

mon_config_service-20.1-HF1-bundle - Release Notes

Artifacts

Hotfix: mon_config_service-20.10-HF1-bundle

1. Release Date: 6th May
2. Applicable for: UIM 20.1

This zip bundle contains the following UIM hotfix packages related to mon_config_service and their respective readme files:

- mon_config_service-20.10-HF1.zip
- mon_config_service_cli-20.10-HF1.zip
- mon_config_service_ws-20.10-HF1.zip

This hotfix addresses the following:

- F819440 : BNPP: Implement SSR's "foreach profile" feature in MCS
- F81998 : Augment UIM API portfolio with Missing SSR/MCS API's

Steps to apply the current patch:

- Read the respective package readme files

Related patches:

- The following hotfixes are related and need to be applied. Refer to the respective release notes.
 - 1) ump_usm-20.1-HF1.zip
 - 2) uimapi-20.1-HF1.zip

Deployment instructions

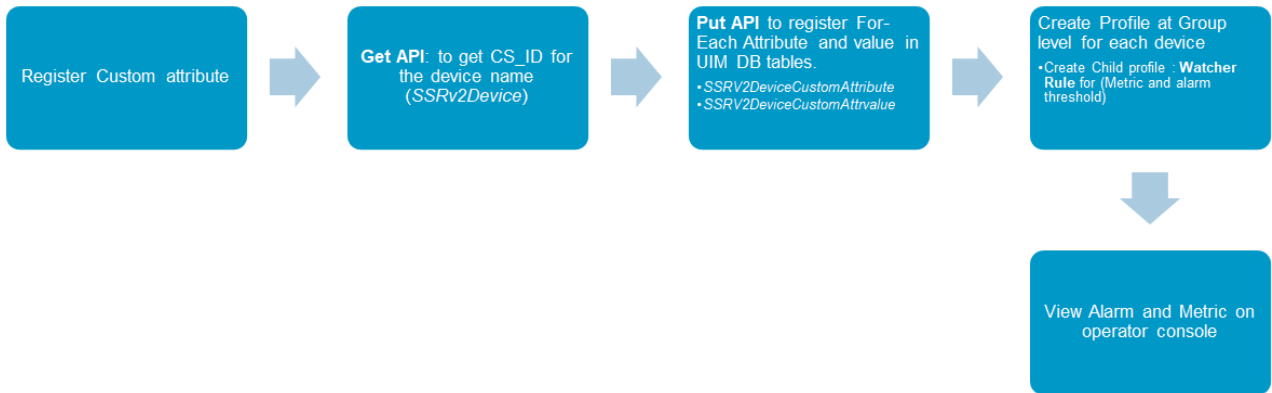
These artifacts are to be applied on UIM 20.1. Perform the deployment of the packages in the following order:

1. Deploy mon_config_service-20.10-HF1.zip , mon_config_service_ws-20.10-HF1.zip,mon_config_service_cli-20.10-HF1.zip (Please refer the respective release notes)
2. Deploy ump_usm-20.1-HF1.zip on UMP system (Please refer the respective release notes)
3. Deploy uimapi-20.1-HF1.zip on UIM server machine (Please refer the respective release notes)

Feature details

For-each feature will be available in usm portlet of ump portal under monitoring tab.

1. For-Each Functionality caters Bulk deployment of profiles at a group level based on custom attribute for each of the devices available in SSRV2DeviceCustomAttribute table.
 1. Ex: If a device in a group have one instance of oracle, on another device in the same group we have 2 instance of oracle and so on.
 - b. As per existing feature , user can create for-each profile at group level. While creating profile user needs to specify on which custom attribute user want to iterate over.
 - c. For example in above case user selects oracle device instance(device.custom.oracleInstance) variable, profile should be created for every instance of oracle on this device in this group, if device have 10 instances it will create 10 profiles, if a device has 1 instance it will create 1 profile and so on.
2. The following steps are needed for using the for-each feature



1. Register the customer attribute in the database and Presence of device custom attributes for a device in database on which the values can be iterated to create profile should be present.
2. Device custom attributes can be present in ssrv2devicecustomattribute, ssrv2devicecustomattrvalue tables.
3. Device custom attributes can be created using manual sql scripts or uimapi (<http://<uimserver>/uimapi>) under DeviceOperations(/PUT /deviceoperations/{identifier}) .
 1. sample device payload to insert with custom attributes is given below as an example which can be executed through UIMAPI.
 2. Perform GET API call to get the device payload using name (GET /deviceoperations/{identifier})
 3. Modify the payload and add the custom attributes elements. Sample payload is shown below.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<device>
<cs_id>1</cs_id>
<device_id>1</device_id>
<csdev_id>D86221456EEA5CF0ABEE4A804860767FF</csdev_id>
<name>pp671588-win1</name>
<status>modified</status>
<custom1>device.name</custom1>
<origin>pp671588-win1_hub</origin>
<os_type>Windows</os_type>
<nimbus_type>2</nimbus_type>
<ip>10.17.165.209</ip>
<cs_type>A</cs_type>
<customattribute>
<name>custom.instance</name>
<scope>default</scope>
<value>4</value>
<value>D:\testdr</value>
<encrypted>true</encrypted>
</customattribute>
<customattribute>
<name>custom.pandu.instance</name>
<scope>default</scope>
<value>5</value>
<value>E:\testdr</value>
<encrypted>true</encrypted>
</customattribute>
<customattribute>
<name>device.usertag1</name>
<scope>default</scope>
<encrypted>true</encrypted>
<modified>0</modified>
  
```

```
</customattribute>
<customattribute>
<name>device.usertag2</name>
<scope>default</scope>
<encrypted>true</encrypted>
<modified>0</modified>
</customattribute>
<customattribute>
<name>device.origin</name>
<scope>default</scope>
<value>pp671588-win1_hub</value>
<encrypted>true</encrypted>
<modified>0</modified>
</customattribute>
<customattribute>
<name>device.ipaddress</name>
<scope>default</scope>
<value>10.17.165.209</value>
<encrypted>true</encrypted>
<modified>0</modified>
</customattribute>
<customattribute>
<name>device.name</name>
<scope>default</scope>
<value>pp671588-win1</value>
<encrypted>true</encrypted>
<modified>0</modified>
</customattribute>
<customattribute>
<name>device.os_type</name>
<scope>default</scope>
<value>Windows</value>
<encrypted>true</encrypted>
<modified>0</modified>
</customattribute>
<deviceattribute>
<name>Origin</name>
<scope>default</scope>
<value>pp671588-win1_hub</value>
</deviceattribute>
<deviceattribute>
<name>RobotInstanceId</name>
<scope>default</scope>
<value>D86221456EEA5CF0ABEE4A804860767FF</value>
</deviceattribute>
<deviceattribute>
<name>DisplayAlias</name>
<scope>default</scope>
<value>pp671588-win1</value>
</deviceattribute>
<deviceattribute>
<name>PrimaryIPV4Address</name>
<scope>default</scope>
```

```
<value>10.17.165.209</value>
</deviceattribute>
<deviceattribute>
<name>typeName</name>
<scope>default</scope>
<value>ComputerSystem</value>
</deviceattribute>
<deviceattribute>
<name>PrimaryRole</name>
<scope>default</scope>
<value>Host</value>
</deviceattribute>
<deviceattribute>
<name>Roles</name>
<scope>default</scope>
<value>Device</value>
<value>Host</value>
</deviceattribute>
<deviceattribute>
<name>label</name>
<scope>default</scope>
<value>pp671588-win1</value>
</deviceattribute>
<deviceattribute>
<name>PrimaryOSType</name>
<scope>default</scope>
<value>WindowsServer-2012-R2</value>
</deviceattribute>
<deviceattribute>
<name>RobotName</name>
<scope>default</scope>
<value>pp671588-win1</value>
</deviceattribute>
<deviceattribute>
<name>CorrelationNames</name>
<scope>default</scope>
<value>pp671588-win1</value>
</deviceattribute>
<deviceattribute>
<name>DisplayName</name>
<scope>default</scope>
<value>pp671588-win1</value>
</deviceattribute>
<deviceattribute>
<name>CorrelationId</name>
<scope>default</scope>
<value>D86221456EEA5CF0ABEE4A804860767FF</value>
</deviceattribute>
<deviceattribute>
<name>OSDescription</name>
<scope>default</scope>
<value>Service Pack 0 Build 9600</value>
</deviceattribute>
```

```

<deviceattribute>
<name>PrimaryOSVersion</name>
<scope>default</scope>
<value>6.3.9600</value>
</deviceattribute>
<deviceattribute>
<name>PrimaryMacAddress</name>
<scope>default</scope>
<value>00-50-56-8D-E9-0A</value>
</deviceattribute>
<deviceattribute>
<name>RobotVersion</name>
<scope>default</scope>
<value>9.2</value>
</deviceattribute>
</device>

```

4. Perform PUT operation /PUT /deviceoperations/{identifier} using the modified payload containing the device custom attributes.
 5. The device custom attributes and its values are populated in ssrv2devicecustomattribute, ssrv2devicecustomattrvalue tables respectively.
3. As an example, we use logmon template for monitoring the log file containing text "error" and the alarm should be raised when the text "error" is found in the file .
 4. For a device there are three entries in ssrv2devicecustomattribute, ssrv2devicecustomattrvalue tables and we are monitoring log files /root/Test/log1.txt, /root/Test/log2.txt, /root/Test/log3.txt. The for-each profile is created for each of the values of log file directories with custom attribute {device.logDir} defined in database.
 5. Once For-each profile is created at group level, the device attribute values selected are iterated to create profiles at device level and alarms are raised matching the threshold.
 6. The sequence of steps executed by user are provided below.
 - Scenario :
 - ii) User monitors all log files on a particular device. Notify user if "error" string exist in each log file,
 - iii) Metric : count of String "Error"
 - iv) Alert : Count > 0

For-Each Profile Creation

- User clicks on Log monitoring(Enhanced) profile under monitoring tab of a group and clicks + icon. User selects for-each deployment as 'yes' .

For-Each Deployment

Enable

Yes | ▼

For-Each Group Profile Name

LinuxVM5GP

For-Each Value Of

device.logdirLinux | ▼

Matching Expression

- User Provides the Profile Name details as {foreach-instance}

Profile Name

- User clicks on save after the setting the alarm message to match.

Active

Profile Name

Pattern to Match

Severity

Alarm Message on Match

- User creates watcher rule (Enhanced) Profile to match the pattern for creating alarm on log file. Here we user searches for string 'error' and configure thresholds.

Alarm On Every Run

Enable

Message Severity

Metric Collection and Alarm Definition

Metrics		Alarms			
	Publish	Baseline	Type	Operator	Value
Count Matches	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="text" value="Static"/>	<input type="text" value=">"/>	<input type="text" value="1"/>
			<input type="text" value="Static"/>	<input type="text" value=">"/>	<input type="text" value="4"/>
			<input type="text" value="Static"/>	<input "="" type="text" value="="/>	<input type="text" value="0"/>

Value	Severity	Alarm Message
<input type="text" value="1"/>	<input type="text" value="Major"/>	<input type="text" value="Found error in {foreach-instance} > 1"/>
<input type="text" value="4"/>	<input type="text" value="Critical"/>	<input type="text" value="Found error in {foreach-instance} > 4"/>
<input type="text" value="0"/>	<input type="text" value="Info"/>	<input type="text" value="Found error in {foreach-instance} = 0"/>

- User clicks on save for creating the for-each group level profile.
- The profiles are created at device level for each of custom attributes configured for the device. In this case we will have 3 profiles created as below.

- ▶ LP/root/Test/log1.txt
- ▶ LP/root/Test/log2.txt
- ▶ LP/root/Test/log3.txt







- The instance variables are substituted at device level for profile name.







Profile Name







Mode

Log File or Command

- The alarms are raised once the string "error" is matched in log files.

 **snrh8vm5** Alarm - snw12vm4 - Sun Apr 26 2020 01:41:24 AM
 Alarm on every run /root/Test/log1.txt
 14,701 since 4 days ago      **MORE**

 **snrh8vm5** Alarm - snw12vm4 - Sun Apr 26 2020 01:35:23 AM
 Alarm on every run /root/Test/log3.txt
 39,001 since 4 days ago      **MORE**

 **snrh8vm5** Alarm - snw12vm4 - Sun Apr 26 2020 01:31:24 AM
 Alarm on every run /root/Test/log2.txt
 29,201 since 4 days ago      **MORE**

Known issues / Limitations / Out of Scope

- For-each do not create default alarm policies. As a pre-requisite disable the flag **policy_mode_enabled = false** in mon_config_service probe under **configure/timed** section. The thresholds are configured at template level.
- We are not supporting the for-each profile at child template level, but sub profiles can be created. The for-each attribute is enabled at group level for parent templates having max profile count as greater than 1.
- For-each flag is visible in UI for those templates, which do not have parent template and max profiles that can be created are more than 1.