Mosyle

aruba
a Hewlett Packard Enterprise company
ClearPass
Change Log

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modified By</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-01</td>
<td>November 2019</td>
<td>Anish Pansare</td>
<td>Initial Release supporting Mosyle Manager</td>
</tr>
<tr>
<td>2020-01</td>
<td>February 2020</td>
<td>Danny Jump</td>
<td>Adding support for Mosyle Business</td>
</tr>
</tbody>
</table>

Copyright

©Copyright 2019 Hewlett Packard Enterprise Development LP.

Open Source Code

This product includes code licensed under the GNU General Public License, the GNU Lesser General Public License, and/or certain other open source licenses. A complete machine-readable copy of the source code corresponding to such code is available upon request. This offer is valid to anyone in receipt of this information and shall expire three years following the date of the final distribution of this product version by Hewlett-Packard Company. To obtain such source code, send a check or money order in the amount of US $10.00 to:

Hewlett-Packard Company
Attn: General Counsel
3000 Hanover Street
Palo Alto, CA 94304
USA

Please specify the product and version for which you are requesting source code. You may also request a copy of this source code free of charge at HPE-Aruba-gplquery@hpe.com.
Contents

Introduction and Overview .............................................................................................................. 5
Software Requirements .................................................................................................................... 5
Installation and Deployment Guide .................................................................................................. 5
Pictorial View of the Integration ...................................................................................................... 6
Access to the Extension Store ......................................................................................................... 7
New Extension Support in ClearPass 6.7+ ...................................................................................... 7
  Extensions and IP address configuration support .......................................................................... 7
  Extensions and web proxy support ............................................................................................... 7
Mosyle Extension Installation and Configuration ............................................................................ 10
  Mosyle configuration .................................................................................................................... 19
  Available Data Fields .................................................................................................................... 19
Appendix A – Additional Diagnostics and Support ........................................................................ 20
  The Extensions Service ................................................................................................................ 20
  Extension logs and debugging ....................................................................................................... 20
  Accessing extension logs within ClearPass ‘Collect Logs’ ............................................................ 21

© 2019 Hewlett Packard Enterprise Development LP. All Rights Reserved.

ClearPass and Mosyle - Integration Guide
Figures

Figure 1: Pictorial view of ClearPass Policy Manager integration with Mosyle .......................................................... 6
Figure 2: Entering HPE Passport credentials ........................................................................................................... 7
Figure 3: Extension framework GUI ......................................................................................................................... 8
Figure 4: Defining the base IP SUBNET and LOCALHOST for the Extensions framework ........................................... 8
Figure 5: Extensions framework GUI ......................................................................................................................... 10
Figure 6: GUI Extension search .................................................................................................................................. 10
Figure 7: GUI Extension install ................................................................................................................................. 11
Figure 8: GUI Extension configuration at install time .................................................................................................. 11
Figure 9: Extension configuration parameters .......................................................................................................... 12
Figure 10: Creating an access-token in Mosyle Manager, part1 .................................................................................. 13
Figure 11: Creating an access-token in Mosyle Business, part2 ............................................................................... 14
Figure 12: Example of a created accessToken .......................................................................................................... 14
Figure 13: Creating an Admin user on ClearPass ..................................................................................................... 15
Figure 14: GUI review and setting the Extension configuration ................................................................................ 15
Figure 15: Log validation ............................................................................................................................................. 16
Figure 16: Attributes fetched ...................................................................................................................................... 16
Figure 17: Example of an Enforcement Policy utilizing attributes returned from Mosyle ....................................... 17
Figure 18: Example Context Server Action .............................................................................................................. 18
Figure 19: Checking on the extensions service and how to start/stop the service ...................................................... 20
Figure 20: Using the GUI to change the DEBUG level ............................................................................................... 20
Figure 21: Extension logs location in 'Collect Logs' diagnostic GZ file ..................................................................... 21
Introduction and Overview

Mosyle provides the most flexible and best-in-class mobile device management (MDM) solution for Apple deployments of any size, complexity and industry. Mosyle solutions simplifies initial setup, provisioning and management of your iPad and iPhone devices, Mac computers and Apple TV. Mosyle provides two different solutions: Manager and Business. Starting from v2 of the Extension, we've added support for Mosyle Business which complements our initial support for Mosyle Manager.

Mosyle Manager was designed exclusively for K-12 schools and districts to incorporate education-focused features and hierarchy, with areas for students, class periods and course information. It also integrates directly with Apple School Manager. In addition, Mosyle Manager will offer Classroom Management capabilities, so teachers can manage student-device interactions within the classroom.

Mosyle Business, on the other hand, was built for enterprise and higher education institutions, and it will have a different hierarchy, as well as integrate with both Apple Business Manager and Apple School Manager. It allows companies to experience the most secure and compliant experience when automating and managing Apple devices to their end-users to be more productive at work.

This integration guide covers the deployment and configuration of ClearPass Policy Manager to interface with Mosyle.

It provides the ability to pull all device details from a customer tenant into ClearPass endpoints database. The contextual data inserted into Policy Manager can then be leveraged to drive granular access policy.

Software Requirements

The minimum software version required for ClearPass is 6.7.2. ClearPass runs on hardware appliances with pre-installed software or as a Virtual Machine under the hypervisors. Hypervisors that run on a client computer such as VMware Player are not supported. Supported Hypervisors are listed under release notes:


Interoperability validation is done with ClearPass version 6.8.2.

Installation and Deployment Guide

The generic ClearPass installation and deployment guide is located here:

Pictorial View of the Integration

The diagram below shows a pictorial overview of the components and how they interact with each other.

Figure 1: Pictorial view of ClearPass Policy Manager integration with Mosyle
Access to the Extension Store

Access to the Extension Store to download extensions was simplified in ClearPass 6.7. The ability to download extensions from the store and to validate support entitlement for access to the Software Updates Portal (e.g. Posture & Profile Data Updates, Software Updates, & Skins) now uses the HPE Passport account credentials that are associated with the customers’ ClearPass licenses. This is configured where previously the subscription-id was defined, under Administration -> Agents and Software Updates -> Software Updates as shown below. Ensure you enter your HPE Passport credentials to enable Extension download capabilities.

**Figure 2: Entering HPE Passport credentials**

New Extension Support in ClearPass 6.7+

With the release of 6.7, several new features have been added to enhance the functionality of the extension framework. Previously, all extension installation and operation tasks required use of the API Explorer to interoperate with the Extension and the underlying framework. Now this functionality has been exposed with a new GUI. The GUI is accessed from within the Guest UI and is shown below, Administration -> Extensions.

**Extensions and IP address configuration support**

The other major additions in the 6.7 release are the ability to define the extension framework base IP network and statically define the IP address of the individual extensions. The latter being useful when deploying extensions in a cluster and the requirement for a fixed IP address for the same extension across a cluster regardless of which ClearPass node or nodes it is installed on.

**Extensions and web proxy support**

Prior to 6.7 support for web proxy was limited to the installation of the extensions. Starting in ClearPass 6.7, extensions now support communications with 3rd parties via a web proxy. This adds incremental web proxy functionality. If a web proxy is defined in ClearPass Policy Manager, then an extension will use that configuration.

---

*The Policy Manager web proxy configuration is ONLY read by the extension at installation time. If the web proxy configuration is changed in Policy Manager, then the extension must be re-installed so the new settings are re-read and bonded to the extension.*
Configuring the base Extension IP subnet, this is defined within Policy Manager as shown below under Administration -> Server Manager -> Server Configuration [chose your node] Service Parameters [ClearPass system service]. The default is 172.17.0.1/16, this address is the non-routed address of the ClearPass node itself. The IP addresses range for the extensions are based upon the network prefix used.

Note that the subnet defined here for the extension framework must be one of the following 10.0.0.0/8, 172.16.0.0/12 or 192.168.0.0/16, i.e. RFC 1918.

Changing the extension base IP address will require the extension service to be restarted.
Changing the “Extensions Network Address” range is necessary if either the MGMT or DATA interface are also using an address in the extension default range of 172.17.x.x/16. Set the new network address range as needed and restart the extension service for this to take effect, else this does not need to be modified.
Mosyle Extension Installation and Configuration

Starting in ClearPass 6.7, a Graphical User Interface (GUI) was introduced to make the process of interacting with the extension framework easier. To access the extension GUI, from the Guest System, under Administration find the Extension User Interface as shown below.

Figure 5: Extensions framework GUI

From here, click on ‘Install Extension’, and the search box below appears. Enter the store ID and click on Search.

Figure 6: GUI Extension search

Starting 6.7, in a cluster environment an extension can be installed on the subscriber nodes directly.
Click on the Extension and then the **Install** option.

**Figure 7: GUI Extension install**

Set a specific IP address for the Extension if required. It will automatically pick an IP address if not assigned.

**Figure 8: GUI Extension configuration at install time**

After the Extension has been installed, review the configuration and adjust as needed. There are options to Start, Delete, Reinstall or Show Logs and the option to edit and set the Extension configuration.
The default configuration used for Extension is below:

```json
{
    "logLevel": "INFO",
    "verifySSLCerts": true,
    "mosyleHost": "managerapi.mosyle.com",
    "isMosyleBusiness": false,
    "accessToken": "...",
    "osTypes": ["ios", "mac", "tvos"],
    "dataColumns": ["wifi_mac_address", "ethernet_mac_address", "os", "device_name", "device_model", "device_type", "model_name", "osversion", "status", "is_supervised", "is_deleted", "date_checkin", "date_info"],
    "syncAllOnStart": true,
    "syncAllSchedule": "0 2 * * 6",
    "cppmUserName": "username",
    "cppmPassword": "password"
}
```

Each of the attributes are explained in the table below in detail.

**Figure 9: Extension configuration parameters**

<table>
<thead>
<tr>
<th>Configuration attribute</th>
<th>Description</th>
<th>Example/Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>mosyleHost</td>
<td>Mosyle Host URL</td>
<td>managerapi.mosyle.com businessapi.mosyle.com</td>
</tr>
<tr>
<td>accessToken</td>
<td>Token to leverage Mosyle API</td>
<td>000000</td>
</tr>
<tr>
<td>isMosyleBusiness</td>
<td>Determines is the Extension is communicating with Mosyle Manager or Mosyle Business</td>
<td>true or false {added in V2 of the extension}</td>
</tr>
<tr>
<td>dataColumns</td>
<td>Specify what information to be synced from Mosyle system. For all, enter “*”</td>
<td>device_name, device_model</td>
</tr>
<tr>
<td>osTypes</td>
<td>Specify OS types to be fetched from Mosyle system.</td>
<td>ios, mac, tvos</td>
</tr>
<tr>
<td>syncAllOnStart</td>
<td>If this option is set to true, when the extension starts, the system will attempt to sync all computers in the Mosyle system to ClearPass.</td>
<td>true or false</td>
</tr>
<tr>
<td>syncAllSchedule</td>
<td>The schedule for when the Sync All Endpoints process should run. Note: This uses CRON type scheduling.</td>
<td>0 2 * * 6</td>
</tr>
<tr>
<td>cppmUserName</td>
<td>The user name of an Admin user in ClearPass. This is used for device profiling</td>
<td>Administrator username</td>
</tr>
<tr>
<td>cppmPassword</td>
<td>The password for the user entered in cppmUserName</td>
<td>Administrator password</td>
</tr>
<tr>
<td>verifySSLCerts</td>
<td>Should SSL certificates be validated when communicating with Mosyle.</td>
<td>true or false</td>
</tr>
<tr>
<td>logLevel</td>
<td>Logging level for troubleshooting</td>
<td>&quot;DEBUG&quot;, &quot;INFO&quot;, &quot;WARN&quot;, &quot;ERROR&quot;</td>
</tr>
</tbody>
</table>

Starting in V2 of the Extension, support for Mosyle Business was added. This additional support added one new switch in the Extension configuration, **isMosyleBusiness**, which is a boolean true or false option and very self-explanatory, finally one other very important configuration option, see below.
Depending on if you're using the Extension to communicate with Mosyle Manager or Mosyle Business will also dictate the setting for the setting `mosyleHost`.

<table>
<thead>
<tr>
<th>Configuration Deployment Mode</th>
<th>Value for <code>mosyleHost</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mosyle Manager</td>
<td><code>managerapi.mosyle.com</code></td>
</tr>
<tr>
<td>Mosyle Business</td>
<td><code>businessapi.mosyle.com</code></td>
</tr>
</tbody>
</table>

When leveraging the sync capabilities of the Extension to get the device attributes from Mosyle, specify `syncAllSchedule`. This attribute is leveraged for periodic sync of endpoints from Mosyle. The `syncAllOnStart` attribute can be leveraged as well to sync everything upon the start or re-start of the extension.

To obtain the `accessToken`, if you're using Mosyle Manager raise a support ticket with Mosyle. They will generate the token. Use this token in the Mosyle extension config under the key-value-pair of `accessToken`. Be sure to keep a safe copy of this.

If you're using Mosyle Business, there is a self-service option within the UI. Navigate from the GUI to ‘Organization’, then click on ‘API Integration’ as shown below.

**Figure 10: Creating an access-token in Mosyle Manager, part 1**

![API Integration](image)

then ‘Add new token’ on the top right-hand-side. Name the Profile with an appropriate name, perhaps ‘clearpass’. 

**NOTE**

Ensure you de-select “Require an user credential (email and password) on API requests” when creating the token.
The token will be created (example below), record this as it will be need in the extension configuration under **accessToken**.

**Figure 12: Example of a created accessToken**

There is a wide range of available information from the Mosyle system. By default, we only ingest a subset of the fields, but you can add or change them using the **dataColumns** configuration value. If you wish to include all fields simply add an asterisk ("**") to the dataColumns list.

The **cppmUserName** and **cppmPassword** should be for a ClearPass Network Administrator level account. The device profiling attributes obtained from Mosyle need to be written into the endpoint repository leveraging the REST APIs which requires a Network Administrator account.

A ClearPass Network Administrator account can be created under **Administration > Users and Privileges > Admin Users**. Click on **Add**. A user with the following **Privilege Level** needs to be created.
A Network Administrator privilege level is sufficient for the action of adding device profiling information into the endpoint database of ClearPass.

A copy of the Mosyle Extension with the desired configuration is shown below, this has to be modified for your deployment. Include accessToken, dataColumns, osTypes, cppmUserName and cppmPassword that will be specific to your environment.

Change or include any other values based on the description of each in the above table. Select Restart and click on Save Changes to restart the extension.

Figure 14: GUI review and setting the Extension configuration
After the configuration and the restart of the extension, click on **Show Logs**.

**Figure 15: Log validation**

![Log validation](image)

The above log shows that the Extension has synced the endpoint data from Mosyle. The above steps sync endpoint details at the time of starting the Extension and then regular updates are fetched at the interval specified by `syncAllSchedule`. Please note the IP address of the Extension. This will be used in the next section where we leverage the Mosyle attributes during enforcement. A sample of endpoint attributes fetched are shown below.

**Figure 16: Attributes fetched**

![Attributes fetched](image)
Multiple use-cases exist for how the data that is returned from Mosyle can be used in the Policy enforcement. In the example below, we are performing multiple checks:

1. Check if the device’s status. If it is not, redirect it to the captive Portal. Create a helpdesk ticket for remediation.
2. If the device is deleted in Mosyle, create a helpdesk ticket.
3. If the device is not supervised by Mosyle, put it in the Quarantine VLAN.
4. Check when the device last checked-in. If the device has not checked in for certain number of days, flag it and redirect it to the Captive Portal.

**Figure 17: Example of an Enforcement Policy utilizing attributes returned from Mosyle**

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Endpoint: Mosyle status NOT_EQUALS INSTALLED)</td>
<td>Mosyle - Create Helpdesk Ticket, Mosyle - Redirect to Captive Portal</td>
</tr>
<tr>
<td>(Endpoint: Mosyle is_supervised EQUALS false)</td>
<td>Mosyle - Quarantine VLAN</td>
</tr>
<tr>
<td>(Endpoint: Mosyle device_type EQUALS SMARTPHONE) AND (Endpoint: Mosyle os EQUALS ios) AND (Endpoint: Mosyle version CONTAINS 9.3.5)</td>
<td>Mosyle - Non Compliant, Mosyle - Create Helpdesk Ticket</td>
</tr>
<tr>
<td>(Endpoint: Mosyle date_checkin NOT_EXISTS )</td>
<td>Mosyle - Quarantine VLAN, Mosyle - Redirect to Captive Portal</td>
</tr>
<tr>
<td>(Endpoint: Mosyle is_deleted NOT_EQUALS false)</td>
<td>Mosyle - Create Helpdesk Ticket, Mosyle - Redirect to Captive Portal</td>
</tr>
</tbody>
</table>

Different companies will have different enforcement profiles and policies. The key takeaway here is to leverage the attributes received from Mosyle to drive the policy engine into making different enforcement actions for the device as they authenticate on the network.
Required Context Server Action can be defined under Administration > Dictionaries > Context Server Actions.

**Figure 18: Example Context Server Action**

Set the HTTP Method to **GET**. There is no additional configuration required in Header, Content and Attributes tab.
Mosyle configuration

It is assumed that a working Mosyle environment is leveraged for this extension. The configuration of Mosyle is beyond the scope of this guide.

To obtain the access-token required, raise a support ticket with Mosyle. They will generate the token. Use this token in the Mosyle extension config under the key-value-pair of access-token. Be sure to keep a safe copy of this.

Available Data Fields

There is a wide range of available information from the Mosyle system. By default, we only pull a subset of the fields, but you can add or change them using the dataColumns configuration value. If you wish to include all fields simply add an asterisk (“*”) to the dataColumns list.

If the “is_deleted” field is true we will set the ClearPass Endpoint status to "Unknown". For all other endpoints the status will be "Known".

All fields added to endpoints are prefixed with "Mosyle ", for example, "Mosyle is_deleted".

Below are the available data fields:

- deviceudid
- total_disk
- os
- serial_number
- device_name
- device_model
- battery
- osversion
- date_info
- carrier
- roaming_enabled
- isroaming
- imei
- meid
- available_disk
- wifi_mac_address
- bluetooth_mac_address
- is_supervised
- date_app_info
- date_last_beat
- date_last_push
- status
- isActivationLockEnabled
- isDeviceLocatorServiceEnabled
- isDoNotDisturbInEffect
- isCloudBackupEnabled
- isNetworkTethered
- needosupdate
- productkeyupdate
- device_type
- lostmode_status
- is_muted
- date_muted
- activation_bypass
- date_media_info
- tags
- is_deleted
- iTunesStoreAccountHash
- iTunesStoreAccountIsActive
- date_profiles_info
- ethernet_mac_address
- model_name
- LastCloudBackupDate
- SystemIntegrityProtectionEnabled
- BuildVersion
- LocalHostName
- HostName
- OSUpdateSettings
- ActiveManagedUsers
- CurrentConsoleManagedUser
- date_printers
- AutoSetupAdminAccounts
- appleTVid
- asset_tag
- ManagementStatus
- OSUpdateStatus
- AvailableOSUpdates
- appleTVid
- enrollment_type
- userid
- date_checkin

Bold fields are required and are always returned.
Appendix A – Additional Diagnostics and Support

The Extensions Service

The ClearPass extension is supported by a new system service that was initially added in 6.6, but we recommend a minimum version of 6.7.2. This service should be running. Note that restarting this service will affect all deployed and running extensions.

To check on the state and to restart the service, go to Administration > Server Manager > Server Configuration [select a ClearPass node] > Service Control. From here start/stop the extension service. By default, this service is automatically started.

*Figure 19: Checking on the extensions service and how to start/stop the service*

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Status</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. AirGroup notification service</td>
<td>Running</td>
<td>Stop</td>
</tr>
<tr>
<td>2. Asyc DB write service</td>
<td>Running</td>
<td>Stop</td>
</tr>
<tr>
<td>3. Asyc network services</td>
<td>Running</td>
<td>Stop</td>
</tr>
<tr>
<td>4. ClearPass IPsec service</td>
<td>Running</td>
<td>Stop</td>
</tr>
<tr>
<td>5. DB change notification server</td>
<td>Running</td>
<td>Stop</td>
</tr>
<tr>
<td>6. DB replication service</td>
<td>Running</td>
<td>Stop</td>
</tr>
<tr>
<td>7. Extensions service</td>
<td>Running</td>
<td>Stop</td>
</tr>
</tbody>
</table>

Extension logs and debugging

Referencing the configuration previously used, adjust the `logLevel` to ‘DEBUG’. In the new 6.7, GUI change the configuration and restart the extension as shown below. Logs can then be viewed from the ‘Show Logs’.

*Figure 20: Using the GUI to change the DEBUG level*

Remember after changing the logging level, the extension will need to be restarted for this change to take effect.
**Accessing extension logs within ClearPass ‘Collect Logs’**

In addition to the logging of messages that be examined in the extension as shown above, it’s possible to configure the extension to log messages so that they can be collected and examined via the Policy Manager ‘Collect Logs’ system function. This is extremely useful for Aruba TAC. The logs are available under Administration > Server Manager > Server Configuration > Collect Logs.

If there is a requirement for Aruba TAC to investigate a system issue, one of the items they regularly ask for is the system logs to aid with their diagnostic investigation. The ClearPass extension can write its logs such that they are available and can be collected with all other system diagnostics information when the ‘Collect Logs’ function is run. Remember that by default, the logLevel is set to INFO but TRACE, DEBUG, INFO, WARN, ERROR, FATAL can also be set. Any of the levels will display the information for the selected state and lower. For example, if INFO is selected, it will show messages for INFO, WARN, ERROR, FATAL.

After the Logs have been collected and exported from the system, expand the GZ file and locate the extension logs in the following location ‘PolicyManagerLogs->extension’ as shown below.

*Figure 21: Extension logs location in ‘Collect Logs’ diagnostic GZ file*