EEM Policy Manager

Embedded Event Manager (EEM) Policy Manager provides automatic policy provisioning on a device. This module describes how to configure the automatic deployment of EEM policies.

Information About EEM Policy Manager

EEM Policy Manager Overview

The EEM Policy Manager, also called the EEM Auto-Deploy provides automatic policy provisioning on a device. An EEM policy is an entity that defines an event and actions to be taken when that event occurs. Devices can automatically download policy files from a central server and provision it based on the manifest file located on the central server; thereby, reducing the manual deployment of scripts.

Devices need to be configured for automatic policy download by specifying a set of parameters, such as the location of the manifest file, log files, scheduling, and so on. Based on the configuration, devices can add or remove policies from the central server. Status logs are sent to the server with the status of the policy provisioning (success/failure).

A central manifest file manages the policy provisioning for the entire network. Update the manifest file to install or remove policies on devices.

The EEM policy manager provides an automatic, network-wide policy deployment, thereby, eliminating manual provisioning. It provides the following advantages:

- A random timer to upgrade devices in staged manner during a maintenance window.
- Installation status logs are uploaded to the central server with classification based on status.
- Specify the time to start policy provisioning.
- Specify the protocol for downloading the manifest file and policy files.
- Policy files can be located in a different machine than the manifest file.
- Retain the automatic provisioning configuration, even after the feature is disabled.

- Enable debug logs on a device to view the progress of an EEM policy addition/removal.
- Reschedule the EEM policy download and provisioning on a device.
- Provision to chose manual policy provisioning.

Automated Download

A device can specify the interval at which it should start the policy provisioning. Time can be specified in days, hours, and minutes.

The device starts the policy provisioning by downloading a manifest file from the central server. The central manifest file manages the policy provisioning for the entire network. This file provides details of the policies to be deployed on hosts, based on the host name, and EEM version.

Based on the host name or the EEM version, the policy manager decides the next action to be taken. If there are no policies to be added or removed, no action is taken.

The following is a sample manifest file:

```
<adploy>
 <CommonSection>
  <RepositoryURL>tftp://10.106.16.20/folder1/EEM</RepositoryURL>
  <StatusLogURL>tftp://10.106.16.20/folder1/EEM</StatusLogURL>
 </CommonSection>

 <MappingGroups>
        <Group>
         <Name>GROUP_1</Name>
            <Operation>DELETE</Operation>
            <FileGroup>FILE_LIST_1</FileGroup>
            <DeviceGroup>DEVICE_LIST_1</DeviceGroup>
        </Group>
        <Group>
            <Name>Group_2</Name>
            <Operation>DELETE</Operation>
            <FileGroup>FILE_LIST_2</FileGroup>
            <DeviceGroup>DEVICE_LIST_2</DeviceGroup>
        </Group>
    </MappingGroups>

 <FileGroups>
  <FileGroup>
   <FilegroupName>FILE_LIST_2</FilegroupName>
   <Policies>
    <Policy>
     <Policy_id>123</Policy_id>
     <Type>tcl</Type>
     <EnvFilename>A.ENV</EnvFilename>
     <PolicyFilename>A.tcl</PolicyFilename>
     <PolicyDescription>Description about this Policy</PolicyDescription>
    </Policy>
    <Policy>
     <Policy_id>123</Policy_id>
     <Type>mlang</Type>
     <EnvFilename>A.ENV</EnvFilename>
     <PolicyFilename>A.py</PolicyFilename>
     <PolicyDescription>Description about this Policy</PolicyDescription>
    </Policy>
   </Policies>
  </FileGroup>
  <FileGroup>
   <FilegroupName>FILE_LIST_1</FilegroupName>
   <Policies>
    <Policy>
     <Policy_id>13</Policy_id>
     <Type>APPLET</Type>
     <EnvFilename>A.ENV</EnvFilename>
     <PolicyFilename>APP</PolicyFilename>
     <PolicyDescription>Description about this Policy</PolicyDescription>
     </Policy>
    <Policy>
     <Policy_id>15</Policy_id>
     <Type>TCL</Type>
     <EnvFilename>A.ENV</EnvFilename>
     <PolicyFilename>A.tcl</PolicyFilename>
     <PolicyDescription>Description about this Policy</PolicyDescription>
    </Policy>
   </Policies>
  </FileGroup>
 </FileGroups>

 <DeviceGroups>
     <DeviceGroup>
            <Name>DEVICE_LIST_1</Name>
            <InclusionList>4.0</InclusionList>
            <ExclusionList>3.0</ExclusionList>
            <DeviceListIs>EEM-Version</DeviceListIs>
            <DeviceMatchIs>EXACT</DeviceMatchIs>
        </DeviceGroup>
        <DeviceGroup>
            <Name>DEVICE_LIST_2</Name>
            <InclusionList>[A-Za-z]$</InclusionList>
            <ExclusionList>2147-29</ExclusionList>
            <DeviceListIs>HOST-Name</DeviceListIs>
            <DeviceMatchIs>REGEX</DeviceMatchIs>
        </DeviceGroup>
 </DeviceGroups>
</adploy>
```

Note XML tags are case sensitive.

Table 1: XML Tags Description

XML Tag	Description
<RepositoryURL>	Location where policies to be configured are stored.
<StatusURL>	Location where status logs are stored.
<Operation>Delete</Operation>	Deletes a policy. The value is not case sensitive.
<Operation> Add</Operation>	Adds a policy. The value is not case sensitive.
<Type>TCL|Applet|mlang</Type>	Type of policy. For example, whether it a TCL policy, a multi-language (Python) policy or an applet.

XML Tag	Description
<InclusionList>	Includes the device for provisioning, if it matches the condition specified in the InclusionList.
<ExclusionList>	Excludes this device for provisioning, if it matches the condition specified in the ExclusionList.
<DeviceListIs>Host-Name|EEM-Version</DeviceListIs>	Filters the device list based on the host name or the EEM version.
<DeviceMatchIs>REGEX | EXACT</DeviceMatchIs>	Device exactly matches the specified host name or matches a regular expression. This value is not case-sensitive. **Note** Match condition is EXACT match for EEM-version and EXACT/REGEX for Host-Name

The following are some of the values that you can specify in the XML tags:

- PolicyFilename: Name of policy file to be provisioned
- EnvFilename: Name of the environment variable file to be provisioned.
- EEM-version: The EEM version, the specified version in the tag should be a match. The value is not case-sensitive.
- FILE_LIST_1: Contains a list of policy files.

Note The FILE_LIST_1 can be any user-defined string value. Ensure that the same string is used consistently in other places where FILE_LIST_1 is referred.

- DEVICE_LIST_1: Selects a device based on this list, if the EEM Version installed on the device is 4.0 not 3.0.

Note The DEVICE_LIST_1 can be any user-defined string value. Ensure that the same string is used consistently in other places where DEVICE_LIST_1 is referred.

- GROUP_1: FILE_LIST_1 should be mapped to DEVICE_LIST_1. If the device matches the DEVICE_LIST_1, based on device group conditions, policy files in FILE_LIST_1 are provisioned on that particular device. If there is no match, provisioning is not done.

Note The GROUP_1 can be any user-defined string value. Ensure that the same string is used consistently in other places where GROUP_1 is referred.

The following is a sample applet file:

```
event manager applet interface_Shutdown
event syslog pattern "Interface FastEthernet1/0, changed state to administratively down"
action 1.0 cli command "show ip interface brief"
exit
```

Note Only one applet per file is allowed.

Configuring EEM Environment Variables

The policy manager does the provisioning of these environment variables. The following is a sample EEM environment variable file:

```
event manager environment A 123
event manager environment B 234
```

Configuring Policies

The policy manager downloads policies from the location specified in the manifest file and performs the provisioning of these policies. If any error occurs during the provisioning, the error is captured in a log file. After provisioning, log files are sent to the central server. The log file name will have the host name, date, and time.

Two types of logs files are sent to the central server, debug logs and summary logs. The progress of the provisioning is captured in debug logs and provisioning summary is captured in the summary logs. Both these files are transferred to the central server based on the log-url specified in the manifest file or profile configuration. The manifest file location takes the precedence over the profile configuration of the log-url. Logs collected from a device are transferred to the central server.

The following is a sample debug log:

```
***************************************************

Start downloading spec file for profile test
Spec file Url tftp://10.106.16.20/manashuk/123.xml:

Downloaded spec file at bootflash:test_spec_file

Parsing xml spec file at bootflash:test_spec_file

Start scheduling profile test

Start scheduling policies

Start scheduling policy APP

Downloaded Environment file variable file A.ENV

Provisioned ENV A.ENV policy
Successfully provisioned env vars

Downloaded APP policy
```

```
Successfully provisioned applet policy APP
Successfully provisioned Policy

Start scheduling policy A.tcl

Downloaded Environment file variable file A.ENV

Provisioned ENV A.ENV policy
Successfully provisioned env vars

Downloaded A.tcl policy

Successfully provision tcl/mlang policy A.tcl
Successfully provisioned Policy

Done scheduling profile test

***************************************************
```

The following is a sample summary log:

```
:: XML FILE ERRORS ::



:: Policy Failure/Success Results ::

Policy id: 13
----------------------------------
Envfilename: A.ENV
Result: Success(Successfully provisioned)
Policyname: APP
Result: Success(Successfully provisioned)

Policy id: 15
----------------------------------
Envfilename: A.ENV
Result: Success(Successfully provisioned)
Policyname: A.tcl
Result: Success(Successfully provisioned)
```

Note The XML FILE ERRORS section in the sample output will display any XML tag-related issues in the manifest file.

How to Configure EEM Policy Manager

Enabling EEM Policy Provisioning at a Specified Time

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **event manager auto-deploy name** *name*
4. **log-url** *url*
5. **manifest format xml url** *url*
6. **retry count** *retry-count*
7. **schedule start-in hours** *hours* **minutes** *minutes* {**oneshot** | **recurring** {**days** *days* | **hours** *hours*}
8. **window** *minutes*
9. **enable**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**event manager auto-deploy name** *name* **Example:** `Device(config)# event manager auto-deploy name deploy1`	Configures the EEM auto-deploy profile for automatic policy provisioning, and enters auto-deploy configuration mode.
Step 4	**log-url** *url* **Example:** `Device(config-auto-deploy)# log-url tftp://10.106.16.20/folder1/EEM`	Specifies the location where provisioning logs must be stored.
Step 5	**manifest format xml url** *url* **Example:** `Device(config-auto-deploy)# manifest format xml url tftp://10.106.16.20/folder1/123.xml`	Specifies the manifest file format, and the location from where manifest file should be downloaded.
Step 6	**retry count** *retry-count* **Example:** `Device(config-auto-deploy)# retry count 3`	Specifies the number of retries to transfer a file, if the file transfer is not successful.

Step 7	**schedule start-in hours** *hours* **minutes** *minutes* {**oneshot** | **recurring** {**days** *days* | **hours** *hours*}
	Example:
	`Device(config-auto-deploy)# schedule start-in hours 9 minutes 30 oneshot`
		Schedules the provisioning of policies at the specified time.

Step 8	**window** *minutes* **Example:**	(Optional) Sets a time duration for the profile provisioning to be triggered.

	Command or Action	Purpose
	`Device(config-auto-deploy)# window 20`	• This command only works if the **schedule start-in** command is configured. The **window** command adds a time interval to the time specified by the **schedule start-in** command. See Step 7. The **window** command adds 20 minutes to 9 hours and 30 minutes specified by the **schedule start-in** command. At a random time between 9 hours and 30 minutes to 9 hours and 50 minutes, the profile provisioning is triggered.
Step 9	**enable** **Example:** `Device(config-auto-deploy)# enable`	Enables an EEM auto-deploy profile. **Note** Unless the configured auto-deploy profile is enabled, it will not be active. The configured schedule will be effective only after the profile is enabled.

Step 10	**end**
	Example:
	`Device(config-auto-deploy)# end`
		Exits auto-deploy configuration mode and returns to privileged EXEC mode.

Enabling EEM Policy Provisioning Immediately

Once the EEM Policy Manager profile is configured and enabled, you can either enable policy immediately or at a specified scheduled time.

SUMMARY STEPS

1. **enable**
2. **event manager auto-deploy start name** *profile-name* {**now** | **window** *duration*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**event manager auto-deploy start name** *profile-name* {**now** | **window** *duration*}
	Example:
	`Device> event manager auto-deploy start name eem-profile1 now`
		Triggers the auto-depolyment of a profile, and starts the provisioning of policies.

Verifying EEM Policy Manager Provisioning

SUMMARY STEPS

1. **show event manager auto-deploy summary**
2. **debug event manager auto-deploy** {**common** | **parser** | **schedule**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show event manager auto-deploy summary** **Example:** `Device# show event manager auto-deploy summary`	Displays the summary of auto-deployed profiles.
Step 2	**debug event manager auto-deploy** {**common** | **parser** | **schedule**} **Example:** `Device# debug event manager auto-deploy schedule`	Enables the policy provisioning debugs.

Example

The following is sample output from the **show event manager auto-deploy summary** command:

```
Device# show event manager auto-deploy summary

EEM Auto-Deploy Profile details:

    Profile Name  : test
    Status        : Enabled
    Running       : Yes
    Status Syslog : No
    Schedule      : start in 0 hours 5 mins oneshot
    Window        : 5
    Manifest URL  : tftp://10.106.16.20/folder1/123.xml
    Log URL       : tftp://10.106.16.20/folder1/EEM
```

Configuration Examples for EEM Policy Manager

Example: Enabling EEM Policy Provisioning

The following example shows how to enable policy provisioning at a specified time

```
Device> enable
Device# configure terminal
Device(config)# event manager auto-deploy name deploy1
Device(config-auto-deploy)# log-url tftp://10.106.16.20/folder1/EEM
Device(config-auto-deploy)# manifest format xml url tftp://10.106.16.20/folder1/123.xml
Device(config-auto-deploy)# retry-count 3
Device(config-auto-deploy)# schedule start-in hours 9 minutes 30 oneshot
Device(config-auto-deploy)# window 20
Device(config-auto-deploy)# enable
Device(config-auto-deploy)# end
Device#
```

The following example shows how to enable policy provisioning immediately:

```
Device> enable
Device# event manager auto-deploy start name eem-profile1 now
Device #
```

Feature Information for EEM Policy Manager

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2: Feature Information for EEM Policy Manager

Feature Name	Releases	Feature Information
EEM Policy Manager	Cisco IOS XE Everest 16.6.1	Embedded Event Manager (EEM) Policy Manager provides automatic policy provisioning on a device. The following commands were introduced or modified: **debug event manager auto-depoly**, **default (EEM)**, **enable (EEM)**, **event manager auto-deploy**, **event manager auto-deploy start**, **log-url**, **manifest format**, **retry count**, **schedule start-in**, **show event manager auto-deploy summary**, **window**.