Linux Device Management

VMware Workspace ONE UEM 2203

vmware®

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

Workspace ONE UEM on Linux 1

Use Workspace ONE UEM to manage and secure your enterprise Linux devices. The Workspace ONE UEM console gives you tools and features to manage the entire lifecycle of Linux devices.

The flexibility of the Linux operating system makes it a preferred platform for a wide range of uses, including developer workstations, Raspberry Pi devices, and many IoT devices. With Workspace ONE UEM, you can build on the flexibility and ubiquity of Linux devices and manage them alongside your other enterprise devices in one central location.

This chapter includes the following topics:

Requirements for Workspace ONE UEM on Linux

Workspace ONE UEM is compatible with all distributions of Linux running on x86_64, ARM5, or ARM7 architectures, although not all features might be available on every distribution. Make sure that your system meets the Workspace ONE UEM version and network requirements before you deploy your Linux devices.

Linux Device Requirements

You can enroll devices running any distribution of Linux running on x86_64, ARM5, or ARM7 architectures into Workspace ONE UEM.

- Installers are created for specific distributions and architectures. Ensure that you are using the correct installer for your use case.
- To run Hub as a system service, the device must be running System D or System V.
- Configurations using Workspace ONE profiles requires a Puppet agent (open source). When running a Debian-based (deb) or Red Hat-based (rpm) system, the puppet agent is installed automatically with Hub. For other systems, or when using the tarball method of installation, install the puppet agent manually prior to Workspace ONE enrollment. For more information, see the Installation section.

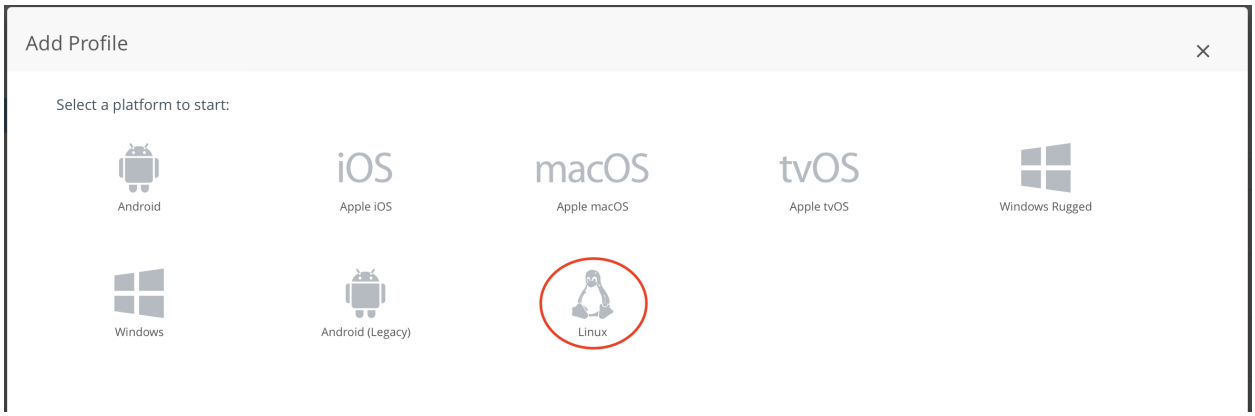
Workspace ONE UEM Requirements

Currently, this feature is in **Limited Availability** and might not be available in your environment, nor is it currently available for on-premises installations of Workspace ONE UEM. New Workspace ONE UEM infrastructure called Control Plane (also required for Freestyle Orchestrator) must be enabled in your environment so that Workspace ONE UEM can support managing Linux devices. This is available in Shared SaaS UAT environments CN135, CN137, and CN138 and we will be rolling it out to Dedicated and Shared SaaS environments following Control Plane and Workspace ONE UEM 2109 being deployed.

- You must deploy the Workspace ONE Intelligent Hub for Linux v21.10 or later. Previous versions including v21.01 and v1 of the Workspace ONE Intelligent Hub for Linux do not support the features described in this guide.
- Workspace ONE UEM 2109 or later with Control Plane implemented are both required to be able to use Linux Management.

Before attempting to enroll a Linux device, determine if Linux management is enabled in your Workspace ONE UEM environment. The easiest way to validate this is to create a new profile. Navigate to **Resources > Profiles & Baselines> Profiles> Add> Add Profile** to see if Linux is an available option.

If Linux as an option, similar to the following, then Linux management is enabled. You can enroll a Linux device. If there is no Linux option, then enrolling a Linux device will not work in your environment. See the following KB article to read more about when environments are scheduled to be upgraded.



Enroll Your Linux Devices

2

Install the Workspace ONE Intelligent Hub on your Linux devices to establish communication between devices and the Workspace ONE UEM console. Linux enrollment is a two-step process; first you install the Intelligent Hub, and secondly you enroll devices with the `ws1HubUtil` command. If desired, you can fully script the enrollment in a single command, or you can prompt the user to enter enrollment information.

Devices can be enrolled at the same time you install the Intelligent Hub, or devices can be enrolled with `ws1HubUtil` after after installation of the Intelligent Hub. In either case, you can fully script the enrollment in a single command, or you can prompt the user to enter enrollment information.

Prerequisites

You must enroll Linux devices to establish communication between the devices and Workspace ONE UEM for devices to access internal content and features.

Currently, SAML authentication or directory lookup for enrollment is not supported. Also, advanced and single-user staging enrollments are not supported, where an admin enrolls on behalf of a user, or enrolls and waits for a user to enter credentials. The enrollment user must exist in the WS1 UEM console (basic or pre-synced Directory user acccounts).

A working knowledge of the linux command line is required. Unlike other platforms, the WS1 Intelligent Hub for Linux does not have a graphical user interface and requires the command line to install, enroll, and interact with it on the device itself.

Before enrolling a Linux endpoint, gather the UEM Device Services URL, organization Group ID, and the enrollment user's username and password. You can also enroll with a token using the '`--token`' argument and use it in the `` `--group` `` argument. You are prompted for username and password, but you can leave these blank when enrolling with a token. Consult the section of this guide titled "Supported Command-line Arguments for the ws1HubUtil Enroll Command" for more details of the enrollment arguments and options available, or run the command `./ws1HubUtil enroll -h`

```
admin@ubuntu-1:/opt/vmware/ws1-hub$ ./bin/ws1HubUtil enroll -h
Usage:
  ws1HubUtil [OPTIONS] enroll [enroll-OPTIONS]

Application Options:
  -v, --version             retrieve hub version

Help Options:
  -h, --help                Show this help message

[enroll command options]
      -u, --user=           username for enrollment
      -p, --password=       password for enrollment
      -g, --group=          organization groupID to which device must enrol
      -t, --token=          auth token for enrollment
      -s, --server=         console DS URL to which device has to enrol
          --proxy-server=   if enrollment needs to go through proxy, provide the proxy server info
          --proxy-password= password for proxy server, applicable if proxy requires username
          --proxy-user=     username for proxy server, applicable if proxy requires username password
```

Installers are created for specific distributions and architectures. Ensure you are using the correct installer for your device and distribution.

The device must be running SystemD or System V init in order for Hub to run as a system service. Puppet agent is required for custom configurations. If running a Debian-based (deb) or Red Hat-based (rpm) system, this is installed automatically with Hub. For other systems, it must be installed manually prior to WS1 enrollment.

Procedure

1 Download the Workspace ONE Intelligent Hub for Linux to your intended device. The downloaded file must correspond to the targeted processor architecture and distribution. The agent is available as deb, rpm, or tgz packages and can be downloaded either directly to your Linux device or can be transferred to the Linux device via USB or SSH.

The installers can be retrieved from the following locations:

Architecture	Debian Based	Red Hat Based	Other (tarball)
x86_64	amd64-21.10.0.1.deb	amd64-21.10.0.1.rpm	amd64-21.10.0.1.tgz
ARM5	arm5-21.10.0.1.deb	arm5-21.10.0.1.rpm	arm5-21.10.0.1.tgz
ARM7	arm7-21.10.0.1.deb	arm7-21.10.0.1.rpm	arm7-21.10.0.1.tgz

2 Run the Workspace ONE Intelligent Hub client installer with root privileges.

For example:

```
For Debian package on Ubuntu:
$ sudo apt install "/tmp/workspaceone-intelligent-hub-amd64-21.10.0.1.deb"

For RPM package on Fedora:
$ sudo dnf install workspaceone-intelligent-hub-amd64-21.10.0.1.rpm

For RPM package on OpenSUSE:
$ sudo zypper install workspaceone-intelligent-hub-amd64-21.10.0.1.rpm
```

```
For Tarball (any other linux distribution):
1. Extract the Package using: $ tar xvf workspaceone-intelligent-hub-<arch>.21.10.0.1.tgz
2. Install the Package using: $ sudo ./install.sh
```

Note When utilizing the Tarball, Ruby must be manually installed prior to installing the Intelligent Hub.

3 Enrollment

Enroll your device in Workspace ONE UEM after the installation by using the `ws1HubUtil`. Choose to send enrollment details in one command or separately. Follow the steps below to send them in one command.

4 1. Change directory to the Hub binary directory under the installation directory.

```
$ cd /opt/vmware/ws1-hub/bin
```

5 Run the **`ws1HubUtil`** and include the enrollment arguments in order.

```
$ sudo ./ws1HubUtil enroll --server https://host.com --user <username> --password
<password> --group <organization group id>
```

To prompt the users for enrollment credentials when they enroll, run the `ws1HubUtil` without these additional arguments. Please see Supported Command Line Arguments for more details before attempting an enrollment.

6 After successful installation and registration, the linux device will be listed in the WS1 UEM Console.

7 Uninstall

To uninstall the Intelligent Hub for Linux you can either send an Enterprise Wipe command from the WS1 UEM Console (for an enrolled device) or you can manually uninstall device side. If the uninstall command is used device side on an enrolled device, the device will be unenrolled first.

For Debian: $ sudo apt remove workspaceone-intelligent-hub

For RPM for Fedora: $ sudo zypper remove workspaceone-intelligent-hub

For Tarball: $ sudo /opt/Workspace-ONE-Intelligent-Hub/uninstall.sh

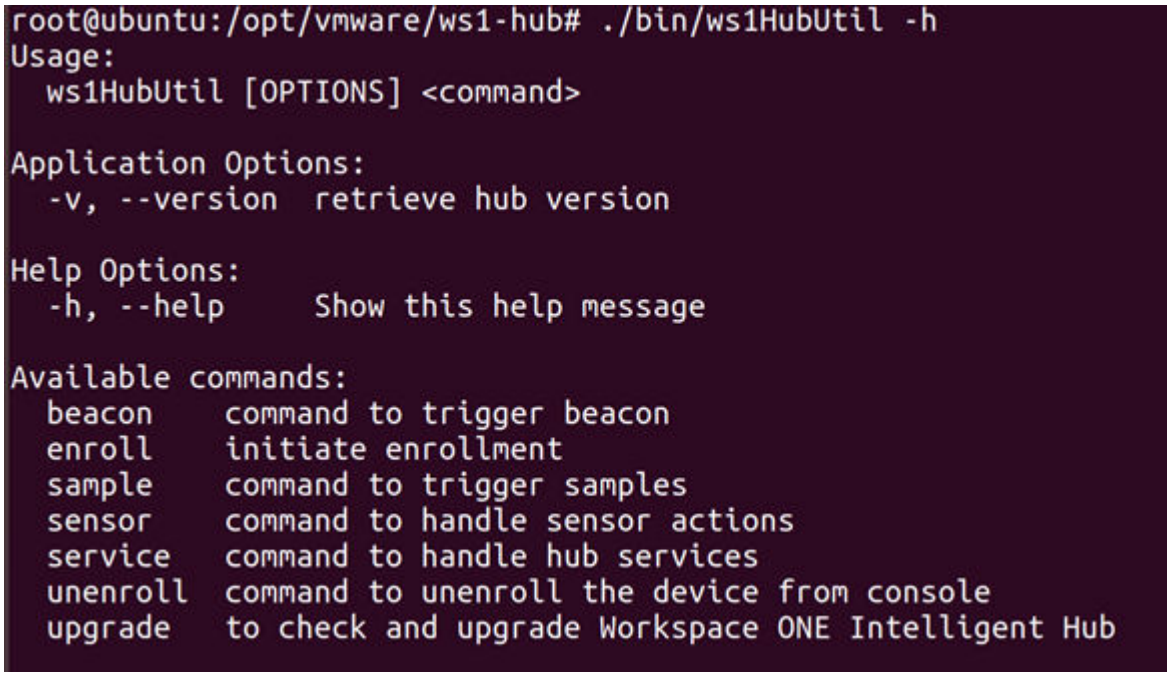
Command-line Utilities for Workspace ONE Intelligent Hub on Linux

Use these command-line utilities to expedite your deployment of the Workspace ONE Intelligence Hub on your Linux devices. The `ws1HubUtil` application is located at the agent binary directory under the installation directory: /opt/vmware/ws1-hub/bin

The `ws1HubUtil` that is installed on the linux device includes the following commands:

- Version

- Enroll
- Beacon
- Sample
- Sensor
- Service
- Upgrade
- Unenroll



Version: The version argument will print the hub installed version.

`./ws1HubUtil –version` or `./ws1HubUtil –v`

Enroll: The enroll command handles the hub native enrollment process.

```
./ws1HubUtil enroll --user xyz --password xyz --group xyz --server https://<host>.com
```

Table 2-1. Supported Command-line Arguments for the ws1HubUtil Enroll Command

Command-line argument	Short Name	Value	Description	Comments
--user	-u	Enrollment user string	User credentials generated from console.	You are prompted to enter the details if the command line argument is not entered.
--password	-p	Password String	Credentials generated from console.	You are prompted to enter the details if the command line argument is not entered.
--group	-g	Organization group String	Organization groupID to which device must enroll.	You are prompted to enter the details if the command line argument is not entered.
--token	-t	Enrollment Token	Used for token based enrollment	Used if OG enrollment type is set for token. Token can also be passed as --group field
--server	-s	Server String	**Fully qualified** UEM console URL to which device has to enroll. This would typically be the device services URL not necessarily the console URL, for example https://ds135.awmdm.com	You are prompted to enter the details if the command line argument is not entered.
--proxy-server	N/A	Proxy server info	Use during enrollment if hub needs to use proxy info (optional)	If enrollment needs to go through a proxy, provide the proxy server info using this argument
--proxy-user	N/A	Proxy username	Username for the proxy (optional)	Applicable if --proxyServer is provided and if proxy requires username and password
--proxy-password	N/A	Proxy password	Password for the proxy (optional)	Applicable if --proxyServer is provided and if proxy requires username and password

Beacon: The beacon command notifies hub scheduler to trigger beacon (heartbeat) immediately

`./ws1HubUtil` beacon

Sample: The sample command notifies hub scheduler to trigger a sample immediately, by default hub will collect and send all the samples. A customized sample can also be triggered. The allowed sample types are - system, network, certificate, profile or all.

`./ws1HubUtil sample` [will trigger all sample]

or

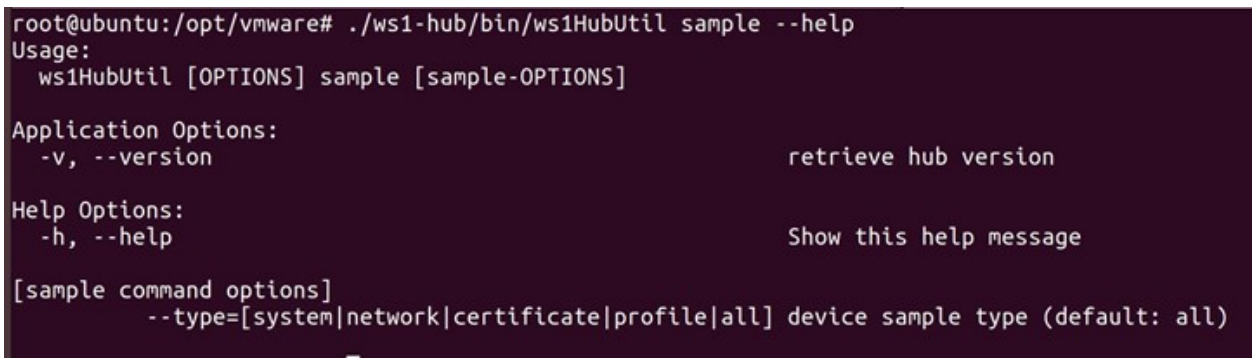
`./ws1HubUtil sample --type` [system] or [network] or [certificate] or [profile] or [all]

Table 2-2. Supported Command-line Arguments for the Agent Utility

Command-line argument	Value	Description	Comments
--type	system | network | certificate | profile | all	Notify scheduler to trigger sample immediately	A sample now job will be queued to the hub scheduler with the specified sample type

Sample Type	Samples Collected
system	System | memory | device capability
network	Network adapter | WLAN
certificate	certificate
profile	profile
all	All of the above

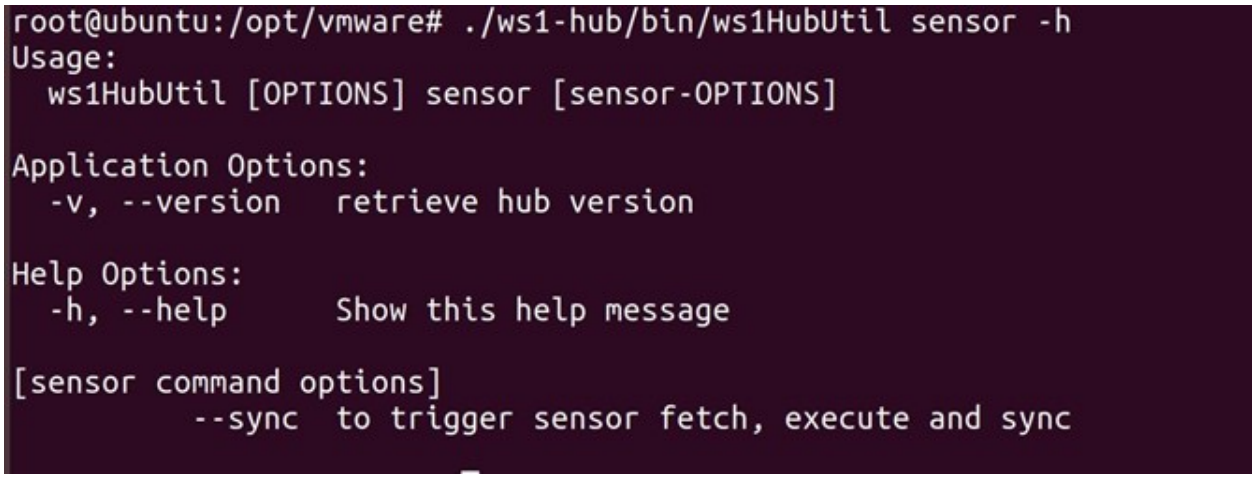
Figure 2-1.



Sensor: The Sensor command notifies hub scheduler to trigger a WS1 Sensor sync immediately. This will fetch latest WS1 Sensors, run periodic WS1 Sensors and transmit the latest values from the device to the UEM console.

`./ws1HubUtil sensor --sync`

Figure 2-2.

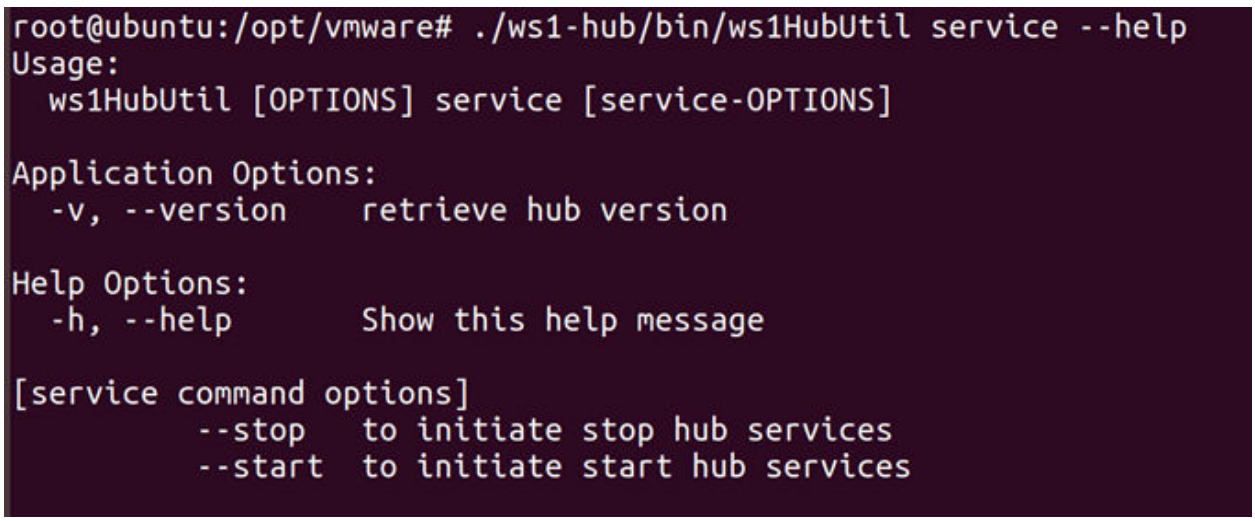


Service: The Service command provides the option to either start or stop hub services running on the device.

```
./ws1HubUtil service [--start] or [--stop]
```

Command options long name	value	description	comments
--start	N/A	Starts hub services	
--stop	N/A	Stops hub services	

Figure 2-3.



Upgrade: The upgrade command will queue an upgrade hub check job with the hub scheduler. This will check if a newer WS1 Intelligent Hub is available. If a newer version is available, it will automatically fetch the latest package and upgrade the Hub.

```
./ws1HubUtil upgrade
```

Unenroll: The Unenroll command sends a request to the UEM console to unenroll the device. This command requires connectivity to UEM to execute. If the command is successful, the UEM console marks the device as unenrolled and send a successful response back to the device, which would be followed by uninstallation of the Hub on the device. If not successful, the failure is captured to the device logs and device remains enrolled.

```
./ws1HubUtil unenroll
```

Linux Device Management

3

After your devices are enrolled and configured, manage these devices using the Workspace ONE UEM console. The management tools and functions enable you to keep an eye on your devices and remotely perform administrative functions.

While you can manage all your devices from the UEM console, the reporting details and available actions for enrolled devices may vary based on your deployment type and device platform.

The Device Dashboard is a searchable, customizable view that you can use to filter and find specific devices. This feature makes it easier to perform administrative functions on a particular set of devices.

The Device List View displays all devices currently enrolled in your Workspace ONE UEM environment and their status. You can filter the list view specific to Linux and see how devices are being managed at a glance.

The Device Details page provides device-specific information such as hardware details, profiles, and network details. You can also perform remote actions on the device from the Device Details page that are platform-specific.

This guide takes you through some specifics related to Management of Linux devices, for further information on any of these functions, see "Managing Devices" in the Workspace ONE UEM Console Documentation.

This chapter includes the following topics:

Device Dashboard

As devices are enrolled, you can manage them from the **Device Dashboard** in Workspace ONE UEM powered by AirWatch.

The **Device Dashboard** provides a high-level view of your entire fleet and allows you to act on individual devices quickly.

You can view graphical representations of relevant device information for your fleet, such as device ownership type, compliance statistics, and platform breakdowns. The Device Dashboard includes the following sections:

- **Security** – View the top causes of security issues in your device fleet. Selecting any of the doughnut charts displays a filtered **Device List** view comprised of devices affected by the selected security issue. If supported by the platform, you can configure a compliance policy to act on these devices.
  - **Compromised** – The number and percentage of compromised devices (jailbroken or rooted) in your deployment (Linux currently does not support this)
  - **No Passcode** – The number and percentage of devices without a passcode configured for security. (Linux currently does not support this)
  - **Not Encrypted** – The number and percentage of devices that are not encrypted for security. (Linux currently does not support this)

  Ownership – View the total number of devices in each ownership category. Selecting any of the bar graph segments displays a filtered **Device List** view comprised of devices affected by the selected ownership type.
- **Last Seen Overview/Breakdown** – View the number and percentage of devices that have recently communicated with the Workspace ONE UEM MDM server. For example, if several devices have not been seen in over 30 days, select the corresponding bar graph to display only those devices. You can then select all these filtered devices and send them a message requesting that they check in.
- **Platforms** – View the total number of devices in each device platform category. Selecting any of the graphs displays a filtered **Device List** view comprised of devices under the selected platform.
- **Enrollment** – View the total number of devices in each enrollment category. Selecting any of the graphs displays a filtered **Device List** view comprised of devices with the selected enrollment status.
- **Operating System Breakdown** – View devices in your fleet based on operating system. There are separate charts for each supported OS. Selecting any of the graphs displays a filtered **Device List** view comprised of devices running the selected OS version. (Linux currently does not support this)

Device List View

Use the Device List View in Workspace ONE UEM powered by AirWatch to see a full listing of devices in the currently selected organization group.

The Last Seen column displays an indicator showing the number of minutes elapsed since the device has checked-in. The indicator is red or green, depending on how long the device is inactive. The default value is 480 minutes (8 hours) but you can customize this by navigating to Groups & Settings > All Settings > Devices & Users > General > Advanced and change the Device Inactivity Timeout (min) value.

Note For Linux devices, the version number shown in the Device List View is the kernel version for the enrolled device, not the version of the distribution.

Select a device-friendly name in the General Info column at any time to open the details page for that device. A Friendly Name is the label you assign to a device to help you differentiate devices of the same make and model.

Sort by columns and configure information filters to review activity based on specific information. For example, sort by the Enrollment column to quickly see any devices that are currently unenrolled from Workspace ONE UEM. You can also search across all devices for a friendly name or user name to isolate one device or user.

Device List View Action Button Cluster

With one or more devices selected in the Device List View, you can perform common actions with the action button cluster including Query, and other actions accessed through the More Actions button.

Available Device Actions vary by platform, device manufacturer, model, enrollment status, and the specific configuration of your Workspace ONE UEM console. For Linux you have the following options:

- Query - This option submits an on-demand request for a device to send updated sample data back to the Console.
- Enterprise Wipe - This option will unenroll, and remove any Wi-Fi and credentials that Workspace ONE UEM pushed down to the device and then ultimately will remove the Workspace ONE Intelligent Hub from the device. This option will not remove any custom configuration profiles that were sent to the device. See Custom Configuration profiles for more information.
- Manage Tags - View the currently assigned device tags and see a list of tags available to be assigned with the Manage Tags screen.
- Assign Tags - Assign a customizable tag to a device, which can be used to identify a special device in your fleet.
- Change Organizational Group - Change the device's home organization group to another existing OG. Includes an option to select a static or dynamic OG.
- Change Ownership - Change the Ownership setting for a device, where applicable. Choices include Corporate-Dedicated, Corporate-Shared, Employee Owned and Undefined.

- Delete Device - If a device is still enrolled, this will issue an unenroll command before removing the entry from Workspace ONE UEM.

Customize Device List View Layout

Display the full listing of visible columns in the Device List view by selecting the Layout button and select the Custom option. This view enables you to display or hide Device List columns per your preferences.

There is also an option to apply your customized column view to all administrators at or below the current organization group (OG). For instance, you can hide 'Asset Number' from the Device List views of the current OG and of all the OGs underneath.

Once all your customizations are complete, select the Accept button to save your column preferences and apply this new column view. You can return to the Layout button settings at any time to tweak your column display preferences.

Some notable device list view custom layout columns include the following.

- SSID (Service Set Identifier or Wi-Fi network name)
- Wi-Fi MAC Address
- Wi-Fi IP Address
- Public IP Address

Exporting List View

Select the Export button to save an XLSX or CSV (comma-separated values) file of the entire Device List View that can be viewed and analyzed with MS Excel. If you have a filter applied to the Device List View, the exported listing reflects the filtered results.

Search in Device List View

You can search for a single device for quick access to its information and take remote action on the device.

To run a search, navigate to Devices > List View, select the Search List bar and enter a user name, device-friendly name, or other device-identifying element. This action initiates a search across all devices, using your search parameter, within the current organization group and all child groups.

Linux Device Details Page

The Linux Device Details page in the Workspace ONE UEM console shows options available for customizing your enrolled Linux devices. Use the Device Details page to review and modify user and device actions.

Device Details

You can access Device Details by selecting a device's Friendly Name from the Device List View, using one of the Dashboards, or with any of the search tools.

From the Device Details page, you can access device information broken into different tabs. Each tab contains related device information, which can vary depending on your Workspace ONE UEM deployment.

- Summary - View general statistics such as enrollment status, compliance, last seen, platform/model/OS, organization group, serial number, power status, storage capacity, physical memory, and virtual memory.
- Profiles - Displays all profiles assigned to the selected device both installed (active) and assigned (inactive).
- Apps - Displays a list of installed apps along with the app status, installation status, and assignment status.
- Sensors – View all sensors assigned to the selected device. This data includes the name, value and last executed date.
- Scripts – Not currently supported for Linux.
- User - Access details about the user of a device and the status of the other devices enrolled to this user.
- Network – View current network (Wi-Fi) status of a device.
- Security - View the last received security information statuses from the device. The Security tab shows if a device is enrolled, if profiles are installed, and the status of certificates. The Security tab also shows the Disk Encryption Status for a device. Passcode and Applications, while shown on the security tab, are not supported for Linux.
- Notes - View and add notes regarding the device. For example, note the shipping status or if the device is in repair and out of commission.
- Certificates – Identify device certificates by name and issuer. This tab also provides information about certificate expiration.
- Terms of Use - View a list of End User License Agreements (EULAs) which have been accepted during the device enrollment.
- Troubleshooting - View Event Log and Commands logging information.
- Status History - View history of device in relation to enrollment status.
- Attachments - Use this storage space on the server for screenshots, documents, and links for troubleshooting and other purposes without taking up space on the device itself.

The More Actions drop-down on the Device Details page is similar to same function on the list view, but with some additional features These actions let you perform remote actions over the air to the selected device. The actions available vary depending on factors such as the device platform, Workspace ONE UEM console settings, and enrollment status.

- Query - This option submits an on-demand request for a device to send updated sample data back to the Console. Selecting the Query option, will submit a request for all of the following. If you would like to request just one of the items, this can be done from the More Actions menu.
  - Device Information - Send an MDM query command to the device to return information on the device such as friendly name, platform, model, organization group, operating system version, and ownership status.
  - Profiles – Send an MDM query command to the device to return a list of the installed device profiles.
  - Certificates – Send an MDM query command to the device to return a list of the installed certificates.
  - Sensors – Send an MDM query command to the device to return updated Sensor values.
- Enterprise Wipe – This option unenrolls, and removes any Wi-Fi and credentials that Workspace ONE UEM pushed down to the device and removes the Workspace ONE Intelligent Hub from the device. This option does not remove custom configuration profiles that were sent to the device, but it runs any defined removal manifests. See Custom Configuration profiles for more information.
- Change Organization Group – Change the device's home organization group to another pre-existing OG. Includes an option to select a static or dynamic OG.
- Manage Tags – Assign a customizable tag to a device, which can be used to identify a special device in your fleet.
- Edit Device – Edit device information such as Friendly Name, Asset Number, Device Ownership and Device Category.
- Delete Device – Delete and unenroll a device from the Workspace ONE UEM console. This action performs an Enterprise Wipe and removes the device from the Workspace ONE UEM console.

Linux Profiles

4

Profiles are the primary means to manage devices. Configure profiles so that your Linux devices remain secure and configured to your preferred settings.

You can think of profiles as the settings and rules that, when combined with compliance policies, help you enforce corporate rules and procedures. They contain the settings, configurations, and restrictions that you want to enforce on devices.

A profile consists of the general profile settings and a specific payload. Profiles work best when they contain only a single payload.

Wi-Fi Profile for Linux

Configuring a Wi-Fi profile lets devices connect to corporate networks, even if they are hidden, encrypted or password protected.

Procedure

1 Navigate to Resources > Profiles & Baselines > Profiles > Add > Add Profile > Linux

2 Configure the profile's General settings as appropriate.

3 Select the Wi-Fi payload.

4 Configure Wi-Fi settings, including:

a

Setting	Description
Service Set Identifier	Provide the name of the network.
Hidden Network	Indicate if the Wi-Fi network is hidden.
Set as Active Network	Indicate if the device will connect to the network with no end-user interaction.

Setting	Description
Security Type	Specify the access protocol used and whether certificates are required. Depending on the selected security type, this will change the required fields. If None or WPA/WPA 2 are selected; the Password field will display. If WPA/WPA 2 Enterprise is selected, the Protocols and Authentication fields display. Protocols - Use Two Factor Authentication SFA Type Authentication - Identity Anonymous Identity Username Password Identity Certificate Root Certificate
Password	Provide the required credentials for the device to connect to the network. The password field displays when WPA/WPA 2 is selected from the Security Type field.
Proxy Type	Enable to configure the Wi-Fi proxy settings.
Proxy Server	Enter the hostname or IP address for the proxy server.
Proxy Server Port	Enter the port for the proxy server.
Exclusion List	Enter the hostnames to exclude from the proxy. Hostnames entered here will not be routed through the proxy. Use the * as a wild card for the domain. For example: *.vmware.com or *vmware.com.

5 Select Save & Publish.

Credential Profile for Linux

For greater security, you can implement digital certificates to protect corporate assets. To do this, you must first define a certificate authority, then configure a Credentials payload alongside your Wi-Fi payload. Each payload has settings for associating the certificate authority defined in the Credentials payload.

Note To install certificates on Linux devices, we utilize the following open source puppet forge module: https://forge.puppet.com/modules/broadinstitute/certs

This module and therefore our support for credentials requires Puppet to be installed on the device and supports the following distributions and versions:

- RedHat 7 & 8
- CentOS 7 & 8
- OracleLinux 7 & 8
- Scientific 8, 9 & 10
- Debian
- Ubuntu
- SuSE

Procedure

1 Navigate to Resources > Profiles & Baselines > Profiles > Add > Add Profile > Linux

2 Configure the profile's General settings as appropriate.

3 Select the Credentials profile and select Configure.

4 Use the drop-down menu to select either Upload or Defined Certificate Authority for the Credential Source. The remaining profile options are source-dependent. If you select Upload, you must enter a Credential Name and upload a new certificate. If you select Defined Certificate Authority, you must choose a predefined Certificate Authority and Template.

5 Select Save & Publish.

Custom Configuration Profile

The Custom Configuration payload can be used to configure your Linux devices with features that Workspace ONE UEM console does not currently support through its native payloads. This payload currently utilizes open source Puppet for this configuration, so nothing other than the free Puppet agent installed on the device to support this functionality.

When a custom configuration profile is assigned to a Linux device, Workspace ONE UEM will pass the manifest to puppet running on the device. Currently, when a device is enterprise wiped or unenrolled, these configuration changes will not be removed from the device unless a removal manifest is defined in the profile.

For more information on Puppet, including sample manifests, please see: http://forge.puppet.com

To validate the syntax of your puppet code, please see: https://validate.puppet.com

Procedure

1 Navigate to Resources > Profiles & Baselines > Profiles > Add > Add Profile > Linux

2 Configure the profile's General settings as appropriate.

3 Select the Custom Configuration profile and select Configure.

4 Configure the payload including:

a

Setting	Description
Name	Populate a name that will distinguish this payload from others.
Enforce Manifest	If checked, the manifest will be reapplied at the data transmit interval configured in Settings > Device & Users > Linux > Intelligent Hub Settings. If left unchecked, the manifest will only be executed once when the profile is initially pushed to the device.
Check for Dependency	If the puppet manifest has a required dependency, it can be included here. For example, "puppetlabs/stdlib"
Install Manifest	Copy and paste the content of your Puppet Manifest here. This manifest will be implemented on the device assigned in the general tab.
Remove Manifest	This manifest will be executed on the device when this profile is unassigned from a device. If this manifest is left blank, when a custom configuration profile is removed from a device, the action dictated by the install manifest will remain on the device.

5 Select Save & Publish

Custom Configuration Examples

Puppet Manifest Examples

Although we encourage you to learn and explore Puppet if you are interested in creating custom configuration profiles, to get you started, following are examples of puppet code that can be used on standard Ubuntu. They will not work on other distributions of Linux.

Install Chrome Browser on Ubuntu:

- Dependency: None
- Installation Manifest:

```
file { 'google-chrome-stable_current_amd64.deb': source => 'https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb', path => '/tmp/google-chrome-stable_current_amd64.deb', ensure => present, } exec { 'install-chrome': command => '/usr/bin/dpkg -i /tmp/google-chrome-stable_current_amd64.deb', logoutput => true, }
```

- Removal Manifest:

```
package { 'google-chrome-stable': ensure => 'absent', }
```

Disabling SSH Server on Ubuntu:

- Dependency: `puppetlabs-stdlib`
- Installation Manifest:

```
service { 'ssh': name => 'sshd', ensure => false, enable => false,}
```

Removal Manifest:

```
service { 'ssh': name => 'sshd', ensure => true, enable => true,}
```

Sensors for Linux Based Devices 5

Linux based devices contain multiple attributes such as hardware, certificates, patches, apps, and more. With Sensors, you can collect data for these attributes using the Workspace ONE UEM console and display the data returned by Sensors in Workspace ONE.

Linux devices have a huge number of attributes associated with them. This number increases when you track the different apps, distro versions, patches, and other continually changing variables. It can be difficult to track all of these attributes.

Workspace ONE UEM tracks a limited number of device attributes by default. However, with Sensors, you can track any specific device attributes needed. For example, you can create a Sensor that tracks the number of battery charge cycles, last updated date of a virus definition file, or the build version of a specific security agent. Sensors allow you to track various attributes across your devices using Bash. For your Linux devices, these sensor scripts can be configured to run periodically.

Find Sensors in the main Workspace ONE UEM console navigation under Resources.

Workspace ONE UEM Options

- Bash Scripts - The Bash script you create determines the value of each sensor. For examples of what scripts you can create, see Bash Examples.
- Support for Variables - If your sensor script requires dynamic or sensitive information that must be defined outside of the script, variables can be used to securely store this information. Variable data is encrypted at-rest and in-transit. The variables can be referenced in the code directly by name $myvariable.
- Sensors Triggers - When configuring Sensors, you can control when the device reports the sensor data back to the Workspace ONE UEM console with triggers. Currently, for Linux devices, you can schedule these triggers based on the Intelligent Hub Sample Schedule (periodically). Other triggers are currently not available.
- Device Details > Sensors - You can see data for single devices on the Sensors tab in a device's Device Details page.

Workspace ONE Intelligence Options

If you use the Workspace ONE Intelligence service, you can run a report or create a dashboard to view and interact with the data from your Sensors. When you run reports, use the Workspace ONE UEM category, Device Sensors. You can find your sensors and select them for queries in reports and dashboards.

For details on how to work in Workspace ONE Intelligence, see VMware Workspace ONE Intelligence Products.

Creating Sensors for Linux Devices

Create Sensors in the Workspace ONE UEM console to track specific device attributes such as remaining battery, specific version or build information, or average CPU usage. Each sensor includes a script of code to collect the desired data. You can upload these scripts or enter them directly into the console.

For Linux devices, Sensors use Bash scripts to gather attribute values. You must create these scripts yourself either before creating a sensor or during configuration in the scripting window.

Note To view Sensors for multiple devices and interact with the data in reports and dashboards, you must opt into VMware Workspace ONE Intelligence. If you want to view Sensors data for a single device, you do not need VMware Workspace ONE Intelligence. Go to the device's Device Details page and select the Sensors tab to view the data.

1 In the Workspace ONE UEM console, navigate to Resources > Sensors.

2 On the Sensors page, click Add and select Linux.

3 In the New Sensor page, navigate to General > Name and enter the following:

a

Setting	Decription
Name	Enter the name of the sensor. The name must start with a lowercase letter followed by alpha-numeric characters and underscores. The name must be between 2 and 64 characters.
Description	Enter the description of the sensor.

4 Click Next.

5 Configure the sensor settings in the Details tab.

a

Setting	Description
Language	Select the language. Currently only Bash is support for Linux devices.
Execution Context	This setting controls the context with which the script runs. Currently only System is support for Linux devices.
Response Data Type	Select the type of response the script will return. You can choose between: ■ String ■ Integer ■ Boolean ■ Date Time
Code	Upload a script for the sensor or write your won in the text box provided.

6 Click Next.

7 In the Variables tab, you can optionally define variable names and values to use in your Sensor script. These variables are securely stored, encrypted at-rest, and only used temporarily during script execution in the scripting environment.

Variables support static text or UEM lookup values. The lookup values are resolved before being delivered to the device for execution.

Bash scripts can reference the variables directly by name from the environment like $myvariable.

8 Click Save or Save and Assign.

You can save the sensors information and go back to menu or can move to the Assignment page to add sensors to a smart group.

What to do next

To add a sensor to a smart group, perform the following steps:

1 In the New Assignment page, enter the Assignment Name and Select Smart Group. Click Next

2 In the Deployment page, currently, the only option available is Periodically. The script will run periodically based on the Intelligent Hub Data Transmit Interval configured in All Settings > Device & Users > Linux > Intelligent Hub Settings

3 Click Save.

After the assignment group is saved, you can prioritize the assignments if multiple smart groups are configured with potentially overlapping sets of devices. Once this step is done, devices with Intelligent Hub installed will receive the Sensor configurations on the next check-in. Intelligent Hub then runs the Sensors and reports the data back to Workspace ONE.

View Sensors in Linux Device Details

Sensor data can be viewed in the Workspace ONE UEM console in Device Details > Sensors tab.

1 In the Workspace ONE UEM console, navigate to Device > Details View and select the Sensors tab. The following details are displayed in the Sensors tab:

   a Name - Name of the Sensor.

   b Value - Value reported by the device.

   c Last executed date - The timestamp for when the Sensor value was collected.

2 To request the device to on-demand, run the Sensor and report the value back, select a Sensor name, and click Run.

   a **Note** Run button is displayed in Device Details only if the Hub version is supported. The minimum supported Linux Hub version is 21.05.

3 To view information about Sensors execution, navigate to Details View > Troubleshooting. In the event log filters, select Sensors.

   a **Note** This is seen only if the event log level is set to capture information or debug messages.

Linux Sensor Examples

When you create Sensors for Linux devices, you must upload a bash script or enter the bash script in the text box provided during configuration in the Workspace ONE UEM console. These commands return the values for the sensor attributes.

Bash Examples: The following examples contain the settings and the code needed for standard Ubuntu and may not work on other Linux distributions.

Get the current Host Name:

- Language: Bash
- Execution Context: System
- Response Data Type: String

```
cat /proc/sys/kernel/hostname |
```

Get a list of all users currently logged in:

- Language: Bash
- Execution Context: System
- Response Data Type: String

```
who | cut -d' ' -f1 | sort | uniq
```

Get current distribution version:

- Language: Bash
- Execution Context: System
- Response Data Type: String

```
( lsb_release -ds || cat /etc/*release || uname -om ) 2>/dev/null | head -n1 |
```

Determine if an SSH Server is running:

- Language: Bash
- Execution Context: System
- Response Data Type: Integer

```
ps -ef | grep sshd | grep -v "grep" | wc -l
```

Note The returned value equates to the number of SSH daemons running on the endpoint