EEM scripts

EEM scripts

A EEM script is an automation tool that

- monitors real-time system activity and events,
- executes defined actions when specific conditions are met, and
- streamlines troubleshooting and operational workflows.

EEM scripts act based on significant system occurrences—such as log messages, interface states, or telemetry changes—not limited to errors. For example, you can automate actions like enforcing LACP dampening if a bundle interface flaps multiple times within seconds by temporarily disabling the interface.

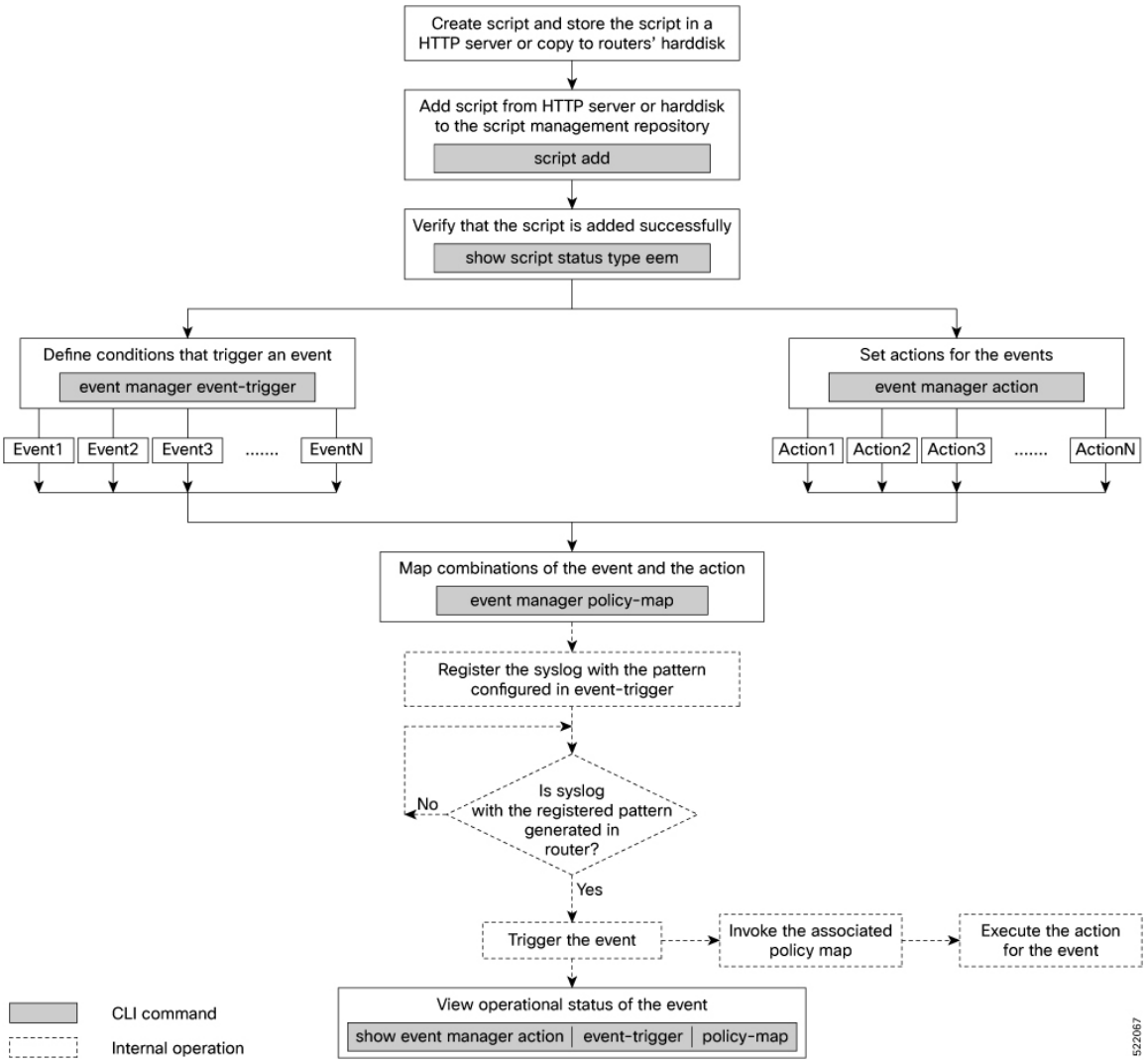
Administrators can define events and actions separately, then link them together using policy maps. A single event and action pair can be reused across multiple policies, and a policy map can contain up to five actions. EEM scripts may be written in Python 3.5 or TCL; script policies can be configured using either Command Line Interface (CLI) or NETCONF RPCs.

- Scripts are stored and managed in subdirectories based on type within the router's script management repository.
- You can map the same event and action in up to 64 policy maps for extensive automation reuse.
- For python scripting, only specific packages are supported; TCL scripts are also an option for event managers.

Manage eem scripts on the router

Complete these steps to provision eem scripts on the router.

Figure 1: Workflow to run events script



Procedure

Step 1 Download the script.

Step 2 Define events.

Step 3 Create actions to the events.

Step 4 Create policy map.

Note

An eem script is invoked automatically when the event occurs. With the event, the event-trigger invokes the corresponding policy-map to implement the actions in response to the event.

Step 5 View operational status of the event.

Download script to the router

To manage the scripts, you must add the scripts to the script management repository on the router. A subdirectory is created for each script type. By default, this repository stores the downloaded scripts in the appropriate subdirectory based on script type.

Table 1: Script download locations

Script type	Download location
config	harddisk:/mirror/script-mgmt/config
exec	harddisk:/mirror/script-mgmt/exec
process	harddisk:/mirror/script-mgmt/process
eem	harddisk:/mirror/script-mgmt/eem

Procedure

Step 1 Add the script to the script management repository on the router using one of the two options:

- Add script from a server.
- Copy the script from an external repository.

Example:

Add the script from a configured HTTP server or the harddisk location in the router.

```
Router#script add eem <script-location> <script.py>
```

The following example shows a config script `eem-script.py` downloaded from an external repository http://192.0.2.0/scripts:

```
Router#script add eem http://192.0.2.0/scripts eem-script.py
eem-script.py has been added to the script repository
```

You can add a maximum of 10 scripts simultaneously.

```
Router#script add eem <script-location> <script1.py> <script2.py> ... <script10.py>
```

You can also specify the checksum value while downloading the script. This value ensures that the file being copied is genuine. You can fetch the checksum of the script from the server from where you are downloading the script. However, specifying checksum while downloading the script is optional.

```
Router#script add eem http://192.0.2.0/scripts eem-script.py checksum SHA256 <checksum-value>
```

For multiple scripts, use the following syntax to specify the checksum:

```
Router#script add eem http://192.0.2.0/scripts <script1.py> <script1-checksum> <script2.py>
<script2-checksum>
... <script10.py> <script10-checksum>
```

If you specify the checksum for one script, you must specify the checksum for all the scripts that you download.

Note
Only SHA256 checksum is supported.

Example:

You can copy the script from the external repository to the routers' harddisk and then add the script to the script management repository.

a. Copy the script from a remote location to harddisk using scp or copy command.

```
Router#scp userx@192.0.2.0:/scripts/eem-script.py /harddisk:/
```

b. Add the script from the harddisk to the script management repository.

```
Router#script add eem /harddisk:/ eem-script.py
eem-script.py has been added to the script repository
```

Step 2 Verify that the scripts are downloaded to the script management repository on the router.

Example:

```
Router#show script status
=============================================================================================
 Name               | Type     | Status           | Last Action | Action Time
---------------------------------------------------------------------------------------------
 eem-script.py      | eem      | Config Checksum  | NEW         | Tue Aug 24 10:44:53 2021
=============================================================================================
```

Define trigger conditions for events

Configure system criteria so specific events are automatically triggered when defined conditions are met.

- The keywords `occurrence` (number of matches before event raises) and `period` (interval for above) can be used with syslog events only.
- To verify a telemetry sensor path or query before configuring an event trigger, use:

  ```
  Router# event manager telemetry sensor-path <sensor-path> json-query <query>
  ```

- Example for syslog trigger with severity:

  ```
  Router(config)# event manager event-trigger eventT10 type syslog pattern "L2-BM-6-ACTIVE"
   severity info
  ```

  The event triggers if both pattern and severity match a syslog message.

Before you begin

Ensure the relevant script is added to the script management repository.

Procedure

Step 1 Register the event.

Example:

```
Router(config)#event manager event-trigger eventT10
```

Step 2 Configure the trigger type and its options for the event.

Syslog event

```
Router(config)# event manager event-trigger eventT10 type syslog pattern "<pattern-to-match>" [severity
 <value>]
```

- Specify a pattern that matches the syslog message.
- Optionally, add a severity value (alert, critical, debug, emergency, error, info, notice, warning).
- The event triggers only when both pattern and severity match, or if severity is not set, any severity matches the pattern.

Timer event

Watchdog timer

```
Router(config)# event manager event-trigger <event-name> type timer watchdog value <seconds>
```

Cron timer

```
Router(config)# event manager event-trigger <event-name> type timer cron cron-entry "<cron string>
```

Track event

```
Router(config)# event manager event-trigger <event-name> type track name <track-name> status {up |
down | any}
```

Triggers when the specified track object's status changes.

Telemetry event:

Match criteria as exact match

```
Router(config)# event manager event-trigger <event-name> query json-path <query> match-criteria
exact-match value <value> type telemetry sensor-path <sensor-path> sample-interval <seconds>
```

Match criteria as threshold

```
Router(config)# event manager event-trigger <event-name> query json-path <query> match-criteria
threshold {equal-to | greater-equal-to | greater-than | less-equal-to | less-than | not-equal-to}
<value> type telemetry sensor-path <sensor-path> sample-interval <seconds>
```

Match criteria as rate:

```
Router(config)# event manager event-trigger <event-name> query json-path <query> match-criteria rate
 direction {any | decreasing | increasing} value {equal-to| greater-equal-to | greater-than |
less-equal-to | less-than | not-equal-to} <value> type telemetry sensor-path <sensor-path>
sample-interval <seconds>
```

Before creating a telemetry event, enable model-driven telemetry:

```
Router# telemetry model-driven
```

The router or system will now automatically trigger the event when the specified conditions are matched.

Create actions for events

Define the actions that must be taken when an event occurs.

Before you begin

- Define trigger conditions for an event.

Procedure

Step 1 Set the event action.

Example:

```
Router(config)#event manager action action1
```

Step 2 Define the type of action. For example, the action is a Python script.

Example:

```
Router(config)#event manager action action1 type script action1.py
```

Step 3 Configure the maximum run time of the script for the event.

Example:

```
Router(config)#event manager action action1 type script action1.py maxrun seconds 30
```

The default value is 20 seconds.

Step 4 Configure the checksum for the script. This configuration is mandatory. Every script is associated with a checksum hash value. This value ensures the integrity of the script, and that the script is not tampered. The checksum is a string of numbers and letters that act as a fingerprint for script.

a) Retrieve the SHA256 checksum hash value for the script from the IOS XR Linux bash shell.

Example:

```
Router#run
[node0_RP0_CPU0:~]$sha256sum /harddisk:/mirror/script-mgmt/eem/action1.py
407ce32678a5fc4b0ad49e83acad6453ad1d47e8dad9501cf139daa75d53e3dd
/harddisk:/mirror/script-mgmt/eem/action1.py
```

b) Configure the checksum for the script.

Example:

```
Router(config)#event manager action action1 type script action1.py checksum
sha256 407ce32678a5fc4b0ad49e83acad6453ad1d47e8dad9501cf139daa75d53e3dd
```

Step 5 Enter the username for the script to execute.

Example:

```
Router(config)#event manager action action1 username eem_user
```

If you load the event manager action commands using configuration files, for example, by using the **load harddisk:config.txt** command, you must make sure that the commands in the configuration files are properly indented and aligned with the running configuration.

In this example, the **username eem** and **type script** commands in the **config.txt** configuration file are properly indented and aligned with the running configuration.

```
event manager action action_all
 username eem
 type script script-name eem.py Marx seconds 7200 checksum
sha256fb2e1f7c4b135c296abb7149cf5fb96f052d3876c35a8422d44f78b9b6d3e452
```

Policy maps

Policy map is a configuration object that

- enables the association of multiple actions with one or more events,
- supports boolean logic (AND or OR) for correlating multiple events, and
- allows conditional triggering based on event occurrences within a specified period.

With a policy map, you can configure the system to execute an EEM (Embedded Event Manager) script only when certain combinations of events happen, such as requiring both a specific status threshold and a timer event before taking action. Optional parameters like occurrence and period control how frequently or under what time constraints the policy is triggered.

Table 2: Feature History Table

Feature Name	Release Information	Description
Add Multiple Events In a Policy Map With a Single EEM Script	Release 25.1.1	Introduced in this release on: 8700 [ASIC: K100](select variants only*) *This feature is now supported on Cisco 8712-MOD-M routers.
Add Multiple Events In a Policy Map With a Single EEM Script	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100(select variants only*)Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is now supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-12TH24FH-E • 88-LC1-36EH+A8:B12 • 88-LC1-52Y8H-EM

Feature Name	Release Information	Description
Add Multiple Events In a Policy Map With a Single EEM Script	Release 7.5.1	With this feature, you can add multiple events to a policy map with boolean (AND or OR) correlation. EEM triggers the script when the correlation defined in the policy map for the events is true. Using EEM scripts, you can create a logical correlation of events in the policy map and configure multiple actions for detectors such as timer, object-tracking, and telemetry events via sensor path.

Associate events and actions with a policy map

Configure a policy map to link network events with automated actions using Embedded Event Manager (EEM).

Policy maps allow you to define which events trigger specific actions on your router. You can use Boolean logic to correlate multiple events, specify how many occurrences trigger the action, and set a time period for event evaluation.

Before you begin

- Ensure you are in global configuration mode on the router.
- Identify the EEM events and actions you want to associate.

Procedure

Step 1 Start a policy map configuration.

Example:

```
Router(config)#event manager policy-map policy1
```

Step 2 Define event triggers.

For a single event

Example:

```
Router(config-policy-map)#trigger event event1
```

For multiple events with Boolean logic, enclose the logic in double quotes:

Example:

```
Router(config-policy-map)#trigger multi-event "<event1> AND (<event2> OR <event3>
```

Step 3 (Optional) Set occurrence count.

Specify how many times the total correlation must occur before the event is raised

Example:

```
Router(config-policy-map)# occurrence 2
```

- Acceptable values: 1 to 32.

- If not specified, the policy map triggers on every occurrence

Step 4 (Optional) Set evaluation period.

Define the time interval (in seconds) during which the events must occur.

```
Router(config-policy-map)# period 60
```

Acceptable values: 1 to 429496729.

Step 5 Specify actions.

Map actions to the policy map (maximum of 5).

```
Router(config-policy-map)# action action-2
```

The policy map is configured. When defined events occur according to your logic, the router automatically executes the associated action.

View operational status of eem components

Retrieve the operational status of events, actions and policy maps.

Before you begin

- Define trigger conditions for an event
- Create actions for events
- Create a policy map of events and actions

Procedure

Step 1 Run the **show event manager event-trigger all** command to view the summary of basic data of all events that are configured.

Example:

```
Router#show event manager event-trigger all

No. Name    esid   Type    Occurs  Period  Trigger-Count Policy-Count Status
1   event1  1008   syslog  2       1800    4             1            active
2   event2  1009   syslog  2       1800    4             1            active
3   event3  1010   syslog  2       1800    4             1            active
4   event4  1011   syslog  2       1800    4             1            active
5   event5  1012   syslog  2       1800    4             1            active
6   event6  1013   syslog  2       1800    4             1            active
7   event7  1014   syslog  2       1800    4             1            active
8   event8  1015   syslog  2       1800    4             1            active
9   event9  1016   syslog  2       1800    4             1            active
```

Use the **show event manager event-trigger all detailed** command to view the details about the match criteria that you configured, severity level, policies mapped to the events and so on.

Use the **show event manager event-trigger <event-name> detailed** command to view the details about the individual events.

```
Router#show event manager event-trigger event1 detailed

Event trigger name: event1
Event esid: 107
Event type: timer
Event occurrence: NA
Event period: NA
Event rate-limit: NA
Event triggered count: 12861
Event policy reg count: 1
Event status: active
Timer type: watchdog
Timer value: 10

Policy mapping info
1    event1                     policy1
```

Step 2 Run the **show event manager policy-map all** command to view the summary of all the configured policy maps.

Example:

```
Router#show event manager policy-map all
No. Name      Occurs   period    Trigger-Count     Status
1   policy1   NA       NA        1                 active
2   policy2   NA       NA        1                 active
3   policy3   NA       NA        1                 active
4   policy4   NA       NA        1                 active
```

Use the **show event manager policy-map all detailed** command to view the details about mapping of associated events and actions in the policy maps.

```
Router#show event manager policy-map policy1 all detailed

Policy name: policy1
Policy occurrence: 3
Policy period: 120
Policy triggered count: 0
Policy status: active
Multi event policy: FALSE

Events mapped to the policy
No.       Name                          Status
1         event2                        active

Actions mapped to the policy
No.       Name                          Checksum
1         action1                       SHA256
```

Use the **show event manager policy-map <policy-map-name> detailed** command to view the details about the individual policy maps.

```
Router#show event manager policy-map policy1 detailed

Policy name: policy1
Policy occurrence: 2
Policy period: 60
Policy triggered count: 0
Policy status: active
Multi event policy: TRUE
Multi event string : "event1 OR (event4 AND event2)"
Current Correlation State : FALSE
```

```
Events mapped to the policy
No.       Name                          Status         Corr Status    Reset time(sec)
1         event1                        active         0              0
2         event2                        active         0              0
3         event4                        active         0              0

Actions mapped to the policy
No.       Name                          Checksum
1         action2                       SHA256
```

Step 3 Run the **show event manager action <action-name> detailed** commad to view the details of an action.

Example:

```
Router#show event manager action action1 detailed
Tue Aug 24 16:05:44.298 UTC

Action name: action1
Action type: script
EEM Script name: event_script_1.py
Action triggered count: 1
Action policy count: 1
Username: eem_user
Checksum: 407ce32678a5fc4b0ad49e83acad6453ad1d47e8dad9501cf139daa75d53e3dd
Last execution status: Success

Policy mapping info
1    action1                       policy1
```

Use the **show event manager action all** and **show event manager action all detailed** command to view the summary and details about all the configured actions.