Tunneling** checkbox: you can select one VPN profile for **Android** and a separate VPN profile for **Android Legacy**.

9 **Android and Windows Rugged only** - Enable or disable **Persistent through enterprise reset**. Enable to keep the profile, application, files/actions, or event action on the device after an enterprise reset. For more information, see Product Persistence, Android, and WinRugg.

10 When finished with the single Manifest action, select **Save**.

11 Select **Add** again to add additional Manifest actions. You can adjust the order of manifest steps using the up and down arrows in the Manifest list view. You can also edit or delete a manifest step.

12 Select the **Conditions** tab if you want to use conditions with your product.

These conditions are optional and are not required to create and use a product.

13 Select **Add** and select either **Download Conditions**, **Install Conditions**, or both.

- A **Download Condition** determines when a product is downloaded but not installed on a device.
- An **Install Condition** determines when a product is installed on a device.

14 Select the **Deployment** tab if you want to control the time and date that products are activated and deactivated.

This tab is optional and is not required to create and use a product.

Setting	Description
Activation Date	Enter the time when a product automatically activates for device job processing. If the activation date is defined and the product is saved, the product stays inactive until the activation date is met according to the Workspace ONE UEM server time. The policy engine wakes up and automatically activates the product. You can manually activate products with activation dates beforehand. Manually activating a product overrides the activation date.
Deactivation Date	Enter the time when a product automatically deactivates from current and new device job processing. If the deactivation date is defined and the product is saved and currently active, it stays active until the deactivation date is met according to the Workspace ONE UEM server time. The policy engine wakes up and automatically deactivates the product. You can manually deactivate products with deactivation dates beforehand. Manually deactivating a product overrides the deactivation date. A deactivation date cannot be set earlier than the activation date.
Pause/Resume	Enable to ensure that an interrupted product provisioning due to Wi-Fi connectivity issues is retried in the following manner. If you use a relay server, the Workspace ONE Intelligent Hub makes 5 attempts to contact the relay server to download the product. If the download fails or Wi-Fi connectivity fails, then the Hub makes 5 attempts to connect to the Device Services URL by way of HTTPS. On-premises admins can configure this setting at **Groups & Settings > All Settings > Admin > Content Delivery Settings > File Sources Include HTTPS**. This HTTPS setting is enabled by default in SaaS environments. If the product is still unavailable, then the job is moved to a Paused state. After a while, the Hub changes the job state to Started and makes another 5 attempts to connect to the relay server and another 5 attempts on the Device Services (provided the option is enabled). It repeats this process for a maximum 24 hours after the first connection failure. If, after this 24-hour period, the download is still not complete, the job is set to Failed.
Product Type	Determine if a product is **Required** or **Elective**. A required product provisions to assigned devices when deployment settings are met. An elective product is only provisioned when it is manually pushed on the Device Details View of a provisioned device.
Deployment Mode	Select from the following how the product is to be deployed. **Relay Server with Workspace ONE Server as Backup** – This is the default deployment mode. The device attempts to receive the product from the relay server initially, making 5 separate attempts, then falling back to device services as a secondary source. **Relay Server Only** – The device attempts to receive the product from the relay server only. In a scenario where the relay server is not configured or deactivated, the fallback source is device services. If multiple relay servers are assigned to a device, the Workspace ONE Intelligent Hub will attempt to download the product 5 times from each relay server.
Expedite Deployment	Enable this check box to give priority to this product. The **Expedite Deployment** check box is editable only when the product is inactive. For more information, see Prioritize Your Product With Expedited Deployment.

15 Select the **Dependencies** tab if you want to set the product to only provision devices that have other products provisioned as well.

   a Select **Add** and add a dependent product.

   You can add as many dependent products as you want.

16 Select to deploy the product immediately by selecting **Activate** or wait to deploy later and select **Save**.

Configure a CDN for Provisioning

Workspace ONE UEM lets you configure a Content Delivery Network to help lighten the distribution of provisioned product content by offloading traffic from your network.

As the administrator of an on-premises environment, you have access to all the resources presented in each of the steps documented here. Administrators of dedicated SaaS and shared SaaS environments, however, may not have access to the same resources.

Setting up Akamai CDN to integrate with Workspace ONE UEM

You must configure Akamai CDN for use in a Workspace ONE UEM production environment.

To learn more about configuring Akamai integration, see Akamai product documentation at https://www.akamai.com.

To configure Akamai to integrate with Workspace ONE UEM, complete the following settings:

1 At the time of configuration, select **AirWatch** as your client name.

2 After you set up properties that control Akamai's edge server traffic, add behaviors to the property as per your requirements. Currently, Workspace ONE UEM requires you to configure the following two behaviors:

   a **Edge Server Identification**: Include a known cookie value that can be verified at the origin server before serving requests back to the edge server.

Setting	Description
	AW-AUTH-KEY.
Cookie Value	Use a hash generator to create the hash key generated value. CDN server uses the key to connect to the origin server. Retain a copy of the hash key for use while installing the origin server.
Cookie Domain	Enter the Origin Server URL. For example, enter origin.acme.com.

   b **Advanced Override**: Use the Advanced Override option to specify the parameter to use for tokens that are passed to the URL. Also, specify the expected shared-secret/salt that is used to generate the HMAC token when validating the file requests to the edge server. Advanced Override is only available by request from Akamai Support, and requires an extra fee. This feature is required to enter the Token key in the Console configuration.

Configure your Origin Server to integrate Workspace ONE UEM with Akamai CDN

An origin server is the physical location from which content is retrieved. It is required in all configurations that retrieve content from an origin. You can set up the Origin Server to integrate Akamai CDN with Workspace ONE UEM.

To set up the Origin Server, complete the following steps:

1 Install the Web Server Role (IIS). he Internet. It is possible to set up the DNS to do routing internally to the proper servers as necessary. For storage, multiply the average file size by the average number of files, then multiply by two to avoid full disk issues that prevent the caching of files.

   Enable the following features:

   a Request Filtering

   b Window Authentication

   c URL Authorization

   d IP and Domain Restrictions

2 Install URL Rewrite IIS from the Microsoft website.

3 Add the following extensions to **Default Website MIME Types**.

Extension	Content Type
.app	application/vnd.android.package-archive
.appx	application/vns.ms-appx
.appxbundle	application/octet-stream
.ipa	application/octet-stream
.lic	text/plain (For BSP)
.msi*	.msi* application/octet-stream
.msp	application/octet-stream
.mst	application/octet-stream
.pkg	application/octet-stream
.xap*	application/x-silverlight-app
.xbap*	application/x-ms-xbap
.ppkg	application/octet-stream
.dmg	application/octet-stream
.mpkg	application/octet-stream

Extension	Content Type
.plist	text/xml
.apk	application/vnd.android.package-archive

Note MIME Types already exist in Windows 2012 R2.

4 Navigate to the CDN content storage location.

5 Create a shared folder named **CDN**. The folder that is configured for the web server must be mapped to a file with both read, write permission that is available to the Workspace ONE UEM console and Device Services.

6 In the **CDN** folder, create a file named **monitor.txt**. Enter some random text into the document so that you can validate the connection at a later stage.

7 Set up the user account credentials for accessing the CDN using a UNC/SMB path. The UNC/SMB path is used during the configuration of the UEM console. The user name and password are used for connecting to the **UNC/SMB** folder and are also entered into the UEM console.

8 Configure the security setup for accessing the folder from the IIS website.

   a Add the application pool user account to the **CDN**folder of the shared drive.

   b Add the following usernts:

      1 ISUR (All but Full control)

      2 IIS_IUSRS (All but Full control)

      3 NetworkService (Full Control)

      4 UNC/SMB Service Account (All but Full control)

9 Under **Application Pools**, right-click **DefaultAppPool** and select **Advanced Settings**. Set the App Pool Identity to **NetworkService**.

10 Right-click **Default Website**, select **Manage Website**, and select **Advanced Settings**.

11 Change the **Physical Path** to the configured drive for the CDN content.

12 After Akamai is configured, you can set up the request filtering for the cookie that is used for authentication of the URL.

   a Obtain the CDN Configuration Tool installer.

   b Run the CDN installation and enter the secret key (SHA256 Hash Key) that is configured with your Akamai account for Edge Server Identification.

13 Make a note of the **Network Path** for the UEM console configuration.

Configure Akamai CDN in the Workspace ONE UEM Console

You can configure Akamai CDN in Workspace ONE UEM console. During the configuration, the values that you enter in the configuration page can be retrieved by logging in to your CDN provider portal and locating the values. If you are an on-premises customer who requires additional assistance, contact Workspace ONE UEM Support.

- **Current Setting** – Select whether to **Inherit** or **Override** the displayed settings. Inherit means use the settings of the parent OG if the current organization group, Override enables the settings for editing so you can modify the current OG settings directly.

Complete the Akamai configuration in the UEM console:

1 In the UEM console, ensure that you are in the Global OG.

2 Navigate to **Groups & Settings > All Settings > System > Enterprise Integration > CDN**.

3 Complete the Akamai configuration settings:

Setting	Description
Enabled	Select **Enabled** to route all the application downloads through the CDN for all the devices that are the managed at the current organization group. Select **Disabled** to route all the application downloads through Workspace ONE UEM server.
Directory	Enter the server name and the directory. The Directory name is the **Network Path** that is used while configuring the origin server.
User name	Enter a dedicated Service Account user name that is placed on the Origin Server side.
Password	Enter the dedicated Service Account password that is placed on the Origin Server side.
Content Server	Enter the DNS of the CNAME that is as per the data center (for example, CDN.acme.com).
Token Parameter	For Akamai, it is the token as per the Advanced Override.
Salt Value	Enter the token that your CDN provides. For Akamai, it is done by enabling the Advanced Override code.
Destination	Enter the destination name of the CDN.

Child Permission – Select the available behavior of child organization groups that exist below the currently selected organization group. Inherit only means child OGs are only allowed to inherit these settings. Override only means they override the settings, and Inherit or Override means you can choose to inherit or override settings in child OGs that exist below the currently selected OG.

Test Connection – Select this button to test the connection between Workspace ONE UEM and Akamai. A status message is displayed to confirm the connection.

Disable CDN System Settings for the Child Organization Group

If the child organization group has devices with IP whitelisting and restricts the application downloads to be routed through the CDN, you can disable the CDN routing. To disable CDN System Settings for the Child Organization Group, navigate to the child organization, select **Override**, and disable **Allow App downloads through CDN**. u choose to override the settings, you can only disable **Allow App downloads through CDN**. However, the system restricts you from editing the **Child Permission**.

Validate Your Workspace ONE UEM Integration with Akamai CDN

Complete the following steps to validate Workspace ONE UEM integration with CDN:

1 In a web browser, navigate to your CDN DNS. For example, CDN.acme.com/monitor.txt), which results in an error. The reason is because the connection to the Origin server from the CDN requires authentication.

2 In a web browser, navigate to your Origin DNS. For example, origin.acme.com/monitor.txt), which succeeds. Accessing the origin server directly only works for the **monitor.txt** file, which is used to validate the connection.

Enable Product Downloads Through CDN

Once you have configured the Akamai and Origin server integrations with Workspace ONE UEM, you must take the last step and enable product downloads to work with a CDN.

1 While logged into the Workspace ONE UEM console, make sure you are in the Global organization group.

2 Navigate to **Groups & Settings > All Settings > Admin > Product Provisioning**.

3 **Enable** the option **Allow Product Downloads Through CDN**.

Product Provisioning Profiles

Workspace ONE UEM powered by AirWatch allows you to create profiles for your rugged devices. The profiles created for rugged devices are installed or uninstalled as part of a product. Unlike device profiles created for MDM deployments, product provisioning profiles have different rules governing editing and deleting.

Profiles created under Products are different than those created through Workspace ONE UEM. This section lists the differences between profiles created for normal device use and those created for use in product provisioning.

Profile Creation and General Settings

Profiles for use with product provisioning must be created by navigating to **Devices > Provisioning > Components > Profiles** and select **Add Profile**.

While creating these product provisioning profiles, the general tab operates differently than the regular general tab for profiles.

Note Assignment of profiles happens at the product level and not at the profile level as it is in MDM (non product) profiles.

The **Profile Scope** option under the **General** tab determines whether the profile is available for products. Only profiles whose Profile Scope is **Production** or **Both** can be pushed through products. **Staging** profiles are meant to stage devices only.

Saving Product Provisioning Profiles

After configuring your product provisioning profile, select **Save** instead of **Save & Publish**.

Profiles names cannot be longer than 255 characters.

Update Profiles

When you edit an existing profile, the version number increases. After saving the edits, Workspace ONE UEM runs a check on all active products to find any that contain the newly edited profile.

If any active products contain the profile, a warning prompt displays listing all active products affected by the edited profile. You can then select to **Activate** or **Deactivate** a product using the profile.

Delete a Product Provisioning Profile

You can delete an unwanted or unused product provisioning profile provided it is not attached to a product. Workspace ONE UEM monitors attempts to delete a profile against the list of active products.

Before you can delete a product provisioning profile, you must detach it from all products, either by deleting the product or removing the association with the profile.

1 Navigate to **Devices > Provisioning > Product List View** and select the product you want to detach or delete. The **View Product** screen displays containing a read-only view of the product specifications.

2 Select the **Edit** button. The **Edit Product** screen displays containing an editable view of the product.

3 Select the **Manifest** tab and remove any Install Profile or Uninstall Profile steps by selecting the delete button (✖) to the far-right of the listing.

4 Select **Save** or **Activate** to leave the product deactivated or activated, respectively.

Application Provisioning, Android

Product provisioning in Workspace ONE UEM console allows you to upload applications to the console for distribution as part of a product. Through product provisioning, you can upgrade and remove applications remotely. You can also apply a VPN configuration to an application.

Silent install of applications is supported on Android Work Managed devices and on Android Legacy devices with an OEM service application.

For more information on configuring a per App VPN, see Create a Product.

Note Smart group assignment happens on the Product level and not on the Application level.

Upload an Application to Provision

Applications added through product provisioning use the rules and restrictions of the product to manage the installation. Add applications that you want installed onto devices as part of a product.

1 Navigate to **Devices > Provisioning > Components > Applications** and select **Add Application**.

2 Enter or select the Organization Group the application is **Managed By**.

3 Select the **Upload** button and browse for the **Application File**.

4 Select the **Choose File** button and browse your device for a local application file.

5 Select **Save** and finish uploading the application.

The uploader reviews the version of the application you have selected and compares it to the applications that have already been uploaded in UEM. If it finds a match, it prevents you from uploading it again.

6 Select **Continue** and add the application to the Product Provisioning application list.

Add New Application Versions to Provision

You can add a version of an already uploaded application. This action enables you to push the newest version of an application to end users using the existing products you have already created.

1 Navigate to **Devices > Provisioning > Components > Applications** and select **More**.

2 Select the **Add Version** option from the drop-down menu.

3 Upload the new version of the application as described above in the section entitled **Upload an Application to Provision**.

4 Select **Save**.

5 Navigate to **Devices > Provisioning > Product List View** and find the product that contains the old version of the application you want to update. If necessary, use the filters to narrow your search.

6 Select the radio button to the left of the product name. This radio button selection displays some action buttons at the top of the **List View**.

7 Select the **Edit** action button. The **Edit product** screen displays.

8 In the **Manifest** tab, find the **Install Application** Action Type that contains the application you want to update and select the small blue pencil icon to the right of the **Description** column. The **Edit Manifest** screen displays.

9 In the **Application** text box, delete the application name. This action causes the drop-down menu to appear which now displays all versions of all applications in your entire Applications library.

10 Select the new version of the application that you uploaded earlier.

11 Select the **Save** button. The **Edit product** screen now shows the **Manifest** that includes the new version of the application.

12 Select the **Activate** button. This action pushes the new version of the application to the devices provisioned with this product.

Result: By taking these steps, you avoid having two versions of the same app assigned to the same device. The risk is that new device enrollments will have both versions assigned to it at first.

Delete Applications from Provisioning

Remove unwanted applications from your products. The Workspace ONE UEM console monitors any attempt to delete an application against the list of active products.

Before you can delete an application, it must be detached from all products.

1 Navigate to **Devices > Provisioning > Applications**.

2 Locate the Application you want to delete from the List View.

3 Select the down arrow button to the far-right of the row belonging to the application.

4 Select **Delete**.

5 The **Restricted Action** screen display. Restricted Action is a security feature designed to prevent accidental application deletions. Enter your four-digit security PIN and proceed with the deletion.

6 If the application you selected is part of a product, then you are not allowed to delete it until the application is detached from the product. A green warning prompt appears. At the bottom of the warning prompt, lists all the products to which the selected application belongs. Take note of these products.

7 Navigate to **Devices > Provisioning > Product List View**.

8 Select the **Product** listed in the warning prompt.

9 Select the **Edit** button. The Edit Product screen displays.

10 Switch to the **Manifest** tab.

11 Select the delete button (**x**) and detach the application from the product. You are prompted to confirm.

12 Select **Save**.

13 Repeat steps 8–12 for all products containing the application.

What to do next: After the application detaches from all products, you can delete the application by returning to complete steps 1–5.

Product Conditions

Product provisioning in Workspace ONE UEM powered by AirWatch allows for conditions to be placed on products. A condition determines when the product or OS upgrade package can be downloaded and installed. Conditions are reviewed when a product is pushed to a device.

Your device fleet is not always readily available for maintenance. You might have devices in different time zones or countries. Some devices may be away from a reliable hotspot. Also, you cannot always ensure that a device is not in use when you push a product. For all these reasons and many more, you can use conditions to delay the download and installation of products.

These conditions defer the product download or installation until the device meets the criteria of the assigned condition. You can set products to download only based on battery life, power adapters, user confirmation, and other criteria. The available conditions for your products vary based on the device platform.

When you edit a condition, be aware that the changes can filter down to active Products or Event-Actions that are a part of active Products. For more information, see What Happens If You Change a Product, Component, or Condition?

Create a Condition

You can set products to download and install on your device only when certain conditions are met. Create a condition and determine when a product downloads and installs onto your devices.

1 Navigate to **Devices > Provisioning > Components > Conditions** and select the **Add Condition** button.

2 Select the Platform you want to create a condition for.

3 Complete the **Create Condition** Type settings.

Settings	Description
Name	Enter a name for the condition. The name cannot be longer than 255 characters.
Description	Enter a description for the condition.
Condition	Select from the drop-down menu of platform-specific conditions. The type of condition affects the parameters on the **Details** tab to follow. For more information about the conditions themselves, see the section on this page entitled **Platform-Specific Conditions**.
Managed By	Select the organization group that manages the condition.

4 Select **Next**.

5 Complete the **Details** settings based on the condition type selected.

6 Select **Finish**.

Copy or Delete Conditions in List View

You can view all conditions in a list view. You can also edit and delete conditions from the list view.

Before you can delete a condition, you might have to detach it from one or more products. For more information, see the section on this page entitled **Detach a Condition from a Product**.

1 Navigate to **Devices > Provisioning > Components > Conditions**.

2 Select the pencil icon ( ) to the left of the name of the condition and open the **Edit Condition** screen.

3 Select the radio button to the far left of the condition and display the **Copy** and **Delete** buttons, offering more actions.

Detach a Condition from a Product

Remove unwanted conditions from your product. The Workspace ONE UEM console checks any attempt to delete a condition against the list of active products.

To delete a condition, it must be detached from all products as detailed below.

1 Select the **Product** listed in the Warning prompt.

2 Select **Edit**.

3 Remove the condition from the product.

4 Select **Save**.

5 Repeat the steps above for all products containing the condition.

6 Once the condition detaches from all products, you can delete the condition. For more information, see the section on this page entitled **Copy or Delete Conditions in List View**.

Results: If a condition is part of an active product, a warning prompt appears listing any product that uses the condition.

Platform-Specific Conditions

Select a condition to be taken to its definition. The File condition has two versions, one for Windows Rugged and QNX, the other for Android. Select the check marks for File to be taken to the platform-specific definition.

				Windows 7	Windows
Condition	Android	macOS	QNX	& Desktop	Rugged
Adapter Condition					✓

Adapter Time Condition	✓	✓	✓	✓
Battery Threshold Condition	✓			✓
Confirm Condition	✓	✓		✓
Connectivity State Condition				✓
File --->	File (2) Android Condition	File (1) QNX and WinRugg Condition		File (1) QNX and WinRugg Condition
Launcher	✓			
Memory Threshold Condition				✓
Power Condition	✓		✓	✓
Recurring Schedule Condition	✓			
Schedule Condition	✓		✓	
SD Encryption Condition	✓		✓	
Time Condition	✓	✓		✓

Adapter Condition

Supported Platform: Windows Rugged – This condition type tests to see which, if any, **Network Adapters** are connected. This condition can be relevant if network connectivity is a scarce or expensive resource and certain operations are limited to use over certain **Network Adapters** or prohibited from use over certain **Network Adapters**.

Settings	Descriptions
What would you like to do with the adapters?	Select to use the adapters you define or exclude them. ■ Only use these adapters to connect. ■ Exclude these adapters from connecting.
Adapter 1	Select to select an adapter from the list or enter the adapter by name. ■ Select From List. ■ Enter name.
Select adapter.	Select the network adapter from the drop-down menu.
Specify Adapter.	Use or exclude the adapter you enter here.
Adapter 2/Adapter 3	Use or exclude additional adapters you enter here.

Adapter Time Condition

Supported Platforms: Android, macOS, All Windows – This condition type tests for various combinations of constraints related to **Network Adapters** including local date, time, and frequency on the device.

Settings	Description
Specify scenario #1?	Set to **Specify this scenario** to begin configuring the condition scenario. Up to 5 scenarios can be entered, each with their own constraint choices. Each Scenario is an OR statement and each option inside a Scenario is an AND statement. For example, a device monitors whether Scenario #1 OR Scenario #2 is true. If Scenario #1 is true, it reviews if all the constraints listed are true because they are AND statements.
Scenario description	Enter a description for the adapter time scenario.
Constrain Network Adapters?.	Set to **Constrain based on the Best Connected Network Adapter** and configure the following. ■ Specify any **Included or Excluded Network Adapters**. ■ Either **Select Network Adapter Class from a drop-down list** or **Type in a Network Adapter Name**. ■ Up to five network adapters can be selected in the **Adapter selection method?** setting. ■ For each adapter you want to include/exclude, **Select a Network Adapter Class** drop-down menu and entering a specific **Adapter name**. If you want to skip this kind of constraint, then select **Don't constrain based on the Best Connected Network Adapter**. Then you can proceed with defining another kind of constraint.
Constrain days of week?.	For each day of the week, select whether it is included or excluded.
Constrain months?.	For each month, select whether it is included or excluded.
Constrain days of month?.	Enter a **Start day of month?** and an **End day of month?**.
Constrain years?.	Enter a **Start year?** and an **Last year?**.
Constrain time of day?.	Enter the **Start hour?**, **Start minute?**, **End hour?**, and **End minute?**.
Set frequency limit?.	Ranges from **Every 15 Minutes** to **Every 1 Week**.
Set frequency limit?.	Ranges from **Every 15 Minutes** to **Every 1 Week**. **Set frequency limit** is a mandatory setting. The Adapter Time condition will not function correctly without it.

Note ActiveSync and VPN Network Adapters are not supported under the Android platform.

Battery Threshold Condition

Supported Platforms: Android and Windows Rugged – This condition type tests the device to see what level battery charge remains. You can test for charge levels under a defined threshold or over a defined threshold.

Settings	Description
Battery Level	Select between **Less than or Equal To**, **Greater Than or Equal To**, and **Between** to define a range of charge levels.
Battery Percentage	Enter a percentage between 1 and 100. When **Between** is selected, you must enter a range comprised of two percentage levels.

Connectivity State Condition

Supported Platform: Windows Rugged – This condition tests the device for the type of connection and the length of time it has been connected.

Settings	Description
the device is connected to.	Select between **Wi-Fi** and **Cellular**.
the device is continuously.	Two selections must be made. First, select whether to test if the device has been Connected or Disconnected. Next, select the length of time it has been in such a state.

Confirm Condition

Supported Platforms: Android, macOS, Windows Rugged – This condition type prompts the end user to determine whether the condition is met. This prompt is customizable so you can control what displays on the prompt.

Table 4-1. Message to Be Displayed (Confirm)

Settings	Description
First line prompt	Enter a header of the prompt.
Second line prompt	Enter the body of the prompt. Enter the subheading of the prompt (macOS Only).

Table 4-1. Message to Be Displayed (Confirm) (continued)

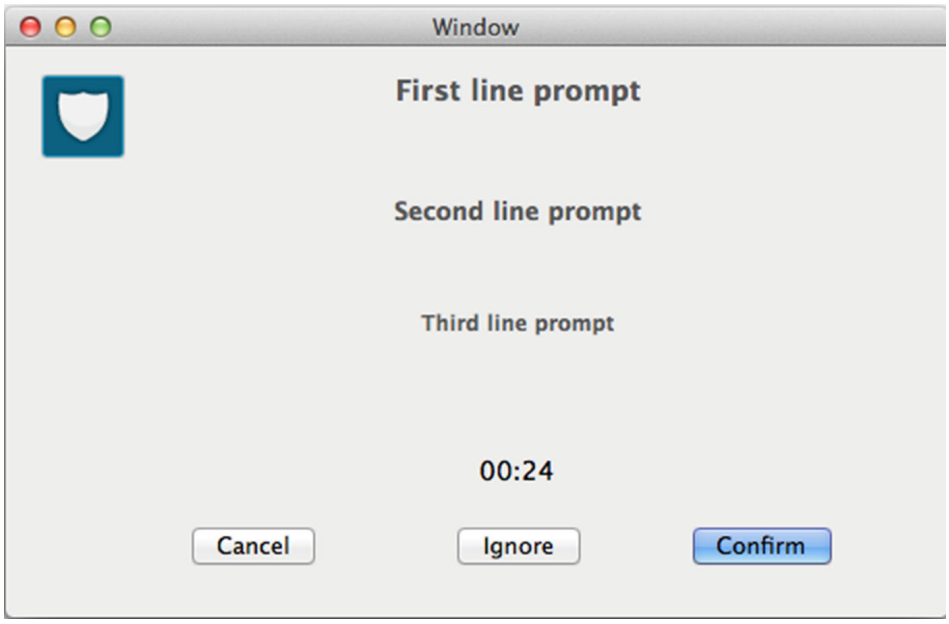
Settings	Description
Third line prompt	If you enable a countdown, you can enter a countdown phrase into this setting. For example, "You have %count% seconds to comply" where %count% is the countdown. Enter the body of the prompt into the Third Line Prompt (Mac OSX Only). 
Allow users to cancel actions?	Select **Yes** if you want to give users a chance to opt out of the action upon which this condition is placed. Select **No** and obligate users to accept the action.

Table 4-2. Delay (Confirm)

Settings	Description
Delay (seconds)	Delay for a specified time or until the end user makes a selection. If you enter a non-zero value, the prompt waits for that value worth of seconds. Then if the end user does not make a selection in the time allowed, the condition is automatically considered not met. If a value of zero is entered, then the prompt displays indefinitely until the user makes a selection.
Enable countdown?	Select **Yes** and allow the delay time to be “counted” down on the device so the end user knows how much time is remaining to make a selection. Select **No** and hide the delay countdown.

Table 4-3. Defer Action (Confirm)

Settings	Description
Defer time.	The minimum time after the condition is not met before the end user is prompted again to determine the state of this condition. If a non-zero value is entered, the end user is not prompted again for at least this number of seconds. If a value of zero is entered, then the end user might be prompted again when the next execution of the Check-In command.
Maximum number of defers	The maximum number of times the condition is not met. After the condition is not met this number of times, it is either met or failed, depending on the setting of the next feature. If a value of zero is entered, then the condition is met or failed on the first time.
Action after maximum defers.	Select the action to trigger after the maximum number of defers is met. ■ **Fail Condition**. ■ **Display Cancel Button**. ■ **Pass Condition**.

File (1) QNX and WinRugg Condition

Supported Platforms: QNX and Windows Rugged – This condition type tests the device to see if a file is present or not. You can set the condition to test if the file is on the device, and if it is, test true. You can also set the condition to test true if a file is not on the device. Finally, you can set the condition to create a flag file and save it on the device for use with third-party programs.

Settings	Description
Specify Flag File	Set to Specify Flag File to enable the use of a flag file for use with third-party programs.
Specify Flag File Location	Enter a location path for the flag file.
Specify Flag File Contents	Enter the content you want in the flag file.
Specify Scenario #1	Set to **Specify this scenario** to enable testing a file's existence on the device.
Test File	Enter the name of the test file the condition tests.
Test Type	Test whether the **File Exists** or the **File Does Not Exist**.
Remove File	Either **Remove File** or **Do Not Remove File** when the test finishes.

File (2) Android Condition

Supported Platform: Android – This iteration of the file condition is specific to Android, but it still tests the device to see if a file is present or not. You can set the condition to test if the file is on the device, and if it is, test true. Alternatively, if a file is not on the device, you can also set the condition to test true.

Settings	Description
File Name	Enter the name of the file the system searches for, including device path. The system searches for this file before it downloads or installs product components.
Condition Met When	Select whether the system downloads or installs based on the existence of the file (File Found) or nonexistence of the file (File Not Found).
Frequency	Select how often the system searches for this file.
Duration	Select how long this condition performs the action.
After Duration Exceeded	Select how to proceed after the condition duration period elapses.

Launcher

Supported Platform: Android – This condition type tests the device whether it is checked in or checked out.

Settings	Description
Launcher	**Check In** – Action will be taken on the device only if Launcher is checked in. **Check Out** – Action will be taken on the device only if Launcher is checked out.

Memory Threshold Condition

Supported Platform: Windows Rugged – This condition type tests the device for the level of system memory that is available.

Settings	Description
Available memory is less than	Enter a percentage of available memory such that your action only runs if the device has less than the indicated amount.

Power Condition

Supported Platforms: Android and All Windows – This condition type tests how a device is being powered, including whether the device is plugged in or has a full battery level. Use a **Power** condition type to prompt users to place the device into the cradle or to insert a charged replacement battery. If your testing needs are particular, the **Battery Threshold** condition (Android and Windows Rugged only) offers more granular battery tests than Power offers.

Table 4-4. Message to Be Displayed (Power)

Settings	Description
First line prompt	Enter a header for the prompt.
Second line prompt	Enter the body of the prompt.
Third line prompt	If you enable a countdown, you can enter a countdown phrase into the **Third line prompt** text box. For example, "You have %count% seconds to comply" where %count% is the countdown clock.

Table 4-5. Condition (Power)

Settings	Description
Required power level	Enter the required power level for the condition to test true. ■ **A/C**. ■ **A/C or Full Battery**.

Table 4-6. Delay (Power)

Settings	Description
Delay (seconds).	Use this to delay for a specified time or until the end user makes a selection. If you enter a non-zero value, the prompt waits for that value worth of seconds. If the end user does not make a selection in the time allowed, the condition is automatically considered not met. If a value of zero is entered, then the prompt displays indefinitely until the end user makes a selection.
Enable countdown?	This allows delay time to be “counted” down on the device so the end user knows how much time is remaining for the user to make a selection.

Recurring Schedule Condition

Supported Platform: Android – Rather than set an event action to begin based on a specific device condition, you can simply assign it to a recurring schedule.

Settings	Description
Scheduled Interval	Select from **Hourly**, **Daily**, and **Weekly** as the basis for the recurrence.
Number of hours / days / weeks between events	Enter the number of hours, days, or weeks between events. For example, if you want an event action to run every other Wednesday, then select Weekly, enter 2 in this text box, then select Wednesday as the **Day of event occurrence**. After the product with this condition is activated, the recurring schedule begins as soon as is administratively possible. Continuing the above example, the recurring schedule begins the **first** Wednesday after you activate the product that uses this condition.

Table 4-7. Days of Event Occurrence (Weekly Only), Recurring Schedule

Settings	Description
Sunday through **Saturday**	Select **Yes** for each day you want the event action to start.

Table 4-8. Start and End Time (Daily and Weekly Only), Recurring Schedule

Settings	Description
Start Time.	Enter the time of day the event action starts.
End Time.	Many things can delay an event action from starting such as the device being offline or a previous event still in progress. When the device is ready, you can allow the event action to always start or set an end time at which point the device waits until the next start time. Enter the time of day the event action stops making start attempts until the next scheduled start time.
Disable end time.	Select **Yes** to disable end time. This option causes the device to make continued attempts to start the event action until it succeeds.

Schedule Condition

Supported Platforms: Android, Windows 7, and Windows Desktop – This condition type tests the device date and time against a specific date/time entered. When the date/time is met, the condition passes and allows the download.

Settings	Description
Date	Select the specific date from the drop-down calendar.
Time	Select the specific hour and minute from the drop-down menu.

SD Encryption Condition

Supported Platforms: Android, Windows 7, and Windows Desktop – This condition type tests whether the device's SD card is encrypted or not encrypted. This condition can be relevant if you must wait for the SD card to be encrypted before downloading a file.

Settings	Description
SD card is	Select **Encrypted** or **Unencrypted** and limit the product based on the state of the SD card encryption.

Time Condition

Supported Platforms: Android, QNX, and Windows Rugged – This condition type tests the local date and time on a device.

Table 4-9. First Time Slot (Time)

Settings	Description
Start **Date** **Time**	Select **Month**, **Day**, **Year** and **Hour** & **Minute**for Start.
Finish **Date** **Time**	Select **Month**, **Day**, **Year** and **Hour** & **Minute** for Finish.

Table 4-10. Second Time Slot (Time)

Settings	Description
Enable time check 2?.	Select **Yes** to display a second set of options identical to the First Time Slot.

Table 4-11. Third Time Slot (Time)

Settings	Description
Enable time check 3?.	Select **Yes** to display a third set of options identical to the First Time Slot.

Event Actions, Android and WinRugg

In product provisioning for Workspace ONE UEM, there are Event Actions, or rules you make to trigger an action when an event occurs. The 'event' occurs followed by the prescribed 'action'. The Event Actions wizard guides you through creating the conditions and actions together.

In cases where you want to perform a device action only when certain conditions are met, event actions allow you to control the timing of these actions. For example, your devices might need new files download to them but only until the device is not in use. A device event can wait until the device is connected to its charger before installing files. In another example, you can set a connectivity condition to wait for the device to connect to Wi-Fi before sending in a device check-in.

Event actions act as a device-based "if-this-then-that" configuration which controls the recurrence of actions on a device. A product only processes once on a device. Event actions, however, process any time the conditions are met.

Push event actions to devices as a component of a product. When you edit an Event-Action, the changes can filter down to Active Products. For more information, see What Happens If You Change a Product, Component, or Condition?

Create an Event Action, Android and WinRugg

You can create event actions that run on a device when certain conditions are met.

1 Navigate to **Devices > Provisioning > Components > Event Actions** and select the **Add Event Action** button.

The **Add Event Action** wizard displays.

2 Select your device platform.

The available conditions and available actions for the platform display.

3 Select **Next**.

4 Complete the **Details** settings and select **Next** when complete.

Settings	Descriptions
Name	Enter a name for the event action. The name cannot be longer than 255 characters.
Description	Enter a short description for the event action.
Managed By	Select the organization group that can edit the event action.

5 Select a **Condition** that triggers the device action.

You can select a previously created condition from the drop-down menu or create a new one.

a To create a condition, select **Create Condition** from the drop-down menu.

b Select **Next** when complete.

Table 4-12. Event-Action Conditions by Platform

Condition	Android	WinRugg	Description
AC Power	✓		Detects when the device is connected to an AC adapter.
Adapter		✓	Detects network adapters, enabling you to select which to allow/disallow.
Battery Threshold	✓	✓	Detects battery charge, enabling you to direct the device as it reaches prescribed battery levels.
Connectivity		✓	Detects connectivity, enabling you to take action when certain requirements are met.
Launcher	✓		Detects when Launcher on the device is checked in or checked out.
Memory		✓	Detects memory levels, enabling you to take actions when it reaches certain levels.
Recurring Schedule	✓		Enables you to schedule your actions on a recurring basis.
Time		✓	Enables you to schedule your actions to take place on certain dates, days of the week, and within time-slots.

For more information, see Create a Condition.

6 Complete the required option (Android only) **Minimum Time Between Actions (hours)** which limits the number of times the action is run when triggered by the prescribed event.

This option is not available when **Recurring Schedule** is included in the selected conditions.

You can select zero hours for this option but only for **AC Power** conditions.

7 Select an **Action** to perform. The actions available depend on the device platform.

Table 4-13. Event-Action Actions by Platform

Action	Android	WinRugg	Description
Apply Custom Settings.	✓		Apply custom, OEM-specific device settings based on the selected XML file, which must be delivered to the device through a separate File/Action. If the file entered in the **File Path** text box is not found, then the Event Action does not run. However, the Event Action remains active, making attempts to run the custom setting XML file each time the condition is met. Supported Devices: ■ Android MSI devices with the Android Hub v7.1+ ■ ■ Upload the ZIP file created by MSI. ■ Android Zebra devices with the Android Hub v7.2+ ■ ■ Upload the XML configuration file created by the Zebra Stage Now program.
Copy Files		✓	Copy files from one location to another on the WinRugg device.
Create Folder.		✓	Create a folder on the WinRugg device.
Delete Files.		✓	Delete folders from the WinRugg device.
Device Check-in		✓	Command the Workspace ONE Intelligent Hub to check in the WinRugg device for updates.
Download Files.	✓	✓	You can download a file from the selected File Server source path onto the selected Destination folder of your Android or Windows Rugged device. For details, see the section on this page entitled **Configure Download and Upload Actions for Event Actions**.
Install		✓	Install files on the Windows Rugged device. You must use the Run manifest action to install files or applications by using command lines. Supports the following file types: .reg, .cab, and .xml
Launcher Logout	✓		Checks-in the device, making it ready for another end user.
Move Files		✓	Move files from one location to another on the WinRugg device.
Reboot	✓		Restart the Android device.
Remove Folder.		✓	Remove a folder from the WinRugg device.
Rename File.		✓	Rename a file on the WinRugg device.

Table 4-13. Event-Action Actions by Platform (continued)

Action	Android	WinRugg	Description
Rename Folder.		✓	Rename a folder located in the WinRugg device.
Run		✓	Run command lines and arguments on the WinRugg device. If you want to install executable files (EXE), then you must use the Run manifest action on Windows Rugged devices. This is accomplished using command lines. The Run command must use the syntax of "\[full file path]". For example, \program files\program.exe.
Run Intent.	✓		Run command lines and arguments on the Android device. See RunIntent Action, File-Action Android for more information.
Send Sample Data.		✓	Send the WinRugg device data sample to the UEM console.
Terminate		✓	End a process or application running on the WinRugg device.
Uninstall		✓	Uninstall a program or application on the WinRugg device. You must enter the application name. The application name must match the name that appears in the Uninstall menu in the Control Panel. **Note** The Uninstall Manifest is for deleting files when a product is removed. If you remove a product from a device, any files installed remain on the device until uninstalled using an Uninstall Manifest.
Upload Files	✓		You can upload a file from a source file path on your Android device to the selected File Server destination folder. For details, see the section on this page entitled **Configure Download and Upload Actions for Event Actions**.
Warm Boot		✓	Restart the WinRugg device.

8 Select **Update** to add the action to the event action. You can add additional actions to the event action. Select **Next**.

9 Review the **Summary** and select **Save**.

What to do next: To push event actions to devices, add them as a component to a product. For more information, see Create a Product

Configure Download and Upload Actions for Event Actions

You must select a previously configured file server when you make Download Files or Upload Files the action of your Event Action. Making such a selection provides a source and destination for these files.

If you have not yet configured a file server for this purpose, see the section on this page entitled **Add a File Server for Event Actions**.

You can only see these options in the **Actions** tab of the **Add Event Action** screen, and only when you select as your action either **Download Files** or **Upload Files**.

1 Complete the following options to configure the Download Files or Upload Files action.

Setting	Description
Select Action	This option is pre-populated with either Download Files or Upload Files.
File Server	Select from the list of **Existing** file servers you have configured previously. If no file servers are configured, select the **Create New** button followed by **New File Server** to add a file server.
Source	For Download Files, the source is a file on the file server. Enter the path on the file server where the file can be found. For Upload Files, the source is a file on your device. Enter the directory or file path on your device where the file can be found.
Destination	For Download Files, the destination is your device. Enter the directory or file path on your device where the file is going. For Upload Files, the destination is the file server. Enter the path on the file server where the file is going.
Folder Naming	This option is available only when Upload Files is the **Select Action**. When multiple Upload Files actions use the same destination folder, there is a risk that identically named files might overwrite each other. To reduce this risk, files can be uploaded into dynamically generated child folders for each assigned device. You can use a lookup value to give identically named files a unique folder name. When you review the uploaded files on the file server, the folder name identifies the user or device from which it came. Select one or more of the following lookup values to make the destination folder name unique. ■ DeviceAssetNumber ■ DeviceUid ■ DeviceFriendlyName ■ User name ■ UserEmailAddress For more information, see Chapter 7 Lookup Values. You can also perform a search on docs.vmware.com.

2 Select the **Update** button to save your configuration.

File Servers for Event Actions

You can configure a file server as a staging and provisioning component. This file server component is in support of the Upload File and Download File actions in Event Actions.

You can coordinate this file server with an organization group and content gateway in Workspace ONE UEM console.

Add a File Server for Event Actions

File servers are used as the source or destination of a download files or upload files event action.

1 Navigate to **Devices > Provisioning > Components > File Servers** and then select **Add File Server**.

2 Complete all applicable settings in the tabs that are displayed.

Setting	Description
Name	Enter a name for the file server.
Link	Enter the path of the source or destination of files. For example, \\home\NetworkFileShare
Organization Group	Enter the organization group you want to coordinate with for the File Server.
Content Gateway	Select the content gateway that is reachable from this resource. Only the Unified Access Gateway (UAG) version is supported. The selection of a content gateway is required.
User name	Enter the user name that is recognized by the file server.
Password	Enter the password that accompanies the user name.
Test Connection Status	Select the **Test Connection** button to test whether the file server can be connected to with the user name and password entered.

Files-Actions for Products

In product provisioning for Workspace ONE UEM powered by AirWatch, a file-action is the combination of the files you want on a device plus the actions you want performed on the device with the file. You assign a file-action to a product. The product is then assigned to the device using Smart Group assignment.

You can install, configure, and upgrade devices by assigning Files-Actions to a product. The Files-Actions component also contains ways to manage the file system of a device.

View the files-actions in the Files-Actions List View.

Table 4-14. Platform-Specific Actions

				Windows	Windows	Windows
Actions	Android	macOS	QNX	7	Desktop	Rugged
Copy Files.	✓	✓	✓	✓	✓	✓
Create Folder.	✓	✓	✓	✓	✓	✓
Delete Files.	✓	✓	✓	✓	✓	✓
Execute Script.		✓				
Install		✓	✓	✓	✓	✓

Table 4-14. Platform-Specific Actions (continued)

Move Files.	✓	✓	✓	✓	✓	✓
Remove Folders.	✓	✓	✓	✓	✓	✓
Rename File.	✓	✓	✓	✓	✓	✓
Run.		✓	✓	✓	✓	✓
Run Intent.	✓					
Reboot	✓					
Terminate.			✓	✓	✓	✓
Uninstall.		✓		✓	✓	✓
Warm Boot						✓
OS Upgrade	✓					
Workspace ONE Intelligent Hub Upgrade.	✓					

Create a Files-Actions Component

You can install, configure, and upgrade devices by assigning files/actions to a product. The files/actions component also contains ways to manage the file system of a device.

Note **Windows Rugged and Windows Desktop:** Windows Unified Agent is a 32-bit application, so when trying to run scripts in a 64-bit machine, proper redirections must be used to get access to the 64-bit folder or the registry hive.

Procedure

1 Navigate to **Devices > Provisioning > Components > Files/Actions** and select **Add Files/Actions**.

2 Select the device Platform for which you want to make the files/actions.

3 Complete the **General** text boxes.

Settings	Descriptions
Name	Enter a name for the files/actions. The name cannot be longer than 255 characters.
Description	Enter a short description for the files/actions.
Version	The UEM console pre-populates this setting.

Settings	Descriptions
Platform	Read-only setting displays the selected platform.
Managed By	Select the organization group that can edit the files/actions.

4 Select the **Files** tab.

5 Select **Add Files**.

The **Add Files** window displays.

6 Select **Choose Files** and browse for a file or multiple files to upload.

There is a 2 GB limit on uploads.

Note **Windows Rugged** devices can use files/actions to install XML onto a device. For more information, see Create an XML Provisioning File, Android, Win7, WinRugg.

Android and **Windows Rugged** devices can use the File-Action function to upgrade the OS of the device remotely. For more information, see Create an OS Upgrade File-Action, Android and Create an OS Upgrade File-Action.

7 Select **Save** and upload the files.

Once the files upload, the file grouping screen opens. File groups allow you to assign different download paths and settings to different groups of files you have uploaded to a single file/action.

8 Select uploaded files and select **Add** to move the files into a new file group.

9 Define the **Download Path** the device uses to store the file group in a specific device folder.

If the download path entered does not exist, the folder structure is created as part of the installation.

Note **Windows Rugged** devices can **Store OS Update Files on Relay Server** to ensure that files used to update the OS remain available on the relay server.

10 Select **Save**.

You can repeat the previous steps for as many files as you want.

11 Select the **Manifest** tab.

If you have at least one file uploaded, Actions are not required.

12 Add actions to the **Install Manifest** or the **Uninstall Manifest** if needed.

The uninstall manifest only runs when the Uninstall action is added to the product. If nothing is added to the Uninstall Manifest, uninstalling the file/action results in no effect.

Select the platform for which you are making a File-Action.

Option	Description
Android	Files-Actions Manifest Options for Android
macOS	Files-Actions Manifest Options for macOS
QNX	Files-Actions Manifest Options for QNX
Windows Desktop	Files-Actions Manifest Options for Windows Desktop
Windows Rugged	Files-Actions Manifest Options for WinRugg

Files-Actions Manifest Options for Android

You must select all the desired Android options for your Files-Actions product component.

Procedure

1 Add the Manifest Actions for your Android File-Action.

Settings	Descriptions
Workspace ONE Intelligent Hub Upgrade	Install the new Workspace ONE Intelligent Hub to the device. Before using this file/action, see Upload the Workspace ONE Intelligent Hub APF File, Upgrade File-Action for more information.
Apply Custom Settings	Apply custom, OEM-specific device settings based on the selected XML file. You must upload the custom XML or ZIP file as part of the file/action. Supported Devices: ■ Android Motorola Solutions devices with the Android Hub v7.1+ ■ ■ Upload the ZIP file created by MSI. ■ Android Zebra devices with the Android Hub v7.2+ and Zebra's MX Service App installed on the device. ■ ■ Create your XML configuration file using Zebra Stage Now. ■ ■ Upload the XML configuration file and select it from the drop-down menu. ■ ■ After pushing the product containing an Apply Custom Setting file/ action, the status information reports in the Job Log. If an error occurs, the failed response XML is reported in the Job Log. For more information, see Product Job Statuses.
Copy Files	Copy files from one location to another on the device.
Create Folder	Create a new folder on the device.
Delete Files	Delete folders from the device.
Install Unmanaged Application	Install an unmanaged .APK file. Workspace ONE UEM does not add the app to the managed app list. Enterprise wipes or unenrollment do not remove the app from the device. You must use the Uninstall Unmanaged Application file/action. Consider adding Uninstall Unmanaged Application to the uninstall manifest of any product including the Install Unmanaged Application file/ action.
Move Files	Move files from one location to another on the device.

Settings	Descriptions
OS Upgrade	Install a new OS upgrade and the relevant Workspace ONE Intelligent Hub. For more information on this option, see Create an OS Upgrade File-Action, Android.
Reboot	Restart the device.
Remove Folder	Remove a folder from the device.
Rename File	Rename a file on the device.
Rename Folder	Rename a folder located in the device.
Run Intent	Run command lines and arguments on the device. See RunIntent Action, File-Action Android for more information.
Uninstall Unmanaged Application	Uninstall an unmanaged application. Enter the package ID of the app.

2 When finished adding actions to the **Manifest**, select **Save**.

What to do next

Path Variables – For all file management-related actions listed above (copy files, create folder, delete files, move files, remove folder, rename file, and rename folder), you have the option of inserting a path variable for both source and target, as applicable. The use of these variables in your Files/Actions path means you do not need to account for the randomly generated OEM-specific path definitions in the creation of your Files/Actions.

$internal$ – Use this variable at the beginning of your path to indicate your source/target path to be read from/written to the internal storage space. Supports read and write actions. For example: `/$internal$/agreement/license.txt` addresses the file `license.txt` in the agreement folder on the device's internal storage space.

Note $internal$ does not work with all Files/Actions.

$external$ – Use this variable at the beginning of your path to indicate your source path to be from the external memory card storage, which the device must feature. External storage supports read-only access so any usage must involve a memory card that has been properly formatted and furnished with the correct files in the correct locations. For example: `/$external$/sdcard/license.txt` reads the file `license.txt` from the `sdcard` folder found on the device's external memory card storage.

Files-Actions Manifest Options for macOS

You must select all the desired macOS options for your Files-Actions product component.

Procedure

1 Add the Manifest Actions for your macOS File-Action.

Settings	Descriptions
Execute Script	Runs the selected script on the device. This command supports .sh and .scpt files. You must enter the script file path and name. Select **Execute as Root** to run the script as the Root user. If you do not enable this option, the script runs as the user currently logged in.
Install	Install files on the device. This is accomplished using command lines. Supports the following file types. **macOS** DMG, PKG, or APP (zipped) If the DMG file contains an APP file, Workspace ONE UEM moves the APP file to the /Applications folder. If the DMG contains a PKG or MPKG file, extract the file from the DMG and push the PKG or MPKG directly. Workspace ONE UEM supports installing and managing .app files as internal applications which provide additional control for removing apps upon unenrollment.
Run	Use the manifest to run an application. This is accomplished using command lines. The Run command must use the syntax of "\[full file path]". For example, \program files\program.exe. You must select the context of the command. Select whether the command runs at the system level, the user level, or the admin account level. For Windows Rugged devices, Workspace ONE UEM supports 3 file types, EXE, AWS, and LNK. The EXE is the app itself while AWS is the AirWatch supported scripting language. LNK files support an inferred execution based on the file extension. For example, if a DOC file is run, the device would use whatever app is associated with DOC files. **Note** With macOS devices, you can run any root command that you normally use within Terminal. The Workspace ONE Intelligent Hub automatically appends sudo before running any command.
Uninstall	Uninstall a program or application on the device. You must enter the application name. **Note** The Uninstall Manifest is for deleting files when a product is removed. If you remove a product from a device, any files installed remain on the device until uninstalled using an Uninstall Manifest.

2 When finished adding actions to the **Manifest**, select **Save**.

Files-Actions Manifest Options for QNX

You must select all the desired QNX options for your Files-Actions product component.

Procedure

1 Add the Manifest Actions for your QNX File-Action.

Settings	Descriptions
Copy Files	Copy files from one location to another on the device.
Create Folder	Create a new folder on the device.
Delete Files	Delete folders from the device.
Install	Install files on the device. This is accomplished using command lines.
Move Files	Move files from one location to another on the device.
Remove Folder	Remove a folder from the device.
Rename File	Rename a file on the device.
Rename Folder	Rename a folder located in the device.
Run	Use the manifest to run an application. This is accomplished using command lines. The Run command must use the syntax of "\[full file path]". For example, \program files\program.exe. You must select the context of the command. Select whether the command runs at the system level, the user level, or the admin account level. For Windows Rugged devices, Workspace ONE UEM supports 3 file types, EXE, AWS, and LNK. The EXE is the app itself while AWS is the AirWatch supported scripting language. LNK files support an inferred execution based on the file extension. For example, if a DOC file is run, the device would use whatever app is associated with DOC files.
Terminate	End a process or application running on the device.

2 When finished adding actions to the **Manifest**, select **Save**.

Files-Actions Manifest Options for Windows Desktop

You must select all the desired Windows Desktop options for your Files-Actions product component.

Procedure

1 Add the Manifest Actions for your Windows Desktop File-Action.

Settings	Descriptions
Copy Files	Copy files from one location to another on the device.
Create Folder	Create a new folder on the device.
Delete Files	Delete folders from the device.
Install	Install files on the device. This is accomplished using command lines. Supports the following file types. CAB, MSI, REG, and XML. CAB and MSI files contain the app itself while REG and XML files are for modifying the registry settings. REG files require batch files and PowerShell commands.

Settings	Descriptions
Move Files	Move files from one location to another on the device.
Remove Folder	Remove a folder from the device.
Rename File	Rename a file on the device.
Rename Folder	Rename a folder located in the device.
Run	Use the manifest to run an application. This is accomplished using command lines. The Run command must use the syntax of "\[full file path]". For example, \program files\program.exe. You must select the context of the command. Select whether the command runs at the system level, the user level, or the admin account level.
Terminate	End a process or application running on the device.
Uninstall	Uninstall a program or application on the device. You must enter the application name. The application name must match the name that appears in the Uninstall menu in the Control Panel. **Note** The Uninstall Manifest is for deleting files when a product is removed. If you remove a product from a device, any files installed remain on the device until uninstalled using an Uninstall Manifest.

2 When finished adding actions to the **Manifest**, select **Save**.

Files-Actions Manifest Options for WinRugg

You must select all the desired Windows Rugged options for your Files-Actions product component.

Procedure

1 Add the Manifest Actions for your Windows Rugged File-Action.

Settings	Descriptions
Workspace ONE Intelligent Hub Upgrade	Install the new Workspace ONE Intelligent Hub to the device. Before using this file/action, see Upload the Workspace ONE Intelligent Hub APF File, Upgrade File-Action for more information.
Copy Files	Copy files from one location to another on the device.
Create Folder	Create a new folder on the device.
Files	Delete folders from the device.
Install	Install files on the device. This is accomplished using command lines. Supports the following file types. CAB, REG, and XML. CAB files contain the app itself while REG and XML files are for modifying the registry settings. Consider using the Workspace ONE UEM CAB Creator to create CAB files that combine multiple files into one CAB file.
Move Files	Move files from one location to another on the device.
Remove Folder	Remove a folder from the device.

Settings	Descriptions
Rename File	Rename a file on the device.
Rename Folder	Rename a folder located in the device.
Run	Use the manifest to run an application. This is accomplished using command lines. The Run command must use the syntax of "\[full file path]". For example, \program files\program.exe. You must select the context of the command. Select whether the command runs at the system level, the user level, or the admin account level. For Windows Rugged devices, Workspace ONE UEM supports 3 file types, EXE, AWS, and LNK. The EXE is the app itself while AWS is the AirWatch supported scripting language. LNK files support an inferred execution based on the file extension. For example, if a DOC file is run, the device would use whatever app is associated with DOC files. **Note** With macOS devices, you can run any root command that you normally use within Terminal. The Workspace ONE Intelligent Hub automatically appends sudo before running any command.
Terminate	End a process or application running on the device.
Uninstall	Uninstall a program or application on the device. You must enter the application name. WD W7 The application name must match the name that appears in the Uninstall menu in the Control Panel. **Note** The Uninstall Manifest is for deleting files when a product is removed. If you remove a product from a device, any files installed remain on the device until uninstalled using an Uninstall Manifest.
Warm Boot	Restart the device.

2 For each manifest action, select a **Critical** option. If the critical 'success' or 'error' options are selected, then the action is skipped if the condition is not met.

Option	Condition
Continue on Error	Execute this manifest action only if the previous action fails.
Continue	Execute this manifest action regardless of the outcome of the previous action.
Continue on Success	Execute this manifest action only if the previous action succeeds. If the previous action is skipped, then this manifest action is also skipped.

3 When finished adding actions to the **Manifest**, select **Save**.

Editing Files-Actions

You can edit existing Files-Actions to keep products and devices up-to-date.

When you edit any existing files/actions, the version number increases. After saving the edits, Workspace ONE UEM runs a check against all active products to find any that contain the newly edited files/actions.

If any active products contain the files/actions, a warning prompt displays listing all active products affected by the edited files/actions. You can then **Activate** or **Deactivate** a product using the files/actions.

Delete Files-Actions

Workspace ONE UEM checks any attempt to delete files-actions against the list of active products. To delete files-actions, it must be detached from all products.

Procedure

1. Select the **Files/Actions** listed in the Warning prompt.
2. Select **Edit**.
3. Remove the files-actions from the product.
4. Select **Save**.
5. Repeat for all products containing the files-actions.
6. After the files-actions detach from all products, you can delete the files-actions.

Results

If the files-actions is part of an active product, a warning prompt displays listing any product that uses the files/actions.

Import Motorola Packages in Files-Actions, Android and WinRugg

You can import MSP (Motorola Services Platform) packages, which can be unpacked into proper files/actions for use in products.

Procedure

1. Navigate to **Devices > Provisioning > Components > Files/Actions** and select **Add**.
2. Select the Platform you want to create a staging configuration for.
3. Select **Import Package**.
4. Select **Upload** to add an APF file.

   Once the file is uploaded, the required text boxes are auto-completed.
5. Select **Save**.

Create an XML Provisioning File, Android, Win7, WinRugg

XML provisioning allows you to download a custom-designed XML file to a device in a provisioning product. After the file is downloaded, it runs an install command to extract the settings from the XML file and install them on the device.

Procedure

1. Navigate to **Devices > Provisioning > Components > Files/Actions** and select **Add Files/ Actions**.
2. Select your platform.
3. Enter the required settings on the **General** tab, then select the **Files** tab and upload the desired XML file and enter the destination path on the device.
4. Select the **Manifest** tab and **Add** an **Install Action** for the XML file.
5. Select **Save**.
6. Navigate to **Devices > Provisioning > Products List View**, and select **Add Product**.
7. Select your platform.
8. Enter the **General** information.
9. Select the **Manifest** tab.
10. Select **Install Files/Actions** and select the files and actions just created.
11. **Save** and **Activate** the product.

Results

The product downloads to all assigned devices and the XML file successfully installs.

Example

Windows Rugged

The following is a sample of an XML file which updates a registry setting that can be installed on a device through product provisioning.

XML Provisioning is for Windows Mobile devices only and not Windows CE devices.

```
<?xml version="1.0"?>
<wap-provisioningdoc name="desiredDocName /V_1">
<characteristic type="com.windowspc.getregistryinfo.managed">
<reg_value value_name="KeyName"
<!-- (i.e. CommonFilesDir) --
key_name="RegistryPath"
<!-- (i.e.Software\Wow6432Node\Microsoft\Windows\CurrentVersion) --
custom_attribute_name="AttributeName"/>
<reg_value value_name="Keyname ..." key_name="Path\...."
custom_attribute_name="AttributeName2"/>
</characteristic>
</wap-provisioningdoc>
```

Windows 7

```
<?xml version="1.0"?>
<wap-provisioningdoc>
<characteristic type="com.airwatch.getregistryinfo.winpc">
```

```
<reg_value value_name="KeyName"
<!-- (i.e. CommonFilesDir) --
key_name="RegistryPath"
<!-- (i.e. Software\Wow6432Node\Microsoft\Windows\CurrentVersion)" --
custom_attribute_name="AttributeName"/>
<reg_value value_name="Keyname ..." key_name="Path\...."
custom_attribute_name="AttributeName2"/>
</characteristic>
</wap-provisioningdoc>
```

Android

```
<?xml version="1.0"?>
<attributes>
<attribute name="attribute 1" value="value 1"/>
<attribute name="attribute 2" value="value 2"/>
<attribute name="attribute 3" value="value 3"/>
</attributes>
```

Workspace ONE Intelligent Hub Upgrading a File-Action

When you upgrade your devices, you can seed the Workspace ONE Intelligent Hub in the Workspace ONE UEM console for use in products. The file-action Workspace ONE Intelligent Hub Upgrade then grabs the list of seeded APF files when creating a manifest action for products.

Use this option to enroll devices with older Hub versions installed. You can enroll the devices then upgrade the device to the new Hub version you want to use.

When using this upgrade option, be alert for failed upgrades. A failed upgrade can cause the product to push repeatedly as the console recognizes the older Hub version. This failed upgrade can tax the network and result in excessive battery use. If the upgrade fails, deactivate the product and look over the configuration to ensure that the settings are correct.

Note The Hub Packages screen is only accessible in Customer type organization groups.

Upload the Workspace ONE Intelligent Hub APF File, Upgrade File-Action

The Hub Package can be uploaded only in specific organization group types, for example, in organization groups of type ‘Customer’. Upload the Workspace ONE Intelligent Hub Package to the topmost organization group level. You can find the file specific to your OEM located in Workspace ONE UEM Resources.

Prerequisites

In order to ensure the Workspace ONE Intelligent Hub persists on your Zebra device, you must supply the correct APF file, which you can download from my.workspaceone.com. Zebra Devices have the following APF files available.

- **Workspace ONE Intelligent Hub xx.xx Zebra APF for Android Enterprise Work Managed (DO)** – Applies to devices in Android Enterprise Work Managed mode together with Agent Upgrade and StageNow barcode enrollment.
- **Workspace ONE Intelligent Hub xx.xx Zebra APF for Android Legacy (DA)** – Applies to devices in Android Legacy mode together with Agent Upgrade and StageNow barcode enrollment.
- **Workspace ONE Intelligent Hub xx.xx Zebra APF for Android (RdClient - DA)** – Applies to devices in Android Legacy mode together with Agent Upgrade and Rapid Deployment barcode enrollment.

Procedure

1. Download the APF files suited to your device.
2. Navigate to **Devices > Provisioning > Components > Hub/Agent Packages** and select **Add Hub/Agent Package**. Make sure that you are using the top-level organization group.
3. Select the platform for which you are adding the Workspace ONE Intelligent Hub package.

   The **Add Hub/Agent Package** screen displays.
4. Select the **Upload** button next to the **Application File** setting.
5. Select **Choose File** and browse for the APF file of the Workspace ONE Intelligent Hub version you want to upload.
6. Select the APF file and select **Open** to select the file.
7. Select **Save** and close the upload dialog box.

   With the uploading of the APF file, the settings are populated with data automatically.
8. Make any desired edits to **File Name**, **Package Name**, and **Version** for the Workspace ONE Intelligent Hub.
9. Select **Save** and upload the APF file to the UEM console.

RunIntent Action, File-Action Android

The runIntent action starts an Android intent that facilitates late runtime binding between the code in different applications. Use these intents to accomplish actions on your Android devices.

The most significant use of runIntent is the launching of activities, where it can be thought of as the glue between activities. It is a passive data structure holding an abstract description of an action to be performed. The runIntent action supports both explicit and implicit intents.

Depending on the arguments used, the Workspace ONE Intelligent Hub uses either of the following to start the specified intent.

- android.content.Context.startActivity(Intent intent)
- android.content.Context.sendBroadcast(Intent intent) to run the specified intent.

RunIntent Syntax

The argument syntax changes depending on whether explicit or implicit mode is specified.

```
mode=explicit, broadcast=[true|false] , action=< action>, package=<package>, class=<class>
[, data=<data>][, extraString=<stringname>=<string value>[,...]][, extraInt=<int name>=<int
value>[,…]]
```

```
mode=implicit, broadcast=[true|false] , action=<action> [,category=<category>][, uri=<uri>]
[, data=<data>] [, extraString=<string name>=<string value>[,...]][, extraInt=<int name>=<int
value>[,…]]
```

Table 4-15. Arguments

Argument	Explanation
mode=[explicit|implicit]	Specifies whether the intent is explicit or implicit.
broadcast=[true|false]	Specifies whether the intent to be launched using startActivity() or sendBroadcast().
action=<action>	Specifies the Android action string for the intent. An example of an Android action string is android.intent.action.MAIN.
package=<package >	Specifies the Android package name of the java class to be explicitly run. Android package names are generally of the format com.mycompany.myapplication.
class=<class>	Specifies the java class in the specified package that is to be explicitly launched.
uri=<uri>	Specifies the URI that is to be passed with the implicitly launched intent.
category =<category>	Specifies the Android category string that is to be passed with the implicitly launched intent. An example of an Android category string is android.intent.category.DEFAULT
data=<data>	Specifies the value of the Android data parameter that is to be passed with the explicitly or implicitly launched intent.
extraBoolean	Specifies the name of an extra boolean parameter that is to be passed with the explicitly or implicitly launched intent. Boolean value specifies the value of the extra boolean. The extraBoolean argument can be used multiple times to specify additional extra boolean name/values.
extraBooleanArray	Specifies the name of an extra boolean array parameter that is to be passed with the explicitly or implicitly launched intent. Boolean array value specifies the value of the extra boolean array. The extraBooleanArray argument can be used multiple times to specify additional extra boolean array name/values.

Table 4-15. Arguments (continued)

extraFloat	Specifies the name of an extra float parameter that is to be passed with the explicitly or implicitly launched intent. Float value specifies the value of the extraFloat. The extraFloat argument can be used multiple times to specify additional extraFloat name/values.
extraFloatArray	Specifies the name of an extra float array parameter that is to be passed with the explicitly or implicitly launched intent. Float array value specifies the value of the extraFloatArray. The extraFloatArray argument can be used multiple times to specify additional extra float array name/values.
extraInt=<int name>=<int value>	Specifies the name of an extra int parameter that is to be passed with the explicitly or implicitly launched intent. int value specifies the value of the extra int. The extraInt argument can be used multiple times to specify additional extra int name/values.
extraIntArray	Specifies the name of an int array parameter that is to be passed with the explicitly or implicitly launched intent. The int array value specifies the value of the extra int array. The extraIntArray argument can be used multiple times to specify additional extra int array name/values.
extraIntArrayList	Specifies the name of an int array list parameter that is to be passed with the explicitly or implicitly launched intent. The int array list value specifies the value of the extra int array list. The extraIntArrayList argument can be used multiple times to specify additional extra int array list name/values.
extraLong	Specifies the name of an extra long parameter that is to be passed with the explicitly or implicitly launched intent. Long value specifies the value of the extra long. The extraLong argument can be used multiple times to specify additional extra long name/values.
extraLongArray	Specifies the name of an extra long array parameter that is to be passed with the explicitly or implicitly launched intent. Long array value specifies the value of the extra long array. The extraLongArray argument can be used multiple times to specify additional extra long array name/values.
extraString=<string name>=<string value>	Specifies the name of an extra string parameter that is to be passed with the explicitly or implicitly launched intent. string value specifies the value of the extra string. The extraString argument can be used multiple times to specify additional extra string name/values.
extraStringArray	Specifies the name of a string array parameter that is to be passed with the explicitly or implicitly launched intent. The string array value specifies the value of the extra string array. The extraStringArray argument can be used multiple times to specify additional extra string array name/values.
extraStringArrayList	Specifies the name of a string array list parameter that is to be passed with the explicitly or implicitly launched intent. The string array list value specifies the value of the extra string array list. The extraStringArrayList argument can be used multiple times to specify additional extra string array list name/values.

The following table indicates which arguments are required, optional, or not applicable for the explicit and implicit modes.

mode	Explicit	Implicit
broadcast	required	required

action	required	required
package	required	n/a
class	required	n/a
uri	n/a	optional
category	n/a	optional
data	optional	optional
extraString	optional	optional
extraInt	optional	optional

Example RunIntent

```
mode=explicit,broadcast=false,action=android.intent.action.MAIN,package=com.examples.myappl,cl
ass=com.examples.myappl.MainActivity
```

APK File Installation

You can use a runIntent action on an APK file on the device's local storage which installs an application on the device.

RunIntent Syntax for APK File Installation

```
mode=implicit,broadcast=false,action=com.airwatch.android.provisioning.INSTALL_APKS_FROM_FOLDE
R,package=com.airwatch.androidagent,extraString=path=/storage/emulated/Download
```

- You must customize the path in the highlighted portion to account for your specific file and folder structure.
- You can specify an individual APK file in this path on the runIntent which installs an application on the device.
- You can also specify a folder in the path of the runIntent, which runs all APK files found in that folder.
- Applications installed on a device using APK files by way of a runIntent are unmanaged.
- You can also use a path variable in the runIntent to represent the device's internal or external storage.

Path Variable Use in RunIntent for APK Installation

$internal$ – Use this variable at the beginning of your path to indicate your source/target path to be read from/written to the internal storage space. Supports read and write actions. For example: `/$internal$/agreement/license.txt` addresses the file `license.txt` in the agreement folder on the device's internal storage space.

Note $internal$ does not work with all Files/Actions.

$external$ – Use this variable at the beginning of your path to indicate your source path to be from the external memory card storage, which the device must feature. External storage supports read-only access so any usage must involve a memory card that has been properly formatted and furnished with the correct files in the correct locations. For example: `/$external$/sdcard/license.txt` reads the file `license.txt` from the `sdcard` folder found on the device's external memory card storage.

Upgrade the OS, File-Action for Android

You can upgrade the OS of your Android devices remotely using product provisioning. This process allows you to keep your entire device fleet up-to-date without needing to have the devices shipped back to you.

After an Android device receives an Android OS Upgrade file/action, the device processes the command in the following order.

Prerequisites

- Support includes Zebra devices using the Zebra MX Service and any OEM supporting the Platform OEM Service v3.0 or later. For more information about the Platform OEM Service, see the **VMware Workspace ONE UEM Android Platform Guide** topics titled **Android OEM Services** and **Platform OEM Service Overview**.
- **Note** Before updating your Motorola device to a new Zebra OS, you must have the Workspace ONE Intelligent Hub for Android v5.1.4+ installed and the 1.9 MX service.

Procedure

1. The device receives the product which you can verify on the device by navigating to **Hub > Products**.
2. Download all the files including the OS update zip which you can verify in the Product logs found on the device in **Hub > Products > Product Name**.
3. Once the downloads complete on a Zebra device, the Workspace ONE Intelligent Hub backs up its data and any installed managed applications to the device enterprise folder which is persistent.
4. The Hub then reboots the device into recovery mode to install the update.
5. Device then applies the OS update.

6 Once complete, the device reboots.

Results

After reboot, the Workspace ONE Intelligent Hub validates that the OS update is successful before reporting that the job completed successfully.

What to do next

If the OS update fails, you can investigate the reasons why by using the Workspace ONE UEM console to navigate to **Devices > Provisioning > Product Dashboard**. Select the failed product in **Recent Product Status** that contains the OS upgrade. In the **View Devices** screen, select the magnifying glass icon to view history, then select the magnifying glass icon again to view the **Job Log**.

Create an OS Upgrade File-Action, Android

Upgrade Android devices remotely with the OS Upgrade File-Action. Add the file-action to a product to update your devices without a need for a manual OS update.

This OS Upgrade task applies to non-Zebra devices including Honeywell devices running any version of Android. If you want to OS Upgrade a Zebra device running Android 8.0 or later, then see Create an OS Upgrade for Zebra Devices, Android 8.0+.

1 Navigate to **Devices > Provisioning > Components > Files/Actions** and select the **Add Files/ Actions** button.

2 Select the **Android** platform.

3 Complete the General text boxes.

- Enter a **Name**.
- Enter a **Description**.
- View the pre-populated **Version** number of the file-action.
- View and confirm the **Platform** for the file-action.
- Enter who the files-actions are **Managed By**.

4 Select the **Files** tab.

5 Select the **Add Files** button and upload the OS update files.

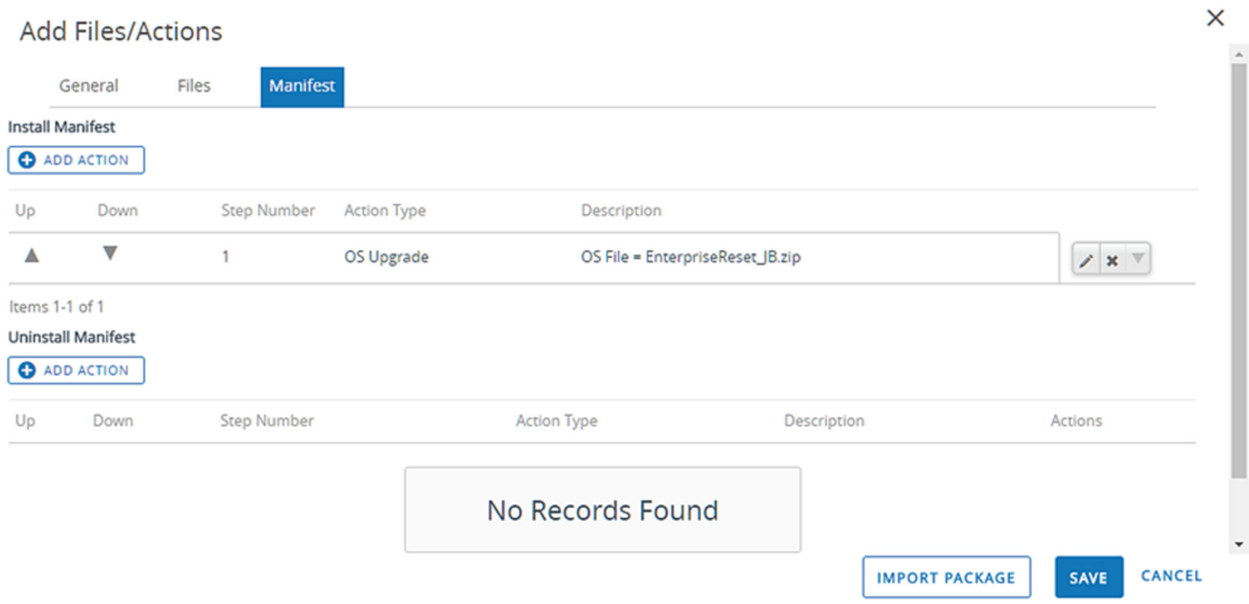
- For **Zebra devices under versions of Android before 8.0**, upload the following files and specify the path as either `/data/tmp` to store the file on the data partition, or as a known internal path to store it on the internal storage.
  - OS Update ZIP file – This file can only be an incremental OS upgrade file. The file can also be an enterprise reset package.
  - [optional] Workspace ONE Intelligent Hub update package (APF) – This optional file can be specified to update the Workspace ONE Intelligent Hub before initiating the actual OS update. Workspace ONE UEM can provide this APK.

- For **Honeywell** devices, direct the OS Update ZIP file to be placed in the `/sdcard/honeywell/autoinstall` folder. Then issue a warm boot on the device. This forces the Honeywell device to run the OS Update.
  - This applies to both incremental full Honeywell OS updates. It assumes that you have knowledge of the file requirements to run a Honeywell OS update before attempting an OS Update File-Action.
  - Additionally, these steps rely on the Honeywell Autoinstall process, which means this process must not be disabled prior to attempting a Honeywell OS update.
  - Lastly, if your Honeywell device is under Android Enterprise (fully managed) enrollment, the Honeywell Setting called "Provisioning Mode" does not need to be enabled.

6 Select the **Manifest** tab and select **Add Action** under the **Install Manifest**.

7 Add OS Upgrade command to the manifest and select the corresponding OS upgrade file that was uploaded earlier.

Your Manifest tab looks similar to the following.



8 Select **Save**.

After creating an OS Upgrade file/action, create a product to push the upgrade to your devices. See Create a Product for more information.

Note Before installing an OS Update, the device checks the battery level. If the level is below a threshold, the product fails. This failure displays in the log.

Create an OS Upgrade for Zebra Devices, Android 8.0+

Zebra devices running Android 8.0 (Oreo) or later with MX 9.1 or later support OS Upgrade using OEMConfig. However, for Zebra devices running Android 8.0 that do not have MX 9.1, this special OS Upgrade task represents a workaround to enable the same convenience of silent, remote OS upgrades.

This OS Upgrade task applies specifically to **Zebra devices running Android 8.0 or later**. If you want to OS Upgrade another Android device of any version or a Zebra device running a version of Android before 8.0 (Oreo), then see Create an OS Upgrade File-Action, Android.

If your Zebra device has MX 9.1 or later installed and is enrolled with Android Enterprise, you can integrate OEMConfig to upgrade the OS for you.

- For more information on integrating the OEMConfig app with Workspace ONE UEM, see OEMConfig on Android Enterprise Devices from the **Application Management for Android Guide** on docs.vmware.com.
- You can also reference Zebra's **OEMConfig Managed Configuration Guide** at https://techdocs.zebra.com/oemconfig.

Prerequisites

Before you begin, decide whether you want to apply a Full BSP upgrade or an Incremental (Lifeguard Patch) upgrade.

The Full upgrade applies the entire OS image in the system update file. This type of upgrade is best suited if you want to go from one major version to another. For example, applying Zebra devices with Android 6.0 or 7.0 all the way up to 8.0.

The Incremental or Lifeguard Patch upgrade is for when you want to apply "decimal point" upgrades within a version number. For example, going from Android 8.0 to 8.1 or applying a security patch.

Procedure

1 For **Full (BSP) Upgrades**, take the following steps.

   a Copy the following code (Ctrl-C) and save it as an XML file.

```
<wap-provisioningdoc>
   <characteristic version="8.1" type="PowerMgr">
      <parm name="ResetAction" value="8" />
      <characteristic type="file-details">
         <parm name="ZipFile" value="/data/tmp/public/FILENAME" />
      </characteristic>
   </characteristic>
</wap-provisioningdoc>
```

   b In the saved XML file, substitute the variable FILENAME for the actual filename of your Zebra OS upgrade file.

c Create a new File/Action. Open Create a Files-Actions Component in another tab of your browser and follow along with this task. When you get to the **Files** tab step, add two files:

- Add the OS upgrade zip file and set the download location to `/data/tmp/public/`.
- Add the modified XML file above and set the download location to any valid path. For example, `/sdcard/Downloads/`.

d In the **Manifest** tab, add an "Apply Custom Settings" action to the Install Manifest. Select the XML file from the drop down menu.

e Save the Files/Actions and provision it through a product.

f Task complete.

2 For **Incremental (Lifeguard Patches) Upgrades**, take the following steps.

a Copy the following code (Ctrl-C) and save it as an XML file.

```
<wap-provisioningdoc>
   <characteristic version="4.2" type="XmlMgr">
      <parm name="ProcessingMode" value="3" />
   </characteristic>
   <characteristic version="6.0" type="ConditionMgr">
      <parm name="DataType" value="1" />
      <characteristic type="BooleanDetails">
         <parm name="BooleanSourceType" value="2" />
         <parm name="BooleanSystemValue" value="2" />
      </characteristic>
      <parm name="ConditionMetAction" value="0" />
      <parm name="ConditionNotMetAction" value="1" />
      <parm name="ConditionFailMessage" value="" />
      <parm name="SuppressMessage" value="1" />
   </characteristic>
   <characteristic version="8.1" type="PowerMgr">
      <parm name="ResetAction" value="8" />
      <characteristic type="file-details">
         <parm name="ZipFile" value="data/tmp/public/FILENAME" />
      </characteristic>
   </characteristic>
   <characteristic version="4.2" type="XmlMgr">
      <parm name="ProcessingMode" value="4" />
   </characteristic>
</wap-provisioningdoc>
```

b In the saved XML file, substitute the variable FILENAME for the actual filename of your Zebra OS upgrade file.

c Create a new File/Action. Open Create a Files-Actions Component in another tab of your browser and follow along with this task. When you get to the **Files** tab step, add two files:

- Add the OS upgrade zip file and set the download location to `/data/tmp/public/`.
- Add the modified XML file above and set the download location to any valid path. For example, `/sdcard/Downloads/`.

d In the **Manifest** tab, add an "Apply Custom Settings" action to the Install Manifest. Select the XML file from the drop down menu.

e Save the Files/Actions and provision it through a product.

f Task complete.

Results

Once the product has been provisioned to your Zebra device running Android 8.0 or later but lacking MX 9.1, it will proceed with the automatic OS Upgrade. After the device reboots, it will be on the new OS version.

You can verify the OS has been successfully updated by checking the build number found by taking the following steps.

1 Navigate to **Devices > List View**.

2 Locate the recently upgraded Zebra device from the listing and select the device's Friendly Name in the **General Info** column.

- Selecting the device Friendly Name displays the **Details View**

3 Select **More > Custom Attributes**.

Upgrade the OS, File-Action for WinRugg

Using the Windows OS Upgrade file/action, upgrade your Motorola and Zebra Windows Rugged devices remotely. By creating a product containing the OS Upgrade file/action, you can upgrade all your devices without having them shipped back to you.

Before You Begin

For Windows Rugged devices, all OS upgrade files are restricted access files and require a valid user account with Motorola to log in with a valid device serial number. The OS upgrade files are specific to both device model and OS version. The Workspace ONE ™ UEM OS upgrade process uses the APF files and requires the administrator to first install the MSP Package Builder utility.

This utility is required to extract the contents of the APF file into individual components that are then pushed to the device and used to upgrade the OS.

Important The Windows Rugged OS Upgrade method requires the Workspace ONE Intelligent Hub v5.3+ for Windows Rugged devices.

To use the Windows Rugged OS Upgrade, you must have the following:

- The MSP Package Builder utility installed on your computer.
- A Motorola user account to download the OS Upgrade.
- The serial number for the device you want to upgrade.
- The OS update utility included with the extracted APF files. This file can be downloaded from the Zebra support website.
- A relay server configured to deliver the product to the device.

OS Upgrade Resources for WinRugg

The Windows Rugged OS Upgrade file/action might require Run command arguments to process the action on the device. You can also change the device registry settings.

Check Device Registry Settings

If at any point an error occurs, review the device registry settings. The 5.x Hub has been designed to update these settings based on the Workspace ONE ™ UEM console configuration settings. The relay server settings are a part of the job XML and are applied during provisioning just before processing the job.

If the OS Update process does not complete successfully, the following settings are monitored as part as any troubleshooting effort.

HKEY_LOCAL_MACHINE\SOFTWARE\AIRBEAM\	
IGNORESERVER	string “1”
SERVERRIP	string [Enter the Server URL or IP Address]
FTPUSER	string [Enter the FTPs user name]
FTPPASSWORD	string [Enter the FTPs password] - THIS FIELD MUST BE ENCRYPTED
TFTP	string "0"
PASSIVEMODE	string "1"
FTPPORT	string "21"
FTPS	string "0" or "1"
VERIFYSERVER	string "0"
SOFTKEY1	number "123"
SOFTKEY2	number "124"
ENTERKEY	number "13"

Run Command Argument Information

- -d option describes the Workspace ONE UEM relay server folder path containing files required to update the device. It uses the following format (with forward slashes and double quotes): "[path of your relay server]/PFILES/[file action name]_version/"
- -y option describes the maximum number of retry attempts while the –z option describes the retry delay.
- -q is the folder on the device that is used to download the files from the FTP server when updating Windows Mobile. This location is typically “\Temp” or “\Storage Card”. In CE updates, the files are downloaded one at a time into memory so the parameter is not used in that case.
- -p is the OSUpdate project name. The project name can be whatever the user wants it to be but is usually something to indicate what the update package contains. This name must be the same as the Motorola APF file. The project (APF) must be named so it can identify the target device, OS type, and so forth.

For the Workspace ONE UEM process, this name is the same as whatever you call the dummy APD file (which can be an empty text file).

- If your command line contained –p”WT41N0”, you must have a WT41N0.APD file put down to the device before applying the update. This is only needed for CE devices.
- For Windows Mobile devices –p”MC9190WM65” is required by the command-line parsing, but in effect is not used.

Extract Required OS Update Files

Before you can create an OS Upgrade file/action, you must extract the required files from the APF files. These files are extracted using the MSP Package Builder Utility.

1 Start the MSP Package Builder Utility.

2 Navigate to **File > Open Project** and select the appropriate APF file and open it in MSP Package Builder.

3 Navigate to **Tools > Convert Project** to open the **Convert to Project** dialog box.

4 Complete the following text boxes.

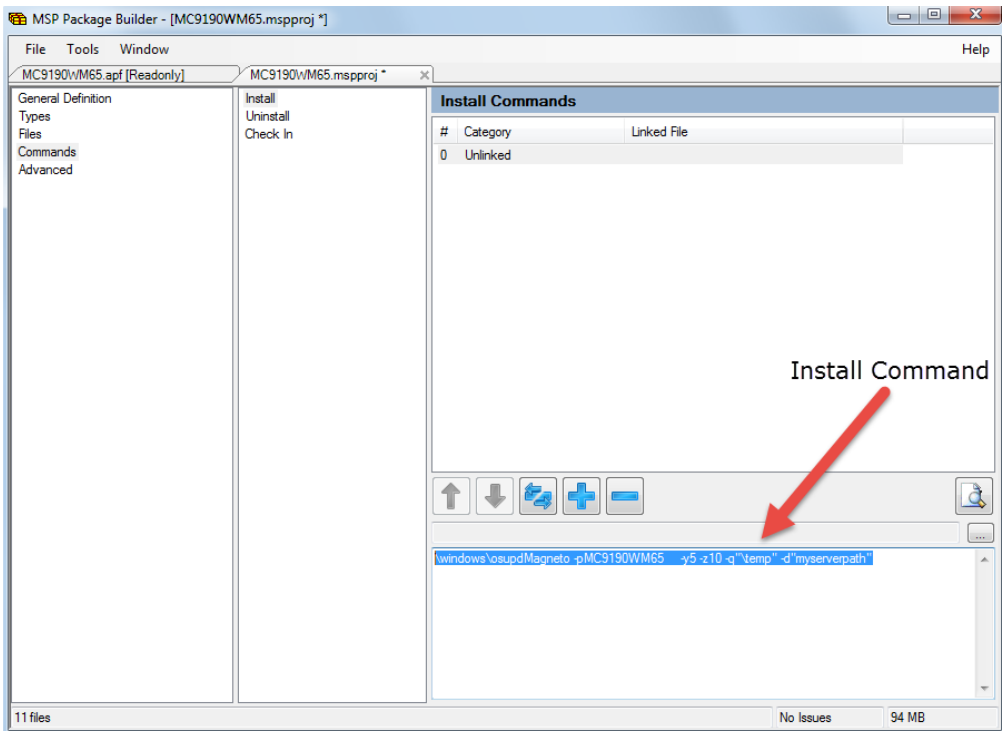
   - **Name** – Enter the name for the OS Upgrade project.
   - **Extract FilesTo** – Enter the location the files should be saved to or select the Browse button to select the location.

5 Select **OK** to close the Convert to Project dialog box.

6 Select **Command** from the left window pane.

7 Select **Install** and copy the install command from the bottom right window pane.

Consider copy-pasting this install command to a file (Notepad, Word, and so on) so you can access it later while creating the File/Action for the OS Upgrade.



Create an OS Upgrade File-Action

Once you extract the OS upgrade files from the APF file, create an OS Upgrade file-action. If you must update the device registry, create a separate files-actions for the Registry Edit file and list that Files-Action first in the product manifest.

1 Navigate to **Devices > Provisioning > Components > Files/Actions** and select the **Add Files/Actions** button.

2 Select the **Windows Rugged** platform.

3 Complete the General text boxes.

   a Enter a **Name**.

   b Enter a **Description**.

   c View the pre-populated **Version** number of the file-action.

   d View and confirm the **Platform** for the file-action.

   e Enter who the files-actions are **Managed By**.

4 Select the **Files** tab and select **Add Files**.

5 Select **Choose Files** and upload the extracted OS update files.

6 Specify the **Download Path** `\[directory]\[full file name]`

7 Enable **Store OS Update Files on Relay Server** to ensure that OS Update files remain on the Relay Server until needed by the OSUpdate utility.

8 Select **Save** and add the file and return to the **Files** tab.

9 Select **Add Files** and add additional files.

10 Upload the OS Update Utility EXE and the "package.TXT" manifest file.

Windows Mobile devices always use a file named "osupdMagneto.exe." For Windows CE devices, the filename varies with the device.

Windows CE devices also require a package .APD. The contents of this file are not important and can be a simple text format file containing something benign. It can even be empty.

11 Specify separate **Download Paths** for both the OS Update Utility and the package .TXT manifest file.

The **Download Path** for the OS Update utility and the package.TXT manifest file must be: "\Windows\".

The Windows CE .APD file must be placed in: "\Application\AirBeam\PKG."

Important Do not select **Relay Server Only**. These files must be delivered directly to the device.

12 (Optional) Create a files/actions to download and install the REG file to the device.

This step is only necessary if the registry settings are not already correctly updated on the device.

Download the REG file to the '\Temp' directory specifying the full filename. On the manifest tab, add an **Install** action type and specify the directory location and the filename of the REG file to be run on the device.

13 Once all the files have been uploaded, select the **Manifest** tab and select **Add Action** to add **Install Actions**.

14 Select the **Run** action in the **Action Types** text box.

15 Copy the Install Command from the MSP Package Builder utility and paste it into the **Command Line and Arguments to Run**.

You must edit the syntax of the Install Command to match Workspace ONE ™ UEM run syntax:

- Windows Mobile
  - You must use double quotes around "\windows\osupdMagento" and add the .exe to the end of osupdMagento.

- -d"myserverpath" must be updated in the following format: "[path of your relay server]/PFILES/[file action name]_version/". If the directory on the relay server that you are using for OS Update is /Motorola/OSUpdateFiles, the name of the OS Update file/action is "OSUpdate" and the "OSUpdate" file/action is on version 5, then the -d argument might look like: -d"/Motorola/OSUpdateFiles/PFILES/OSUpdate_5/".
  - The full run command syntax with all the arguments looks like the following.

```
\windows\osupdMagneto.exe" -p"MC9190WM65" -o -y5 -z10 -q"\temp\" -d"/Motorola/
OSUpdateFiles/PFILES/OSUpdate_5/
```

- Windows CE
  - You must use double quotes around "\windows\ [OS Update Utility filename]" and add the .exe to the end of the filename.
  - -d"myserverpath" must be updated in the following format: "[path of your relay server]/PFILES/[file action name]_version/". If the directory on the relay server that you are using for OS Update is /Motorola/OSUpdateFiles, the name of the OS Update file/action is "OSUpdate" and the "OSUpdate" file/action is on version 5, then the -d argument might look like: -d"/Motorola/OSUpdateFiles/PFILES/OSUpdate_5/".
    - The full run command syntax with all the arguments looks like the following.

```
\windows\[OS Update filename].exe" -p"WT41N0" -o -y5 -z10 -d"/Motorola/
OSUpdateFiles/PFILES/OSUpdate_5/
```

16 Select **Save**.

Next Step: Create a Product that includes the OS Upgrade File-Action you just made and assign it to all applicable devices.

Results: Once the device receives the product, the OS Update process initiates. A notification screen displays when the upgrade is complete.

Product Sets

Occasionally there are conflicting products provisioned to devices due to similar grouping in smart groups and custom attributes in Workspace ONE UEM. Product Sets allow you to group conflicting products and rank the products based on business needs.

Product Sets Basics

Product sets contain multiple products that you want to keep mutually exclusive. Product sets are useful for situations where the products contained inside the product set are meant only for specific devices within the parameters set by the rules engine using custom attributes.

The products in the product set follow a hierarchy based on ranking according to business needs. From a given product set, a device receives only one product that applies to the device. This product is the highest ranked among devices that meet the smart group and custom attribute rules criteria. After a device receives a product from a product set, the device will not receive any other products from the set unless the rank of a subsequent product is elevated or a new product is created in the set with a higher rank.

Important A product must exist as either a standalone product or as part of a product set. The product set ensures the integrity of mutual exclusivity of products for a given device.

Create a Product Set

Create a product set to control the delivery of multiple products so a device receives only the specific product that applies to the device based on your business rules.

1 Navigate to **Devices > Provisioning > Product Sets** and select the **Add Product Set** button.

2 Select the platform for which you want to create the product set.

3 Complete the **General** text boxes.

Settings	Descriptions
Name	Enter a name for the product sets. The name cannot be longer than 255 characters.
Description	Enter a short description for the product sets.
Managed By	Select the organization group that can edit the product sets.

4 Select the **Products** tab.

5 Select **Add** to add products to the product set.

6 Create a product including manifest items, conditions, and deployment settings.

See Create a Product for more information on creating a product. Ensure that you use the rules engine to create custom attribute-based rules for each product so the policy engine can properly assign the products.

7 Use the **Up** and **Down** arrows to adjust product ranking based on business needs.

8 Set products to **Active** if needed.

9 Select **Save** to create the product set.

Add a Product to a Product Set

You can add a product to an existing product set. This action requires following specific rules due to the complicated relationship between products and business rules.

A new product in a product set is added with the lowest ranking in the set by default. If the new product should be a higher rank, you must edit the ranking. For more information on what happens when product ranks are adjusted, see the section on this page entitled **Change the Product Ranking in a Product Set**.

1 Navigate to **Devices > Provisioning > Product Sets**.

2 Find the product set you want to add a product to and select the **Edit** icon ( ).

3 Select the **Products** tab.

4 Select **Add Product**.

5 Manually adjust the product rank as needed according to your business needs.

6 Select **Save** to add the product to the product set.

Results: Any modifications made during the edit of a product set do not take effect until you save the product set. Once saved, the product set enters the policy engine for evaluation.

Remove a Product from a Product Set

Remove a product from an existing product set. This action requires following specific rules due to the complicated relation between products and business rules.

Removing a product from a product set raises the rank of all products previously ranked below the removed product by one. If multiple products are removed, the remaining products are adjusted by one rank for each product removed. For more information on what happens when product ranks are adjusted, see the section on this page entitled **Change the Product Ranking in a Product Set**.

Any modifications made during the edit of a product set do not take effect until you save the product set.

1 Navigate to **Devices > Provisioning > Product Sets**.

2 Find the product set you want to remove a product from and select the **Edit** icon ( ).

3 Select the **Products** tab.

4 Select the check box for each product you want to remove from the product set.

5 Select the **Delete** button to remove the products.

6 Manually adjust the product rank as needed according to your business needs.

7 Select **Save** to remove the product from the product set.

Results: Once saved, the product set enters the policy engine for evaluation.

Activate and Deactivate Products in a Product Set

When you activate or deactivate a product in a product set, you must be aware of and prepare for possible consequences.

1 Navigate to **Devices > Provisioning > Product Sets** and select the pencil icon ( ) to the left of the Product Set name in the listing.

The **Edit Product Set** screen displays.

2 Select the **Products** tab.

The full list of products in the product set displays including the green and red traffic light toggles. Products with a green light are active, products with a red light are inactive.

3 Select the traffic light toggle and switch between activated (green) and deactivated (red) products.

Results:

- Deactivating a product in a product set sends a removal command to all devices with that product, and the next highest ranked product is installed.
- Activating a product in a product set might trigger other products to be removed on devices, and the newly activated product to be installed.

View Product Sets in Device Details

Product Sets display on individual device detail pages to show the status of the products' deployments to the device. The products listed that are part of a product set display the product set they pertain to and the deployment status of the products.

The **Products** tab displays all the products in a product set that is assigned to a device. The status of the products in relation to the device is displayed as well. Not all the displayed products from a product set are applicable for the device viewed.

To see the product sets in the Device Details, navigate to **Devices > List View** and select the device you want to view. Then select the **More** option and select **Products**.

- **Product Set** – Displays the product set that contains the product. Select the product set to view the product set details.
- **Status** – Displays the status of the product. For products in a product set, the appropriate product deployed to the device is labeled as **Compliant**. The other products contained in the product set eligible for deployment but are not deployed to the device are labeled as **Outranked**. Any product that is not eligible to be deployed to the device is labeled as **Not Applicable**.

Change the Product Ranking in a Product Set

Product set ranking controls which product of a product set is sent to a device. Since the ranking is the key feature of product sets, changes in ranking cause a series of reactions in the product set.

The following are examples of rank changes and what happens to the product, product set, and devices as a result.

Table 4-16. Rank Changes

Reason for Edit	Effect of Edit
To add a product.	The new product is set at the lowest rank. You must manually change the rank of the new product as needed.
Changing rank of existing products	Increasing the rank (selecting **Up** arrow) of a product decreases the rank of all subsequent products by one. Decreasing the rank (selecting **Down** arrow) of a product increases the rank of previously lower-ranked products. After you complete the rank changes and save the product, the product set enters the policy engine for evaluation. The engine assesses the custom attribute for each device against the new device rankings. If you reorder the Products priority within a Product Set, then the Products are reassigned based on the new priority order. As a result, the Workspace ONE UEM console sends removal commands for all devices affected by the reorder and assign Products based on the new order. After editing product ranking, only the products affected by the new ranking receive removal and install commands. Products outside the change in ranking are not affected.
Removing a Product	Removing a product increases the rank of all products previously ranked below the deleted product by one. If multiple products were removed, the ranking increases by one for each product removed. All products that preceded the deleted product's rank remain unchanged. Any products that had the removed product installed receives a new product based on the new rankings.

1 Navigate to **Devices > Provisioning > Product Sets**.

2 Find the product set you want to add a product to and select the **Edit** icon ( ).

3 Select the **Products** tab.

4 Manually adjust the product rank as needed according to your business needs.

5 Select **Save** and apply the rank changes.

Custom Attributes

Custom attributes in Workspace ONE UEM enable you to extract specific values from a managed device (for example IMEI, location, among many others) and use it as assignment criteria for products. You can also configure a 3rd party application to create custom attributes and display them on the launcher.

What Is A Custom Attribute?

A custom attribute is a placeholder for additional device information collected by Workspace ONE Intelligent Hub or by a third party application. This placeholder can be used in many different ways.

- It can be used to assign content such as provisioned products.
  - *...for example, you can provision product XYZ to only devices that are checked out and in the field.*
- It can provide information to the admin on the UEM console or to the end user on the device.
  - *...for example, a delivery driver can view an in-house developed app to determine their next stop, furnished by a custom attribute that collects the location of the device*.
- It can be used to move newly enrolled devices to a specific organization group.
  - *...for example, you can move all newly enrolled devices whose model number equals Zebra VC80 to an organization group that is designed to serve that specific model.*

Note Custom attributes (and the rules generator) are only configurable and useable at Customer-level organization groups.

For details about available options regarding device assignment rules based on custom attributes, see Enable Device Assignments.

There are many ways to generate custom attributes, both within the UEM console and on the device itself.

How Do You Create Custom Attributes On the Device Using an XML File Generated by the UEM Console

In some use cases, you may need to create custom attributes on the device that are then transferred back to the Workspace ONE UEM console by way of the Workspace ONE Intelligent Hub.

You can do this by writing an XML file that generates the custom attribute. There are two ways the XML file can be placed on a device: 1) let a 3rd party application on the device write an XML file for you or 2) send the XML file to the device in a provisioned product.

This task allows you to send an XML file to a device in a provisioned product.

1 Create an XML file that defines your custom attributes and their values using the following example as a template.

```
<?xml version="1.0" encoding="utf-8"?>
<attributes>
  <attribute name="CustomAttribute1" value="12345" />
  <attribute name="CustomAttribute2" value="67890" />
  <attribute name="CustomAttribute3" value="abcde" />
</attributes>
```

2 Save this file as `CustomAttributeExample.XML`.

3 Navigate to **Devices > Provisioning > Components > Files/Actions** and select the **Add Files/ Action** button.

4 Select **Android** as the platform.

5 Complete the **Name** and **Description** text boxes in the **General** tab.

6 Select the **Files** tab and then select the **Add Files** button.

7 Browse to the location where you saved the `CustomAttributeExample.XML` file, select it, and then select the **Open** button.

8 Select **Save** to upload the XML file to the UEM console.

9 The **Download Path** text box must be completed in the following way.

- For **Zebra devices**, the download path must be `/enterprise/usr/attributes`
- For **non-Zebra devices**, the download path must be `/sdcard/Android/data/com.airwatch.androidagent/files/attributes`

10 Select **Save** to include the XML file and download path to the Files/Action component.

11 Select **Save** again to save the Files/Action itself.

12 Navigate to **Devices > Provisioning > Product List View** and select the **Add Product** button.

13 Select **Android** as the platform.

14 In the **General** tab, complete the **Name**, **Description**, and **Assigned Groups** options.

15 In the **Manifest** tab, select the **Add** button.

16 In the **Action(s) To Perform** drop down menu, select "File/Action - Install"

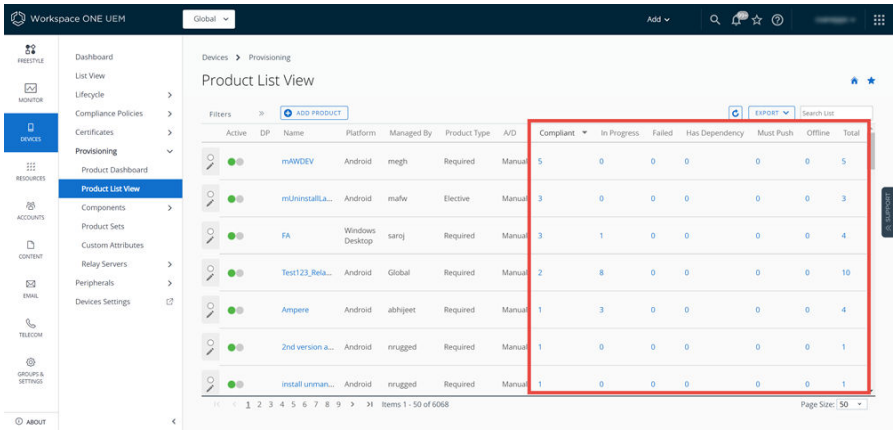
17 In the **Files/Actions** text box, select the name of the Files/Action you saved previously.

18 Select **Save** to save the Files/Action to the product.

19 Select **Activate** to display a preview of devices scheduled to be provisioned with the product.

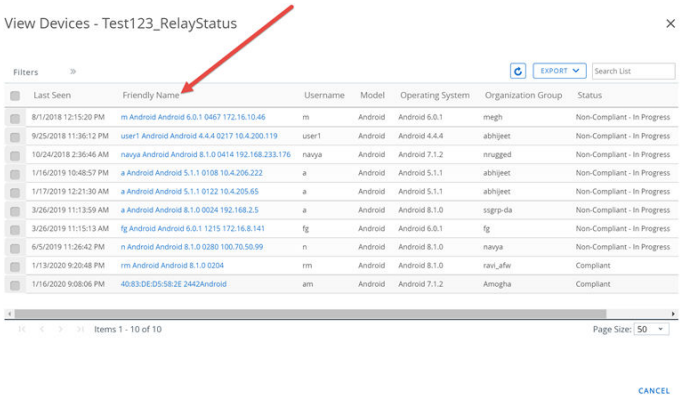
20 Select **Activate** again to finalize provisioning.

21 Navigate to **Devices > Provisioning > Product List View** and locate your product from the listing. Notice the 6 columns in the listing that display the various statuses of the devices that are provisioned with this product that contains your custom attribute.



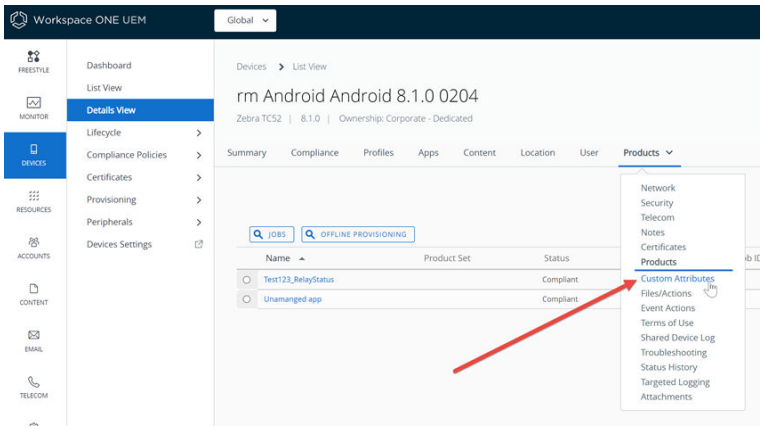
22 Select the number in the **Total** column for your product.

The **View Devices** screen displays, containing all the device scheduled to be provisioned by the product.



23 Select a device's Friendly Name from the listing.

The device **Details View** displays, showing you the activated products.



24 Select **Products > Custom Attributes** from **Details View**.

Verify that the custom attributes added from the XML file are displayed in the listing. If the custom attributes are not listed, check the device in to the console and repeat steps 21 - 24 again.

How Do You Create Custom Attributes On the Device Using an XML File Generated by a 3rd Party App

In some use cases, you may need to create custom attributes on the device that are then transferred back to the Workspace ONE UEM console by way of the Workspace ONE Intelligent Hub.

You can do this by writing an XML file that generates the custom attribute. There are two ways the XML file can be placed on a device: 1) send the XML file to the device in a provisioned product or 2) let a 3rd party application on the device write an XML file for you.

This task allows a 3rd party application sent to the device to write an XML file that creates a custom attribute on the device.

1 Work with your developer to create an application that has the ability to generate an XML file of the following structure.

```
<?xml version="1.0" encoding="utf-8"?>
<attributes>
  <attribute name="CustomAttribute1" value="54321" />
  <attribute name="CustomAttribute2" value="09876" />
  <attribute name="CustomAttribute3" value="edcba" />
</attributes>
```

2 The file must be written according to the following OEM-specific locations.

- For **Zebra devices**, the storage path must be `/enterprise/usr/attributes`
- For **non-Zebra devices**, the storage path must be `/sdcard/Android/data/com.airwatch.androidagent/files/attributes`

3 Create a product that includes this application and provision it to the devices you want the custom attribute to be written to.

4 Activate this product and wait for the device to check in to the console.

5 Navigate to **Devices > List View** and select the Friendly Name from the listing of one of the devices that is provisioned with your 3rd party application.

The device **Details View** displays.

6 From **Details View**, select **Products > Custom Attributes**.

The custom attributes you defined in the XML code should appear in the list.

How Do You Create a Custom Attribute and Assign it to a Single Device

Create a custom attribute and values to push to a device in Workspace ONE UEM. You can create assignment rules for products to provision based on these attributes and their values.

1 If you are not already in a customer type organization group (OG), move to one now. You cannot create custom attributes in any other kind of OG. For more information, see Organization Group Type Functions.

2 Navigate to **Devices > Provisioning > Custom Attributes**.

3 Select **Add** and then select **Add Attribute**.

4 Under the **Settings** tab, enter an **Attribute Name**.

5 Enter the optional **Description** of what the attribute identifies.

6 Enter the name of the **Application** that gathers the attribute. The application can be a third-party app or Workspace ONE Intelligent Hub.

   If you are not using a third-party app to generate the custom attribute, then use any name with an XML extension. For example `TruckRegCA.xml`

7 Select **Collect Value for Rule Generator** to make the values of the attribute available in the drop-down menu of the rule generator.

8 Select **Use in Rule Generator** if you want to use the attribute in the rule generator.

9 Select **Persist** to prevent the removal of the custom attribute from the Workspace ONE UEM console unless an Admin or an API call explicitly removes it.

   Otherwise, the attribute is removed as normal. If you delete a custom attribute reported from a device to the UEM console, a persisted custom attribute remains in the UEM console. Custom attribute persistence is only available to Android and Windows Rugged devices.

10 Select **Use as Lookup Value** to use the custom attribute as a lookup value anywhere in the UEM console.

   This option is important if you want the custom attribute to be available to Launcher or other tools.

11 Select the **Values** tab.

12 Select **Add Value** to add values to the custom attribute.

   You do not need to enter all possible values of the attribute. The list of attributes entered here is not a requirement or constraint on what values the device *can* report. Instead, enter only expected values used to pre-define organization group assignment rules.

13 Select **Save**.

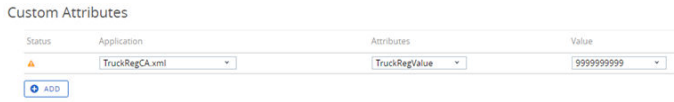
14 Navigate to **Devices > List View**, search for the device you want to assign the custom attribute to, and select its Friendly Name to display **Details View**.

15 Select the More tab, then select **Custom Attributes**, followed by the **Add** button.

The **Custom Attributes** screen displays.

16 Select the **Add** button on the **Custom Attributes** screen.

17 In the **Application** column drop-down, select the application name you entered in step 6. Then select the **Attributes** and **Value** in their respective drop-downs.



18 Select **Save**.

You can now use this custom attribute in the UEM console and in Launcher, in the following form.

```
{{CustomAttributeName::ApplicationName}}
```

Continuing the example from this task, the custom attribute looks like this in the Launcher config.

```
{{TruckRegValue::TruckRegCA.xml}}
```

For more information, see the section in this topic entitled **Display Custom Attributes in Launcher**.

How Do You Import Custom Attributes and Assign Them to Specific Devices

The custom attribute batch import feature in Workspace ONE UEM allows you to load custom attributes and corresponding values into the system in bulk. In the templates provided, each column corresponds to one custom attribute and each row corresponds to their different values or devices.

With the templates, you can import custom attributes in different ways and with different information.

Caution The syntax of the first column of each template must be replicated exactly. Failure to use the proper syntax can cause database issues and result in loss of data.

In order to import custom attributes and assign them to specific devices, you must import two different CSV files, 1) one that defines the custom attributes and 2) another that defines the value of the custom attribute and assigns it to devices.

1 Navigate to **Devices > Provisioning > Custom Attributes** and select the **Add** button followed by **Batch Import**.

The **Batch Import** screen displays.

2 Scroll down to where the template links can be seen and select two of the provided links to download both templates and save them to your device.

- **Custom Attributes** – Download template for this batch type
- **Device Custom Attribute Values** – Download template for this batch type

You can either leave the **Batch Import** screen displayed while you complete the process of filling out the templates OR you can select the **Cancel** button, complete the templates, and return to the console later.

3 Browse your device and locate the two CSV files. Open the one titled **Template - CustomAttributes.csv** and fill in the fields that correspond to the headers.

4 Leave the **Value** column blank.

5 For the **ApplicationName** column, if you are not using an actual app on the device to generate the custom attribute, just enter any name with an XML extension, for example, SampleAppName.xml

6 The **ShowOnDevicesGrid** column corresponds to the **Use As Lookup Value** option. If you plan to display these custom attributes on the Launcher screen, the records for each attribute must be 1 in this column to enable the option.

7 Save the CSV file as `SampleCustomAttributes.CSV`.

8 Open the other template you downloaded in step 2, titled **Template - CustomAttributeValuesXref.csv**.

9 You must customize this template differently than the first template. The value entered in column A, **XRefType**, defines for the console how you want to identify devices, according to the following list.

   1. DeviceID
   2. Serial Number
   3. UDID
   4. MAC Address
   5. IMEI Number

10 In column B, **XRefValue**, supplies the value that corresponds to column A.

   For example, if you enter 2 for each row in the *XRefType* column A, then you must enter the *serial number* for each device that accepts a custom attribute in the XRefValue column B.

11 For column C, change the header using the format `ApplicationName||CustomAttributeName`, according to your actual custom attribute designations. When you fill out the values in column C, you are assigning values to the custom attribute that is specific to the device identified in column B.

12 Add subsequent columns for each custom attribute you want to assign to devices. Here is a sample completed Device Custom Attribute Values template. The text colored red are the portions you must customize.

	A	B	C	D	E
1	XRefType	XRefValue	SampleAppName.xml||CA_1	SampleAppName.xml||CA_2	SampleAppName.xml||CA_3
2		2 VMrkQarWWq2q	ca1_ab	ca2_ab	ca3_ab
3		2 ZY22524JCP	ca1_cd	ca2_cd	ca3_cd
4		2 KR37fhUd34oL	ca1_ef	ca2_ef	ca3_ef
5		3 f269c8c535fc8a1efc089d4efa68	ca1_gh	ca2_gh	ca3_gh
6		3 b09a02fe8690dc57d944b662d41	ca1_ij	ca2_ij	ca3_ij
7		3 gj84jfk94jhgkld947632jf947jfd4	ca1_kl	ca2_kl	ca3_kl
8		4 00:50:56:81:23:36	ca1_mn	ca2_mn	ca3_mn
9		4 2c:0e:3d:10:5d:94	ca1_op	ca2_op	ca3_op
10		4 00:34:5k:53:11:79	ca1_qr	ca2_qr	ca3_qr
11		5 358991077927844	ca1_st	ca2_st	ca3_st
12		5 586947630847859	ca1_uv	ca2_uv	ca3_uv
13		5 562047586970943	ca1_wx	ca2_wx	ca3_wx

Note the different device identifiers in column A and how column B changes with it. You are free to use a single device identifier for the entire template. This is just an example that demonstrates the relationships.

13 Save the CSV file as `SampleCustomAttributeValues.CSV`.

14 Navigate back to **Devices > Provisioning > Custom Attributes** and select the **Add** button followed by **Batch Import**.

15 Enter **Batch Name**, **Description**, **Batch Type** as 'Custom Attributes', then select the **Choose File** button.

16 Locate your previously saved `SampleCustomAttributes.CSV` file (from step 7) and select it to upload.

17 Select **Save** to complete the batch import process.

18 Navigate to **Devices > Provisioning > Custom Attributes** and review all the newly imported custom attributes.

19 If you want to use these custom attributes in Launcher, select the pencil edit icon ( ) for each custom attribute and enable the **Use As Lookup Value** check box.

20 Repeat steps 14 through 17 for the other CSV file (`SampleCustomAttributeValues.CSV`) you saved in step 13.

Result: You can display any custom attribute (with the **Use As Lookup Value** option enabled) in the console and Launcher using the `{{CustomAttributeName::ApplicationName}}` format. For more information, see How Do You Display Custom Attributes in Launcher (Multi App Mode) and How Do You Display Custom Attributes in Launcher (Template Mode).

How Do You Display Custom Attributes in Launcher (Multi App Mode)

1 If you are not already in a customer type organization group (OG), move to one now. You cannot assign custom attributes in any other kind of OG. For more information, see Organization Group Type Functions.

2 Navigate to **Resources > Profiles & Baselines > Profiles** and select the **Add** button followed by **Add Profile**.

3 Select either **Android** or **Android (Legacy)**. Launcher is available for each, however the Launcher configuration layout is slightly different.

The **Add a New Android Profile** screen displays with the payload listing on the left side panel.

4 Select the **Launcher** payload.

5 Select the **Configure** button.

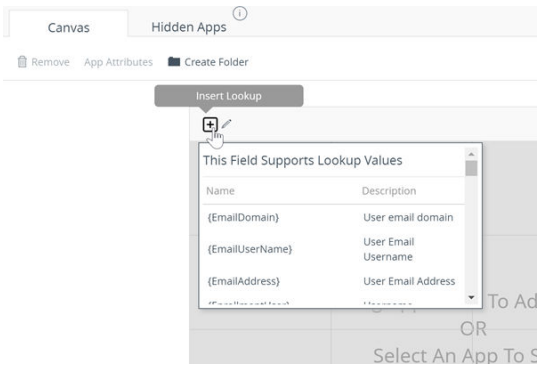
The **Select App Mode** screen displays.

6 Select the **Multi App** mode, which allows you to configure a collection of approved apps on the home screen for your end users to access. You can also customize the layout including company branding and more.

A self directed tutorial appears. Acquaint yourself with each step to see what the Launcher Multi App can offer, including layout, apps, the canvas, app level actions, hidden apps, settings, and the preview feature.

7 You can insert the following standard lookup values into the title bar of the launcher shown below and also to section headings.

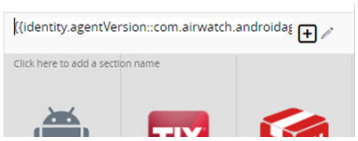
- {DeviceAssetNumber}
- {DeviceFriendlyName}
- {DeviceModel}
- {DeviceOperatingSystem}
- {DeviceSerialNumber}
- {DeviceSerialNumberLastFour}
- {DeviceUid}
- {DeviceUidLastFour}
- {DeviceUuid}
- {DeviceWLANMac}
- {EnrollmentUser}
- {FirstName}
- {GroupIdentifier}
- {LastName}



8 You can display your own custom attributes on the Launcher screen in place of lookup values, provided you insert them onto the canvas using the following format.

```
{{CustomAttributeName::ApplicationName}}
```

Get the Application Name and Custom Attribute Name by navigating to **Devices > Provisioning > Custom Attributes**.



You can also add custom attributes to section headers in the same way, using the same `{{CustomAttributeName::ApplicationName}}` format. You are limited to one custom attribute per section header and the font cannot be adjusted. Here is an example of a configured Launcher Multi App layout and how it looks on the device.

How Do You Display Custom Attributes in Launcher (Template Mode)

Template Mode is the fully customizable mode of Workspace ONE Launcher. You can add apps, images, text boxes, backgrounds, and other layout settings to customize the device lock down screen. Text fields in the Launcher Template layout can display lookup values including custom attributes.

1 If you are not already in a customer type organization group (OG), move to one now. You cannot assign custom attributes in any other kind of OG. For more information, see Organization Group Type Functions.

2 Navigate to **Resources > Profiles & Baselines > Profiles** and select the **Add** button followed by **Add Profile**.

3 Select either **Android** or **Android (Legacy)**. Launcher is available for each, however the Launcher configuration layout is slightly different.

The **Add a New Android Profile** screen displays with the payload listing on the left side panel.

4 Select the **Launcher** payload.

5 Select the **Configure** button.

The **Select App Mode** screen displays.

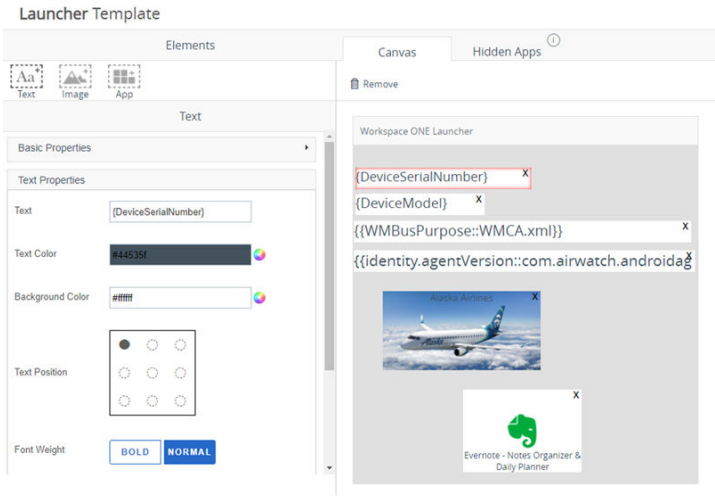
6 Select the **Template** mode, which offers the greatest amount of flexibility in designing the appearance of Launcher.

A self directed tutorial appears. Acquaint yourself with each step to see what the Launcher Template can offer, including layout, apps, the canvas and its images and text boxes it can display, app level actions, hidden apps, settings, and the preview feature.

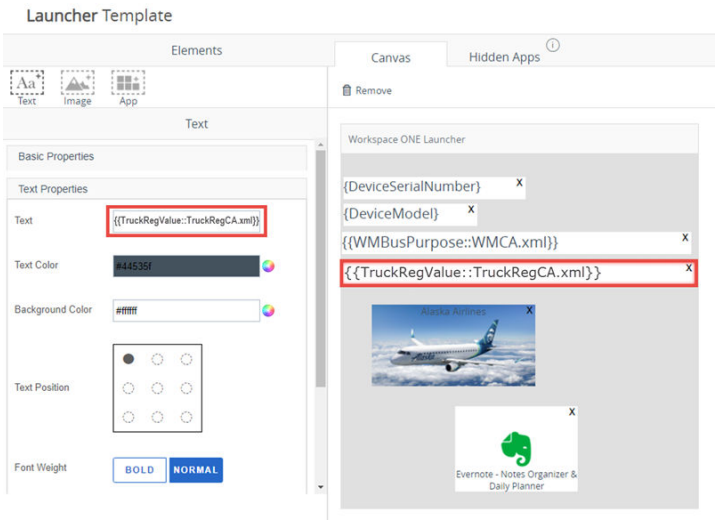
7 In the **Launcher Template** screen, drag **Elements** (Text, Image, and App) from the left side and drop them onto the Launcher **Canvas**, completing options such as **Text Color**, **Backgroud Color**, **Text Position**, and so on.

8 You can insert the following standard lookup values into Text Boxes that you create on the canvas.

- {DeviceAssetNumber}
- {DeviceFriendlyName}
- {DeviceModel}
- {DeviceOperatingSystem}
- {DeviceSerialNumber}
- {DeviceSerialNumberLastFour}
- {DeviceUid}
- {DeviceUidLastFour}
- {DeviceUuid}
- {DeviceWLANMac}
- {EnrollmentUser}
- {FirstName}
- {GroupIdentifier}
- {LastName}



9 Provided you enabled the **Use as Lookup Value** option when you made your custom attributes, you can also display these custom attributes onto the Launcher canvas, using the `{{CustomAttributeName::ApplicationName}}` format.



Note Custom Attribute values cannot return the following special characters: **/ \ " * : ; < > ?** |. If a script returns a value which contains these characters, the value is not reported on the console. Trim these characters from the script's output.

How Do You Assign Organization Groups Using Custom Attributes

Configure rules that control how devices are assigned to organization groups following enrollment in Workspace ONE UEM. You are limited to one custom attribute assignment rule per organization group (OG).

1 Ensure that you are currently in a customer type organization group.

2 Navigate to **Groups & Settings > All Settings > Devices & Users > General > Advanced**.

3 Set **Device Assignment Rules** to **Enabled**.

4 Set the **Type** to **Organization Group by Custom Attribute**.

For more information about device assignment rules, see Enable Device Assignments.

5 Select **Save**.

6 Navigate to **Devices > Provisioning > Custom Attributes > Add > Add Attribute** and create a custom attribute if you have not already done so.

See the section on this page entitled **Create a Custom Attribute**.

7 Navigate to **Devices > Provisioning > Custom Attributes > Custom Attributes Assignment Rules > Add Rule**.

8 Select the **Organization Group** to which the rule assigns devices.

9 Select **Add Rule** to configure the logic of the rule.

Setting	Description
Attribute/ Application	This custom attribute determines device assignment. Select from among Device Model, Serial Number, and any custom attribute or XML file that is available in the customer OG you are in.
Operator	This operator compares the **Attribute** to the **Value** to determine if the device qualifies for the product. When using more than one Operator in a rule, you must include a **Logical Operator** between each **Operator**. **Note** When making an assignment rule, comparisons using the less than (<) and greater than (>) operators (and their variants) can only be used to compare numerical values including integers. The exception is when you are comparing OEM build versions, you can apply < and > operators on non-numerical ASCII strings. An example is when an OEM update filename includes hyphens, periods, and other characters together with numbers. Such assignment rules must identify a device manufacturer in the rule logic and that comparison is deemed accurate when the format on the device matches the one specified on the server.
Value	All values from all applicable devices are listed here for the **Attribute** selected for the rule.
Add Logical Operator	Select to display a drop-down menu of logical operators such as AND, OR, NOT, and parentheses. Allows for more complex rules.

10 Select **Save** after configuring the logic of the rule.

Results: When a device enrolls with an assigned attribute, the rule assigns the device to the configured organization group.

Custom Attributes Database

Custom attributes are stored as XML files and in the Workspace ONE Intelligent Hub database, each stored on the device. When using the database, custom attributes are sent as samples to Workspace ONE UEM periodically for asset tracking of key/value pairs.

If a record in the device database is configured with 'Create Attribute' = TRUE, then the Workspace ONE Intelligent Hub automatically retrieves the Name and Value sent with the custom attributes sample. The key/value pair displays in the Device Details page for the device in the Custom Attributes tab.

Custom Attributes, Android

XML provisioning can collect custom attributes based on your Android device details. You can use custom attributes for advanced assignment criteria for product provisioning, as lookup values, and for automatically moving devices to organization groups based on these attribute values.

Third Party Custom Attributes XML for Android Devices

In Android 11, customers using third party custom attributes must use the Custom Settings profile to specify an alternate location for storing custom attribute files. Customer apps also must target this same folder path, which may require changes to their app.

Example Custom XML (edit per your preferences):

```
<characteristictype="com.android.agent.miscellaneousSettingsGroup"uuid="2c787565-1c4a-4eaa-8cd
4-3bca39b8e98b"><parmname="attributes_file_path"value="/storage/emulated/0/Documents/
Attributes"/></characteristic>
```

Add a Custom Attribute

Custom Attributes on the Android platform can be used in two basic scenarios.

- **Simple XML File Push (Tags)** – You push an XML file with a pre-defined attribute value and the Workspace ONE Intelligent Hub reports it back to the console, functioning like a device tag. You can build these tags into products. While performing other actions, the product then adds an attribute that tells the console that those actions have been taken. You can then use these values for product assignment, OG assignment, and lookup values.
- **Hub Directory Save** – You can code an internally-developed application that saves an XML file with the attribute values to the Workspace ONE Intelligent Hub directory. The Hub then reports these values back to the console. The benefit of this scenario over the first scenario (Tags) is that, because you are using the Workspace ONE Intelligent Hub, which collects information from other apps, you have access to information you would not ordinarily have.

1 Navigate to **Devices > Provisioning > Components > Files/Actions > Add** and select **Android** as your platform.

2 Create an XML provisioning file.

   - **For Zebra devices** – See Create an XML Provisioning File, Android, Win7, WinRugg. The manifest must include an action to download the XML file in the following path `/enterprise/usr/attributes`
   - **For non-Zebra devices** – the XML file location is `/sdcard/Android/data/com.airwatch.androidagent/files/attributes/`

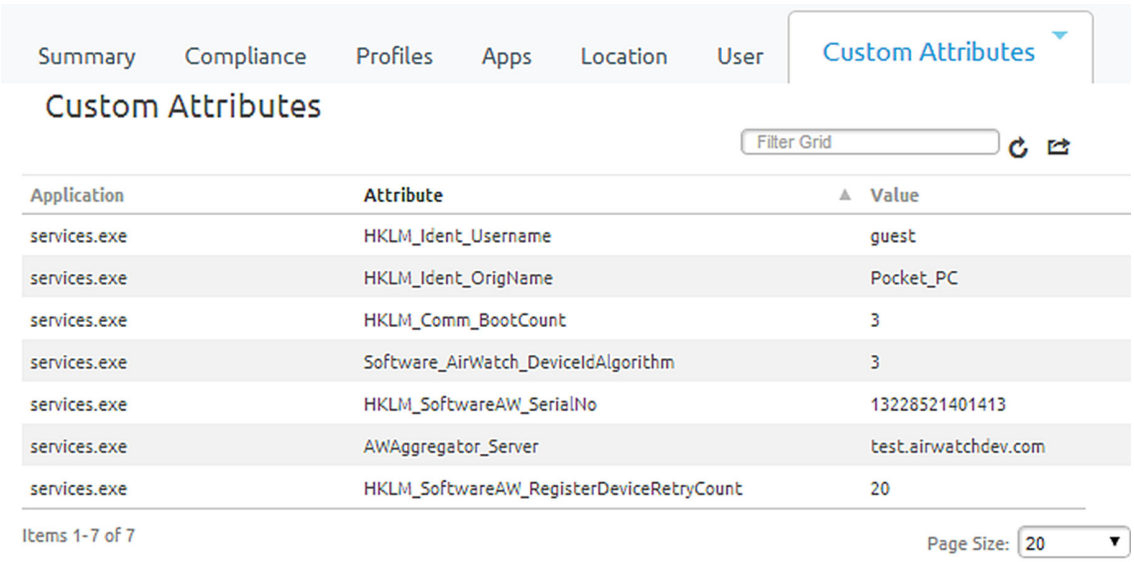
Results: Upon receiving the XML file, the Workspace ONE Intelligent Hub for Android creates a custom attributes output file.

During the next check-in with AirWatch, the Workspace ONE Intelligent Hub sends the output file to the Workspace ONE UEM console.

Once the XML file installs, the custom attributes requested in the file exported to the console. These values display in the console in the Device Details page under custom attributes. The Device Details page enables you to view the name of the attribute and the values returned from each device. These values can be used to create product assignment rules using the Custom Rules system.

Example:

```
<?xml version="1.0"?>
<attributes>
    <attribute name="attribute 1" value="value 1"/>
    <attribute name="attribute 2" value="value 2"/>
    <attribute name="attribute 3" value="value 3"/>
</attributes>
```



What to do next: You can also view existing custom attributes for all devices at a particular organization group and manually create custom attributes directly in the console. Navigate to **Devices > Provisioning > Custom Attributes > List View**. Any custom attribute created in this manner automatically associates with a device and its respective custom attribute value that is successfully transmitted to the console.

Custom Attributes, macOS

You can use custom attributes to unlock product provisioning's more advanced functionality for your macOS device. Write a command or script and report it as a custom attribute using the Workspace ONE Intelligent Hub for macOS v.2.3 or later. Select when to run the command or script on hourly intervals or during an event.

Custom Attributes can also be used in Assignment Rules for Products.

Procedure

1. Navigate to **Resources > Profiles & Baselines > Profiles** and select **Add > Add Profile**. Select Apple macOS, and then select **Device Profile**, because this profile is only applicable to the device, not the user.
2. In the left panel, select the **Custom Attributes** profile and select the **Configure** button.
3. Enter the **Attribute Name**.
4. Enter the **Script/Command** to run. Expand the text box as needed.
5. Select an **Execution Interval** and allow for scheduling to report either in hours or as an event occurs.
6. Use the **+** and **-** buttons at the bottom of the payload to create multiple scripts.
7. Select **Save & Publish** when you are finished to push the profile to devices.

> **Note** Custom Attribute values cannot return the following special characters: **/ \ " * : ; < > ? |**. If a script returns a value which contains these characters, the value is not reported on the console. Trim these characters from the script's output.

Results

The script included in the profile runs on the device to gather the values for each attribute.

Example: Key-Value Pair Examples

The following is an example of commands you can use for creating, removing, or modifying key-value pairs. Use these commands to dynamically change the values for a custom attribute on the device.

Add a Key-Value Pair.

```
/usr/libexec/PlistBuddy -c "Add :ASSET_ID string '1'" "/Library/Application Support/AirWatch/
Data/CustomAttributes /CustomAttributes.plist"
```

Delete a Key-Value Pair.

```
/usr/libexec/PlistBuddy -c "Delete :ASSET_TAG" "/Library/Application Support/AirWatch/Data/
CustomAttributes/ CustomAttributes.plist"
```

Modify a Key-Value Pair.

```
/usr/libexec/PlistBuddy -c "Set :ASSET_ID '2'" "/Library/Application Support/AirWatch/Data/
CustomAttributes/ CustomAttributes.plist"
```

What to do next

You can view existing custom attributes for all devices at a particular organization group and manually create custom attributes directly in the console.

Navigate to **Devices > Provisioning > Custom Attributes > List View**. Any custom attribute created in this manner automatically associates with a device and its respective custom attribute value that is successfully transmitted to the console.

Custom Attributes, Win7

XML provisioning can collect custom attributes based on your Win7 device details. You can use custom attributes to unlock product provisioning's more advanced functionality.

Procedure

1 Navigate to **Devices > Provisioning > Components > Files/Actions > Add** and select **Windows 7** as your platform.

2 Create an XML product.

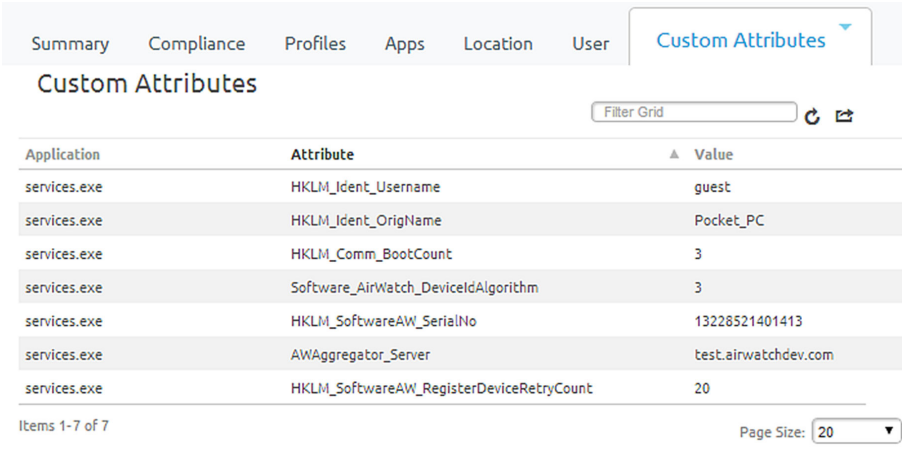
For more information, see Create an XML Provisioning File, Android, Win7, WinRugg. The manifest must include an action to download the XML file to **{installation path}\AirWatch\AgentUI\Cache\Profiles**.

Results

Upon receiving the XML file, the Workspace ONE Intelligent Hub for Windows 7 creates a custom attributes output file. During the next check-in with Workspace ONE ™ UEM, the Workspace ONE Intelligent Hub sends the output file to the UEM console.

Once the XML file installs, the custom attributes requested in the file exported to the UEM console. These values display in the console in the Device Details page under Custom Attributes. The Device Details page enables you to view the name of the attribute and the values returned from each device. These values can be used to create product assignment rules using the Custom Rules system.

Upon enrollment, the Workspace ONE Intelligent Hub for PC automatically generates the Device Model and Serial Number attributes.



Example: Syncing Registry Settings

To synchronize the registry settings on a Windows 7 device with the console, which is likely the most common use of custom attributes for Windows 7 devices, you must create a custom XML file.

This sample XML file pulls information from the registry on a device:

```
<xml version="1.0">
<wap-provisioningdoc name="desiredDocName /V_1">
    <characteristic type="com.windowspc.getregistryinfo.managed">
        <reg_value value_name="KeyName(i.e. CommonFilesDir)"
        key_name="RegistryPath(i.e.Software\Wow6432Node\Microsoft\Windows\
        CurrentVersion)"custom_attribute_name="AttributeName"/>
        <reg_value value_name="Keyname ..." key_name="Path\...."
        custom_attribute_name="AttributeName2"/>
    </characteristic>
</wap-provisioningdoc>
```

It must be in the previous format for the XML file to get correctly parsed and the registry settings to be outputted to a key value pair that can be exported back to the UEM console.

In this example, the registry key path is “HKEY_LOCAL_MACHINE\Ident” for two of the values and within that key path it is reading the values of “Username” and “OrigName”. The ‘custom_attribute_name’ parameter is simply the name of the custom attribute that displays in the console and corresponds to the value read from the device.

What to do next

You can also view existing custom attributes for all devices at a particular organization group and manually create custom attributes directly in the console. Navigate to **Devices > Provisioning > Custom Attributes > List View**. Any custom attribute created in this manner automatically associates with a device and its respective custom attribute value that is successfully transmitted to the console.

Custom Attributes, WinDesk

XML provisioning can collect custom attributes based on your Windows Desktop device details. You can use custom attributes to unlock product provisioning's more advanced functionality.

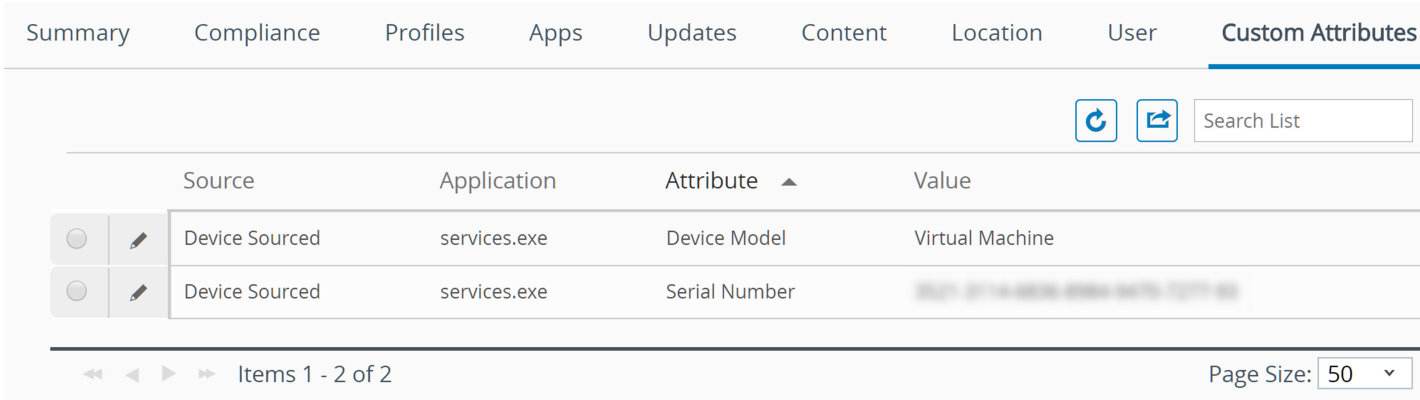
Procedure

1. Navigate to **Devices > Provisioning > Components > Files/Actions > Add** and select **Windows > Windows Desktop** as your platform.
2. Complete the steps to create an XML product as mentioned in Create an XML Provisioning File. Upload the XML file and specify the download path as **{installation path}\AirWatch\AgentUI\Cache\Profiles**.

Results

Upon receiving the XML file, the Workspace ONE Intelligent Hub creates a custom attributes output file. During the next check-in with Workspace ONE ™ UEM, the Workspace ONE Intelligent Hub sends the output file to the UEM console.

Once the XML file installs, the custom attributes requested in the file are reported to the UEM console. These values display in the console in the Device Details page under custom attributes. This page allows you to view the name of the attribute and the values returned from each device. These values can be used to create product assignment rules using the Custom Rules system.



Note Note: Custom Attributes support the HKLM registry hive only.

Example: Fetching Registry Settings

A common use of custom attributes for Windows devices is to fetch registry settings with the UEM console. To do this, you must create a custom XML file and deploy it using a custom settings policy targeted to the AirWatch Unified Agent. Here is an example of the format of an XML file that can pull information from the registry on a device.

Windows 10 Example

```
<xml version="1.0">
<wap-provisioningdoc name="System Info /V_1">
     <characteristic type="com.windowspc.getregistryinfo.managed">
          <reg_value custom_attribute_name="Hostname"
          key_name="SYSTEM\CurrentControlSet\Control\ComputerName\ActiveComputerName"
          value_name="ComputerName"/>
     </characteristic>
</wap-provisioningdoc>
```

The custom payload must be in the above format. When the correct syntax is used, the XML is parsed and the registry settings are outputted to a key value pair that are exported back to the Workspace ONE UEM console.

In the above example, the registry key name or path is “Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\ComputerName\ActiveComputerName” and the value_name is “ComputerName”. The data corresponding to this value name is retrieved and stored in the console with the label specified under the ‘custom_attribute_name’ parameter which is “Hostname” in the examples.

The final output of this configuration stores the computer name of the device in the Workspace ONE database with a key Hostname and a data value retrieved from the device. For example, Hostname -> JSMITH-PC01. This data can be viewed in the Custom Attributes tab on the Device Details view.

What to do next

You can also view existing custom attributes for all devices at a particular organization group and manually create custom attributes directly in the console. Navigate to **Devices > Provisioning > Custom Attributes > List View**. Any custom attribute created in this manner automatically associates with a device and its respective custom attribute value that is successfully transmitted to the console.

Custom Attributes, WinRugg

XML provisioning can collect custom attributes based on device details for your Windows Rugged device. You can use custom attributes to unlock product provisioning's more advanced functionality.

Procedure

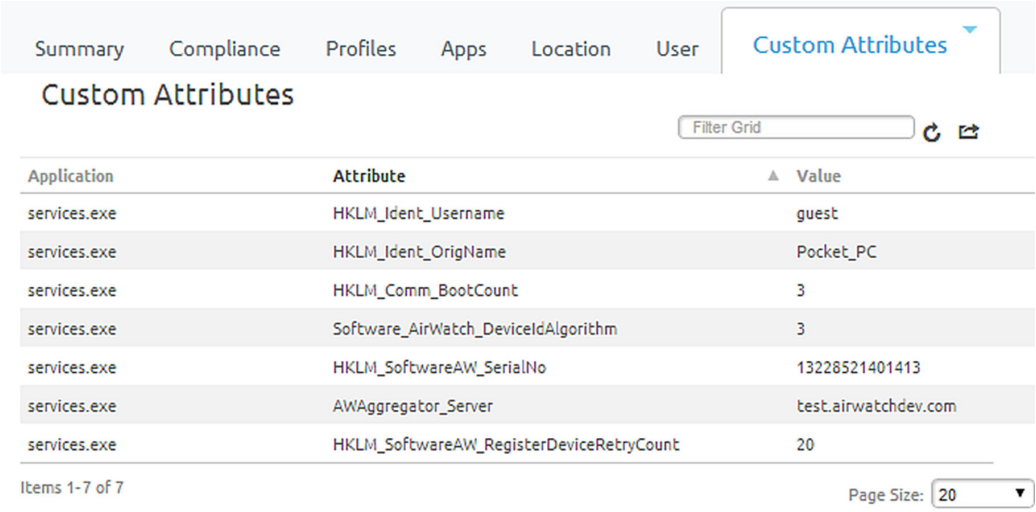
1. Navigate to **Devices > Provisioning > Components > Files/Actions**, then select the **Add Files/Actions** button, and then select **Windows Rugged** as your platform.
2. Create an XML product.

   For more information, see Create an XML Provisioning File, Android, Win7, WinRugg. The manifest includes an action to download the XML file to **\Program Files\Airwatch\Cache\Profiles**.

Results

Upon receiving the XML file, the Workspace ONE Intelligent Hub for Windows Rugged creates a custom attributes output file. During the next check-in with Workspace ONE UEM, the Workspace ONE Intelligent Hub sends the output file to the Workspace ONE UEM console.

Once the XML file installs, the custom attributes requested in the file are reported to the UEM console. These values display in the UEM console on the Device Details page under custom attributes. The Device Details page enables you to view the name of the attribute and the values returned from each device. These values can be used to create product assignment rules using the Custom Rules system.



Example

Sampling Registry Settings to Workspace ONE UEM

In order to sample the registry settings on a Windows Rugged device and collect them in the console, the most common use of custom attributes for Windows Rugged devices, you must create a custom XML file. Below is an example of the format of an XML file that can pull information from the registry on a device:

```
<?xml version="1.0"?>-<wap-provisioningdoc allowRemoval="True" name="GetTypicalRegValues/V_1"
id="5a63204f-848c-42d5-9c14-4ca070743920">
    -<characteristic uuid="f49a9cb5-48e9-47cd-84cc-ef122dcb5d50"
type="com.airwatch.getregistryinfo.winmo">
        <reg_value value_name="Username"
          key_name="HKEY_LOCAL_MACHINE\Ident"
          custom_attribute_name="HKLM_Ident_Username"/>
        <reg_value value_name="OrigName"
          key_name="HKEY_LOCAL_MACHINE\Ident"
          custom_attribute_name="HKLM_Ident_OrigName"/>
        <reg_value value_name="BootCount"
          key_name="HKEY_LOCAL_MACHINE\Comm"
          custom_attribute_name="HKLM_Comm_BootCount"/>
        <reg_value value_name="DeviceIdAlgorithm"
          key_name="HKEY_LOCAL_MACHINE\Software\AirWatch"
          custom_attribute_name="Software_AirWatch_DeviceIdAlgorithm"/>
    </characteristic></wap-provisioningdoc>
```

It must be in the previous format for the XML file to get correctly parsed and the registry settings to be outputted to a key value pair that can be exported back to the UEM console. In this example, the registry key path is “HKEY_LOCAL_MACHINE\Ident” for two of the values and within that key path it is reading the values of “user name” and “OrigName”. The ‘custom_attribute_name’ parameter is simply the name of the custom attribute that displays in the console and corresponds to the value read from the device.

Using Third-Party Applications to Create Custom Attributes

If you want to create custom attributes using a third-party application, you need that application to export an XML file with a key value pair to the **Program Files\AirWatch\Cache\CustomAttributes** directory on the device. Once an XML file with a key value pair is present in this directory, it is parsed by the Workspace ONE Intelligent Hub and included in the next interrogator sample. The XML key/value pair must be in the following format.

```
<?xml version="1.0"?>-<attributes>
    <attribute name="HKLM_Ident_Username" value="guest"/>
    <attribute name="HKLM_Ident_OrigName" value="Pocket_PC"/>
    <attribute name="HKLM_Comm_BootCount" value="1"/>
    <attribute name="Software_AirWatch_DeviceIdAlgorithm" value="3"/>
    <attribute name="HKLM_SoftwareAW_SerialNo" value="13233521403231"/>
</attributes>
```

‘Attribute name’ is the name of the attribute in the console while ‘value’ is the corresponding value that is associated with that attribute.

What to do next

You can also view existing custom attributes for all devices at a particular organization group and manually create custom attributes directly in the UEM console. Navigate to **Devices > Provisioning > Custom Attributes > List View**. Any custom attribute created in this manner automatically associates with a device and its respective custom attribute value that is successfully transmitted to the UEM console.

What Happens If You Change a Product, Component, or Condition?

When you change an active product in Workspace ONE UEM powered by AirWatch by editing one of its Components like an Event Action, Condition, or even a specific product option, problems can result. You can mitigate these problems if you deactivate the product **before** making changes, then reactivate the product after.

Product Inventory vs Product Option

When something changes in your product, the course you take depends entirely on what, specifically, has changed. It is helpful to think of product elements in terms of 'inventory' versus 'options'.

Product Inventory includes *content* like software and files. Application Install Manifests, Files Actions, and Event Actions are all considered product inventory because they are content that must be transferred to a device. Most of the time, changes to product inventory do not require that you Force Reprocess the product. When you change a product's inventory, Workspace ONE UEM queues the new job automatically.

Product Options include settings that affect the *provisioning* of the product. Settings like Per App VPN, Persistent Through Enterprise Reset, Conditions, and even making changes to the manifest order are all product options. Changes to product options require a Force Reprocess in order for the device to receive the latest configuration.

Changing a Product Option

After making your changes to product options, take the following steps.

1 Save the product.

2 Navigate to **Devices > Provisioning > Product List View** and locate the newly saved product from the listing.

3 Select the radio button to the left of the product listing.

Result: A new button cluster appears under the **Add Product** button.

4 Select the **More Actions** button.

5 Select **Force Reprocess**.

Result: A confirmation appears warning you that the Force Reprocess action resends Products to all assigned devices regardless of compliance status. The devices fully download and install every component of the Product manifest, even if it already exists on the device.

6 To proceed with the Force Reprocess, select **OK**.

Changing an Event-Action

When you change an Event-Action, only those Event-Actions that are currently a component of an **Active Product** can you affect the product to which the Event-Action belongs.

To edit the Event-Action, take the following steps.

1 Navigate to **Devices > Provisioning > Components > Event Actions**.

2 Locate the Event-Action you want to edit and select the edit icon ( ) to the left of the listing.

3 Proceed through the **Platform**, **Details**, **Conditions** and **Actions** tabs, making edits per your preferences.

4 On the **Summary** tab, select **Save**.

- If no **Warning** screen displays, then the Event-Action is not a current component of an active product.
- If a **Warning** screen displays with an Active Product name, then you must **Activate** or **Deactivate** the product shown.

Selecting **Activate** means that the product is effectively re-provisioned and distributed as an active job.

Selecting **Deactivate** resets the status of the product as inactive.

Changing a Condition

When you change a condition, the only time you can affect a product is when that condition is…

- …a direct component of that **Active Product**.

  OR

- …used in an Event-Action that is a component in that **Active Product**.

To edit the Condition, take the following steps.

1 Navigate to **Devices > Provisioning > Components > Conditions**.

2 Locate the Condition you want to edit and select the edit icon () to the left of the listing.

3 Proceed through the **Type** and **Details** tabs, making edits per your preferences.

4 On the **Details** tab, select **Save** when you are finished making changes.

   - If no **Warning** screen displays, then the Condition is not a current component of an active product.
   - If a **Warning** screen displays with an Active Product name, then you must **Activate** or **Deactivate** the product shown.

Selecting **Activate** means that the product is effectively reprovisioned and distributed as an active job.

Selecting **Deactivate** resets the status of the product as inactive.

Product Management

5

Manage products using the product provisioning management functionality built into Workspace ONE UEM powered by AirWatch. Product management uses the Products Dashboard, Products List View, and Device Details View to manage how devices use products.

Rugged devices have different device actions and options than consumer devices. Some actions, such as Remote Management, require additional configuration before using with devices. Products must be deactivated before most device actions work. You must also disable any components before using device actions.

XML Provisioning

For Android, Win7, and WinRugg Only, XML provisioning allows you to download a custom-designed XML file to a device in a provisioning product. After the file is downloaded, it runs an install command to extract the settings from the XML file and install them on the device. For more information, see Create an XML Provisioning File, Android, Win7, WinRugg.

Products List View

The Product List view in Workspace ONE UEM powered by AirWatch allows you to view, edit, copy, reprocess, and delete products and view the devices on which your products are provisioned.

Navigate to **Devices > Provisioning > Product List View**. Listed here are all the available products for the current organization group. The products can be sorted using the columns.

- **Platform** sorts by the device platform.
- **Managed By** sorts by the organization group the product is assigned to.
- **A/D** sorts by if the product uses activation/deactivation dates or manual.
- **Compliant**, **In Progress**, **Failed**, and **Total Assigned** sort by the status of the product on devices.

View the details and settings of the product by selecting a product by name. The View Product screen displays the general settings, manifest items, conditions, deployment settings, and product dependencies for the product. You can also select the number links in the **Compliant**, **In Progress**, **Failed**, **Has Dependency**, **Must Push**, **Offline**, and **Total** columns, allowing you to see device details as they pertain to these product provisioning statuses.

Select the edit radio button to the left of each product name and you have access to the following actions.

- You can **Deactivate** a product, making it no longer accessible. Deactivating the product also clears all pending provisioning commands.
- Select the **Edit** button to edit a product. You can only edit products after they are deactivated. **Edit** displays the Product Wizard allowing you to change any part of a product.
- Select the **View Devices** button to view all devices to which the product provisions. A quick summary of information on each device allows you to see which devices are at specific statuses. Select a device **Friendly Name** to open the Device Details Page for that device.

More Actions

- You can view the **Activation Log** for the selected product, which displays detailed information about the product including date of activation and the name of the admin who initiated the activation.
- You can make a **Copy** of a product. If one of your products has detailed and intricate parameters, you can save time programming them from the beginning by making a copy of an existing product. You can then, for example, change the application in the manifest of the copy, and as a result, make an entirely new product that shares the same detailed parameters.
- You can attempt to fix non-compliant products and push the product to the device again by selecting the **Reprocess** button.
- The **Force Reprocess** action resends Products to all assigned devices regardless of compliance status. The devices fully download and install every component of the Product manifest, even if it exists on the device already. You can perform this action on multiple products simultaneously.
- Select the **Relay Server Status** button to see the status of the relay server associated with the product. Only active products have the **Relay Server Status** button. You can also view history from the View Devices page to see the past and future products pushed to the device based on Product sync.
- The **Inherited Products** option displays all inherited products a child organization group receives from the parent organization groups. As products are provisioned based on smart groups and not organization groups, your devices can receive products from a parent organization group.

This chapter includes the following topics:

- Products Dashboard

- View Information About Products in Device Details
- Product Job Statuses

Products Dashboard

Workspace ONE UEM powered by AirWatch lets you view and manage products to provision from the Products Dashboard. Navigate to **Devices > Provisioning > Products Dashboard**.

The dashboard provides an easy method of viewing the status of your products and the devices they provision. The charts of information allow you to examine specific products or devices so you can remain informed about your device fleet.

Recent Product Status

This chart displays the 10 most recently created products and the status for each product. You can select any section of the bar graph to view the devices to which that product status applies.

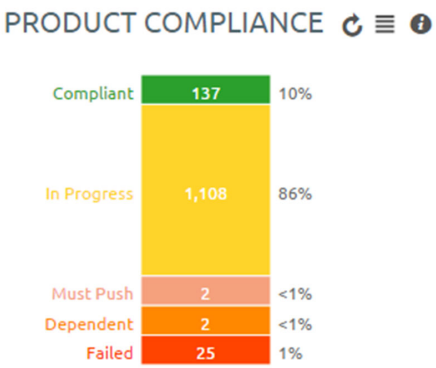
- Compliant – The product installed on the device and the inventory data of the product reported by the device matches the requirements of the product.
  - **For Windows Rugged Only:** The product also passes a file hash validation, ensuring the product on the device is the exact same product provisioned by the console or API call.
- In Progress – The product has been sent to the device and is pending a compliance check based on inventory.
- - **For Windows Rugged Only:** The compliance check also factors in file hash data to be received from the device.
- Must Push – The product deployment type is set to elective. The admin on the console side must initiate product installation.
- Dependent – The product depends on another product installation before installing onto devices.
- Failed – The product reached maximum attempts to install on the device and is no longer attempting to install.

Filters

You can filter the Recent Product Status chart to refer to specific device platforms that support product provisioning. To filter your results, select the **Menu** icon ( ) in the top right corner. Select the platforms you want to filter by.

Product Compliance

The Product Compliance chart shows the total percentage of each compliance status. The number displayed in each status is the total number of product statuses reported from each device.



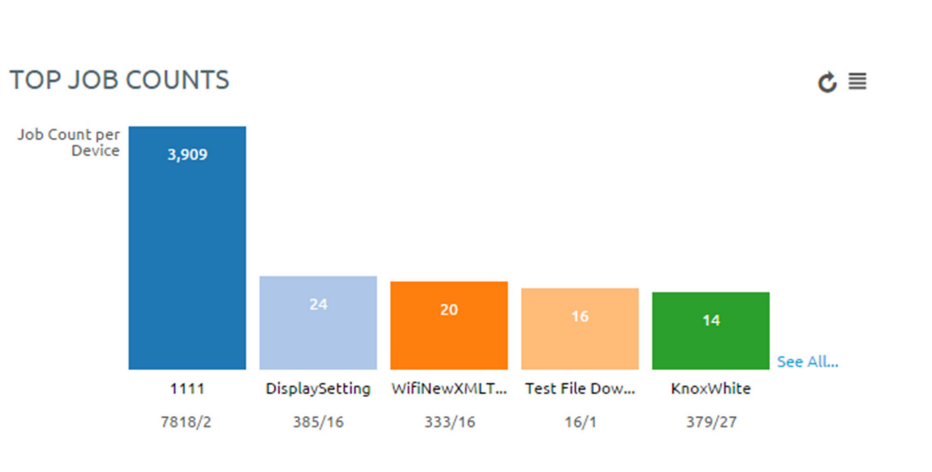
Filters

You can filter the Product Compliance chart to display specific device platforms that support product provisioning and the total percentage of each compliance status for a specific products.

To filter your results, select the **Menu** icon ( ) in the top right corner. Select either the platform or the product by which you want to filter.

Top Job Compliance

This chart displays a ratio of total job count to the number of devices to which the product is provisioned. This ratio gives you information on what products are having issues running.



For example, if the number shown is a 3, then you know that an average of 3 jobs per device happens for this product. If you select the bar for each product, the View Devices screen displays with all devices currently assigned the product. You can then determine which jobs are failing and the reason for those failures.

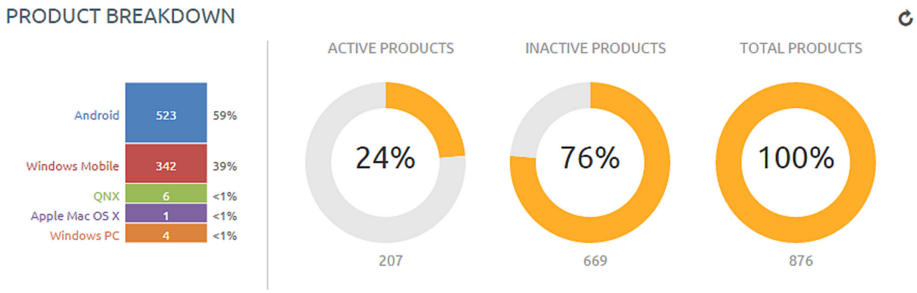
Filters

You can filter the Total Job Compliance chart to refer to specific device platforms that support product provisioning. To filter your results, select the menu icon ( ) in the top right corner. Select the platforms you want to filter by.

Product Breakdown

This section shows you the breakdown of your products. The first chart shows the breakdown of products by platform. Selecting a platform displays the Products List View filtered by that product. This arrangement allows you to see the products available for each platform quickly.

The second chart displays the percentage of your products that are active vs. inactive and a total number of products. Selecting a chart displays the Products List View page filtered by the status of the product.



View Information About Products in Device Details

You can use the Device Details View in Workspace ONE UEM powered by AirWatch to see the products, files/actions, apps, and profiles pushed to a device.

Products

To view the products on a device, navigate to **Devices > List View > Select a device > More > Products**.

Any product that fails to push to devices can be reprocessed by selecting the **Reprocess** button next to the failed product.

Product Sets

Product Sets display on individual device detail pages to show the status of the products' deployments to the device. The products listed that are part of a product set display the product set they pertain to and the deployment status of the products.

The following text boxes display relevant product set information.

- **Product Set** – Displays the product set that contains the product. Select the product set to view the product set details.
- **Status** – Displays the status of the product. For products in a product set, the appropriate product deployed to the device is labeled as **Compliant**. The other products contained in the product set that are eligible for deployment but are not deployed to the device are labeled as **Outranked**. Any product that is not eligible for deployment to the device is labeled as **Not Applicable**.

Files/Actions

Navigate to **Devices > List View > Select a device > More > Files/Actions** to access the files/actions on the device.

Applications

For Android devices only, navigate to **Devices > Details View > Apps** to access the Applications on the device.

Profiles

For Windows Rugged devices, Windows Desktop devices, QNX devices, and Android devices only, navigate to **Devices > Details View > Additional Options > Profiles** to access the Profiles on the device.

Product Job Statuses

Product provisioning works by interpreting each item in a product as a different task or *Job* that the UEM server handles. As a product is pushed to a device, Workspace ONE UEM powered by AirWatch updates the status of each job to display any errors or issues that are in process.

Product Job Logs

You can view more detail about product jobs by viewing the job logs.

Navigate to **Devices > List View** and select the friendly name of a device that has been provisioned with a product. Next, select the **More** tab, select **Products**, then select the magnifying glass icon to the right of the **Last Job Status** column. This action displays the **Jobs** screen which provides access to the contents of the Job logs.

The Job logs provide a detailed history of events that have elapsed for the device in question as it pertains to the assigned product. This history includes timestamps, progress, error messages, and pause/resume history.

Each job follows a workflow and the statuses reflect the position in the process.

Job Status	Description
Canceled	Job canceled while deferred or waiting.
Completed / Failed	Job processing complete. Complete means that the process was a success. Failed means that the process failed.
Deferred	Job download conditions not yet met.
Deleted	The job was canceled by the user on the device.
Delivered	Job initially delivered to device database.
Download Pending	The download remains in a pending state until download conditions are met.

Job Status	Description
Downloaded	The job downloaded to the device.
Install pending	The install is pending until install conditions are met.
Installed	The job installed on the device.
n/a	Not applicable, which can arise in a number of scenarios. If the job is more than 90 days old and has been purged, the job status becomes n/a. Also, if a product contains an application that the device already has installed, the job is not queued, the product status is Compliant, and the Job status would be n/a.
Orphaned	Job being processed by device is incomplete when job is reprocessed. Job automatically restarts when able.
Paused	Job was previously started but a failure occurred. Jobs resume before other jobs are processed.
*Queued	The job is created and currently in line to be started.
Waiting	Job is processing on the device but the status of the job is not confirmed.

Note * If one or more relay servers are part of your product provisioning environment, you might see extended periods in which "Queued" is the official job status. While each relay server added to the processing chain serves to enhance greatly the performance of a busy provisioning environment, they also tend to delay the reporting of a conclusive product status.

You can use the following methods to troubleshoot and investigate a job status where relay servers are involved.

- **Check Relay Server Advanced Info** - Navigate to **Devices > Provisioning > Relay Servers > List View** and select the radio button to the left of the relay server name. Select the **More Actions** button that appears and lastly, select **Advanced Info**. This displays the **Relay Server Advanced Information** screen, where you can verify a product's **Queued Count**. Device commands for products are not sent to devices until any recently created product stops being reported in this text box. Once you see a zero value in this text box, then device commands can be sent.
- **Check Relay Server Status.** - Navigate to **Devices > Provisioning > Product List View** and select the radio button to the left of the product you want to research. Select the **More Actions** button and lastly, select **Relay Server Status**. You can monitor the specific status of the relay server by pointing your cursor over the displayed Relay Server icon.

Job Log Detail Level

You can set the amount of detail captured in the Job Log for Android and Windows Rugged devices only by navigating to **Groups & Settings > All Settings > Devices & Users > Android** or **Windows > Windows Rugged** then continue on to **Hub Settings** then scroll down to the **Product Provisioning** section and select the **Job Log Level** you prefer.

Configure the Collection of Job Logs for Specific Devices

You can target individual devices to collect their provisioning job logs.

1 Navigate to **Groups & Settings > All Settings > Admin > Diagnostics > Logging**.

2 Select the **Enabled** slider for each component and **Scheduled Services** for which you want to collect data.

3 Scroll down to the **Targeted Logging** section, Enable the **Targeted Logging** slider, and complete the settings.

Setting	Description
Organization Groups	Select the organization groups where the devices reside.
Device IDs	Enter the device IDs for which you want to enable targeted logging. Use commas to separate multiple device IDs.
File Storage Impersonation Enabled	Enable if you are using a file storage server to store these targeted logs and enter the appropriate authentication credentials.
File Path	Enter the path and filename of the LOG file where you want the data saved.
File Storage Impersonation User Name	This option appears only when **File Storage Impersonation Enabled** is selected. Enter the user name of the storage server where you targeted logs are saved.
File Storage Impersonation Password	This option appears only when **File Storage Impersonation Enabled** is selected. Enter the corresponding password of the user name of the storage server where you targeted logs are saved.
Test Connection (button)	This button tests various possible scenarios which the logging process uses and makes sure it is working as expected.

4 **Save** to apply Targeted Logging.

What to do next: For Android and Windows Rugged only, you can target an individual device for troubleshooting purposes. See the section below, **Target a Device Log Level for Troubleshooting Purposes**.

Define How Much Data to Collect, Product Job Log

You can define the length of time job log data is collected. Define this timescale by taking the following steps.

1 Navigate to **Groups & Settings > All Settings > Admin > Data Purging**.

2 Locate the purge module named **DevicePolicyJobPurge** and select the pencil icon ( ) to open the **Data Purging** screen.

3 Complete the **Purge older than (days)** setting with the length of time in days that you want to keep job log data.

4 Select **Save**.

Job logs older than the selected number of days are purged from the Workspace ONE UEM console.

Target a Device Log Level for Troubleshooting Purposes, Android and WinRugg

You can target an individual device and temporarily change its logging level for troubleshooting purposes.

1 Navigate to **Devices > List View**, locate the device you want to troubleshoot and select the device-friendly name to display the **Device Details**.

2 Click the **More** tab and select **Targeted Logging**.

3 Select **Create New Log** and select the length of time you want the log to capture data.

4 Select **Start** to begin the logging.

Device Management

6

You can manage devices you have provisioned products onto using the tools and features included with Workspace ONE UEM, such as Device Details, Enterprise Reset, Advanced Remote Management, and much more.

The features presented in the topics that follow are specific to devices associated with product provisioning, including Windows Rugged, QNX, and others. For information about device management or the Workspace ONE UEM Console in general, see VMware Workspace ONE Managing Devices and VMware Workspace ONE Console Basics.

Device Details

The Device Details page displays detailed device information and lets you quickly access user and device management actions.

You can access the Device Details page by selecting a device's Friendly Name from the Device Search and Device List View pages. You can also use one of the available Dashboards or any of the available search tools with the Workspace ONE UEM console.

From the Device Details page, you can access specific device information broken into different menu tabs. Each menu tab contains related device information depending on your Workspace ONE UEM deployment.

Workspace ONE Assist

Workspace ONE Assist, previously named Advanced Remote Management (ARM), allows you to connect remotely to end-user devices so you can help with troubleshooting and maintenance. The Assist Server facilitates communication between the Workspace ONE UEM and the "host" device.

For more information, see VMware Workspace ONE Assist Documentation.

This chapter includes the following topics:

- Configure Settings, QNX
- Enterprise Reset a Rugged Device, Android and WinRugg
- Enable AirWatch Cloud Messaging, Android Rugged
- Platform OEM Service, Android Provisioning

- Batch (BAT) File Guidelines, Win7 and WinDesk

Configure Settings, QNX

Workspace ONE UEM powered by AirWatch, together with the Workspace ONE Intelligent Hub app, manage all the special settings that QNX devices have. You can change these settings when you need the Hub to meet certain business needs.

Configure QNX Settings

1 Edit the Workspace ONE Intelligent Hub Settings by navigating to **Groups & Settings > All Settings > Devices & Users > QNX > Hub Settings**.

Setting	Description
Device ID Algorithm	Set the unique device identification algorithm used on the device.
Heartbeat Interval (min)	Set the time (in minutes) the Workspace ONE Intelligent Hub waits before checking-in with the Workspace ONE UEM console.
Data Sample Interval (min)	Set the time (in minutes) the Workspace ONE Intelligent Hub waits to collect data from the device.
Data Transmit Time Interval (min)	Set the time (in minutes) the Workspace ONE Intelligent Hub waits to send data to the UEM console.

For information about initiating a remote management session for troubleshooting purposes, see VMware Workspace ONE Assist.

QNX devices use the Workspace ONE Intelligent Hub to deliver Job notifications for products provisioned to the device. Notifications for jobs that are completed display on the device side through the Workspace ONE Intelligent Hub for QNX devices.

AWTrigger Commands, QNX

AWTrigger allows third-party applications to interact with the Workspace ONE Intelligent Hub for QNX.

The applications interact in two ways. The first interaction is the ability to enable/disable the process of products using the fast track. This setting allows you to evaluate products regardless of any conditions enabled for the product. For example, you can install all products available for a device immediately instead of waiting for conditions or dependencies.

The second interaction ability allows you to evaluate all device readiness and detached conditions immediately. This interaction circumvents the normal check-in interval for the Workspace ONE Intelligent Hub and monitors all conditions at that moment as opposed to waiting for the normal interval.

This appendix lists the commands available for use while using the Workspace ONE ™ UEM Installation Directory Command-Line entries.

AWTrigger –installnow true

This command processes all the jobs that have already reached the device (and also all the jobs that reach the device after this command has been successfully run) to be processed with immediate effect by disabling all conditions.

AWTrigger –installnow false

This command disables the installnow functionality. All the jobs that reach the device after this command has been run, will be processed normally (by evaluating conditions).

AWTrigger –condition true

This command causes all the deferred jobs on the device to be reevaluated again with immediate effect to see if the condition specified (in each job) has been met or not. Useful in the case of file conditions.

For example, suppose that a job has been pushed onto the device which has a file condition associated with it and the file condition specifies to monitor the presence of a test file. If the test file is missing on the device, AWApplicationManager creates a flag file. Hub defers this job by 5 minutes. After 5 minutes, the AWApplicationManager will again check for the presence of the test file). If the application creates a test file after 2 minutes (of deferring the job) and if the technician does not want to wait for another 3 minutes for the job to be processed, he can run this command and the job will be immediately evaluated. This command causes ALL the deferred jobs to be evaluated with immediate effect.

AWTrigger –h

This prints the usage of the utility into the respective log file.

AWTrigger -migrateca true

This migrates your XML custom attribute files to the new custom attribute database.

In order for the new text boxes to be present in a custom attribute XML file created from a profile, the UEM console version must be at least 8.1, and the Workspace ONE Intelligent Hub version must be at least 5.4.66.98.

For clean migration of custom attribute data from XML files to the database, Workspace ONE UEM recommends repushing any profiles that are already installed on any devices so that the new text boxes are present.

Migrate from XML to CA Database, AWTriggers

You can migrate information in XML format to the custom attribute database by taking the following steps.

1 Repush existing profiles to update XML files with data for all profile text boxes.

2 Update configuration file "~/airwatch/General-Config.cfg" to use DB_BASED_CA.

```
[CustomAttributes] Type = DB_BASED_CA
```

3 Run command for custom attribute migration utility.

```
~/airwatch/AWTrigger -migrateca true
```

4 Check Status.

```
~/airwatch/AWStatusFinder -migrateca
```

Migrating from AirWatch 8 XML to CA Database, AWTriggers

Workspace ONE UEM recommends re-pushing the custom attribute profile from the UEM console after upgrading to AirWatch v8.1.

If you choose to migrate to a CA Database without pushing the updated profile, the following decisions are made by the migration process.

- Importing Application values.
  - If an Application value does exist for a custom attribute record in an XML file, then the existing value is used as the value for application when the record is inserted into the database.
  - If an Application value does not exist for an attribute record in an XML file, then the **File Name** is used as the Application value when the record is inserted into the database.
- Importing Attribute Name values.
  - The name of the custom attribute record in the XML element is imported as the name of the custom attribute database record.
- Importing value.
  - The value of the custom attribute record in the XML element is imported as the value of the custom attribute database record.
- Importing is_dynamic values.
  - If an is_dyanmic value does exist for a custom attribute record in an XML file, then the existing values are imported as the is_dynamic value for the database record.
  - If an is_dyanmic value does not exist for a custom attribute record in an XML file, then the is_dynamic value is set to "True" for the database record.
- Importing Permission values:
  - If a Permission value does exist for a custom attribute record in an XML file, then the existing value is imported as the Permission value for the database record.
  - If a Permission value does not exist for a custom attribute record in an XML file, then the Permission value is set to "read/write" for the database record.

- Importing sync.
  - If a Sync value does exist for a custom attribute record in an XML file, then the existing value is imported as the Sync value for the database record.
  - If a Sync value does not exist for a custom attribute record in an XML file, then the Sync value is set to "True" for the database record.

Enterprise Reset a Rugged Device, Android and WinRugg

Workspace ONE UEM powered by AirWatch lets you "Enterprise Reset" a device, which is a process that is similar to an enterprise wipe, but with one important difference. Profiles and files/ actions set to persist on a device are not removed and automatically reinstall on a device following the first reboot after an enterprise reset.

Enterprise Reset is only available for Windows Rugged Devices and Android legacy Motorola, Zebra, and Honeywell devices.

Note Enterprise Reset cannot run on devices with low battery levels. If you attempt an Enterprise Reset on a device with low battery level, a warning displays alerting to you about potential issues with the Enterprise Reset. On Android devices, the Enterprise Reset command is held until the device reaches sufficient battery level. Once the device is charged, the Reset occurs automatically.

Note **Honeywell Work Managed** devices do not currently support the Enterprise Reset action.

Note **Zebra Work Managed** devices have special requirements for applying an Enterprise Reset.

- Zebra device must have Workspace ONE Intelligent Hub 1909 or later installed.
- Zebra devices must have Android 7.0 (Nougat) or later installed.
- Zebra devices must have MX 4.5.0.3 or later installed.
- Zebra device must be enrolled as Work Managed device with a Stage Now barcode using the Device Owner (DO) APF file.
- Zebra device must be connected to Wi-Fi before launching the Workspace ONE Intelligent Hub, otherwise enrollment is blocked. Admin must ensure a Wi-Fi profile is pushed with the persist option enabled before initiating the Enterprise Reset.
- Once the Enterprise Reset has finished, the device re-registers itself as Android Enterprise when the Workspace ONE Intelligent Hub is run for the first time.
  - If the registered Android Enterprise account is Laforge, the device end user can skip the gmail account registration.
  - If the registered Android Enterprise account is G Suite, the device end user must provide the previously registered account to finish the device wizard setup.
- Agent Upgrades
  - Persisted profiles, Event/Actions, and Files-Actions pushed to Hubs earlier than 1908 are retained after an Enterprise Reset but applications must be reinstalled. These applications persist after subsequent resets.
  - Persisted profiles, Event/Actions, Files-Actions, and Applications pushed to Hubs version 1908 and later are retained after an Enterprise Reset.

Prerequisites

Once you have complied with the above prerequisites, take the following steps to run an Enterprise Reset.

Procedure

1. Navigate to **Devices > List View** and select a device you want to Enterprise Reset.
2. On the Device Details View, select the **More Actions** button.

3 Select **Enterprise Reset**, located under Management section.

If you do not see the Enterprise Reset option for the selected device, then the device might not be eligible for this device action. Please see the list of prerequisites for details.

4 Enter your **Security Pin** in the **Restrict Action** prompt to perform the Enterprise Reset.

Enable AirWatch Cloud Messaging, Android Rugged

AirWatch Cloud Messaging (AWCM) provides an internal communication solution for the entire Workspace ONE UEM powered by AirWatch solution as a comprehensive replacement for Google Cloud Messaging (GCM).

AWCM provides real-time device management status and command pushes for:

- Devices that cannot be configured with a Google Account.
- Devices restricted to internal network communication.
- Devices without public Internet access.

Enable AWCM

Enable AWCM by navigating to **Devices > Device Settings > Android > Hub Settings > AirWatch Cloud Messaging**.

Select **Enabled** on **Use AWCM Instead of C2DM** to enable AWCM. Selecting this option locks the deployment type to **Always Running** so that the system and device have a constant and ongoing line of communication. You can also leave the **Use AWCM Instead of C2DM** check box deselected and decide to make the deployment type **Always Running** or **Manual**, with an associated timeout value.

Platform OEM Service, Android Provisioning

The Platform OEM (POEM) Service is an extra application that allows the console for Workspace ONE UEM powered by AirWatch to provide extended management capabilities to Android legacy devices only.

After you enroll, the Workspace ONE UEM console automatically detects if the device can take advantage of additional device capabilities, and deploys an Original Equipment Manufacturer (OEM) specific service application to your Android. The OEM Service app is a plug-in app that is only installed and used with Workspace ONE Intelligent Hub enrollment.

It allows for additional MDM capabilities that only pertain to a specific OEM device. All these APKs are available through AirWatch Resources by request. There are a few service applications that we publish to the Google Play Store (see list below).

Here is a sample of supported features and available OEMs for the Platform OEM Service.

POEM Service Features

- Silent App installation, uninstallation, and updates
- Silent Device Administrator Activation on start
- Date/Time configuration (date format, time format, time zone, server time, SNTP, HTTP URL, or Auto)
- Toggle Bluetooth on/off with the Disable Bluetooth restriction.
- Disable installation from unknown sources on 5.0 Lollipop and above.
- Device Reboot

POEM Service Versions

- Bluebird
- Kube
- Getac
- Honeywell
- HP
- Intermec
- Lenovo
- Mediawave
- Panasonic
- Sonim
- Zebra CC5000

POEM Service Version Available on the Google Play Store

- Samsung
- Sony
- LG
- Huawei
- Zebra
- Honeywell

Batch (BAT) File Guidelines, Win7 and WinDesk

For product provisioning in Workspace ONE UEM powered by AirWatch, Windows 7 and Windows Desktop devices can be configured using batch (BAT) files. While writing and running these batch files, there are best practices you should follow.

Accounting for Path

Windows Unified Agent is a 32-bit application, so when trying to run scripts in a 64-bit machine, proper redirections must be used to get access to the 64-bit folder or the registry hive.

There are two `%windir%\System32` directories on a Windows x64 system.

- **%windir%\System32** directory is for 64-bit applications. This directory contains a 64-bit cmd.exe.
- **%windir%\SysWOW64** directory is for 32-bit applications. This directory contains a 32-bit cmd.exe.

Since Workspace ONE Intelligent Hub is a 32-bit application, it can access %windir%\System32 for running 64-bit applications by using **%windir%\Sysnative** in path.

Admin must use **%windir%\Sysnative** in script to access any 64-bit applications.

For example,

```
%windir%\Sysnative\manage-bde -on c: -skiphardwaretest
```

- **manage-bde** is a 64-bit application and it can be accessed only by providing proper path **%windir%\Sysnative**.
- **Certutil** is part of both folders (32-bit and 64-bit), so no need to give %windir%\Sysnative in the script.

Writing Scripts for Registry

Since Windows Unified Agent is a 32-bit application, it always creates a record or performs any action on WOW6432 Node.

On 64-bit Windows, `HKLM\Software\Wow6432Node` contains values used by 32-bit applications running on the 64-bit system.

32-bit applications do not create records in HKLM\Software directly.

To write explicitly to a 64-bit hive, add the /reg:64 modifier to the end of your REG ADD command in scripts to create a record in the HKLM\Software registry path.

For example, `REG ADD HKLM\Software\MyApp /reg:64`

General Instructions

- Running scripts in admin context when Standard User is logged in performs actions for Admin User.

  For example,

  Running a script in User context installs a certificate for a standard user in the Current user store.

  Running a script in Admin context installs a certificate for an Admin in the Current user store.

- The path must be quoted while passing arguments to batch files.

  For example,

  ```
  "C:\Passing_Argument.bat" Hello World
  ```

- The BAT file extension must always be included in the file path. Omitting this extension causes a file not found error and the script fails to run.
- You must always have file action as run while deploying batch files.

Lookup Values 7

A lookup value is a variable that represents a particular data element of a device, user, or admin account in Workspace ONE UEM and Workspace ONE Express. Lookup values can be invaluable in completing a process or a form.

In several different text boxes in the Workspace ONE UEM console and Workspace ONE Express, you can add lookup values in place of manually entered or static values. In most cases, lookup values function as a stand-in for a piece of information you do not know or do not have access to.

For example, the **Add Device** screen is used to add a device to your fleet. One of the text boxes on this screen that can be completed with lookup values is the **Expected Friendly Name**.

The friendly name represents the device on many different screens in the UEM console including the **Device List View** and the **Details View**. And while you can manually enter a static friendly name when you add a device, you can instead use lookup values to standardize the friendly name and make it a valuable identifier.

A common friendly name format can be constructed with the following lookup values.

```
{EnrollmentUser} {DeviceModel} {DeviceOperatingSystem} {DeviceSerialNumberLastFour}
```

If you enter this string in the **Expected Friendly Name** text box, it produces a friendly name that appears this way on the **Device List View**.

```
jsmith iPad iOS GHKD
```

This friendly name instantly provides you with at least three useful pieces of information. And with the last four digits of the device serial number at the end, the friendly name is almost sure to be unique.

Data Overhead

When used, lookup values do not add to the device's memory with an extra load. Lookup values are a construct of the console itself, not something that is transferred to the device.

Static Strings Versus Lookup Values

Lookup values cannot be applied once a static string has been entered in a text box.

For example, assume that you have 100 devices to enroll. You add the first 50 devices using a manually entered static string for **Expected Friendly Name**. For the next 50 devices, you opt to use a lookup value for **Expected Friendly Name** instead. Those 100 devices, half with static friendly names and the other half with lookup values, can coexist perfectly well. There is no issue with mixing and matching static strings and lookup values.

However, you cannot return to the first 50 devices and replace the static string-friendly name with a lookup value.