

# Cisco EEM Basic Overview and Sample Configurations

---

Introduction Sample EEM Scripts: Related Information

## Introduction

The EEM(Embedded Event manager is a software component of cisco IOS, XR, and NX-OS makes life easier for administrators by tracking and classifying events that take place on a router and providing notification options for those events. EEM allows you to automate tasks, perform minor enhancements and create workarounds.

There are two independent pieces: Applets and Scripting-> Applets are a collection of CLI commands-> Scripts are actions coded up in TCL(interpreter language)EEM uses event detectors and actions to provide notifications of those events:EEM detectors can be:1) SNMP:-Monitoring SNMP objects.2) Syslog:-Responds to various syslog messages, allowing for matching on regular expressions.3) Counter: Monitoring and responding to interface counter when cross threshold settings.4) CLI events: Screening CLI input for a regular expression match.5) None: This event detector is use to test EEM script/applet using "event manager run" command.6) Timers :(Countdown, watchdog and CRON)7) IP SLA and Netflows events.

Common regular expressions:

^ = Start of string

\$ = End of string

. = Any single character

\* = Zero or more instances

+ = One or more instance

? = Zero or one instance

EEM Actions can be: 1) Sending an email message 2) Executing a Cisco command 3) Generating SNMP traps 4) Reloading the router 5) Generating prioritized syslog messages 6) Switching to a secondary processor in a redundant platform 7) Requesting system information when an event occurs (like `sh tech`, `sh process cpu history`).

## Sample EEM Scripts:

### A) Syslog Event detector:

This example shows the syslog event detector. When the ISP1 interface has been shutdown, the below applet runs to turn on the interface and send the alert to the below mentioned mail ID with logged-in users' information.

### Configuration:

```
event manager applet interface_Shutdown
```

```
event syslog pattern "Interface FastEthernet1/0, changed state to administratively down"
```

```
action 1.0 cli command "enable"
```

```
action 1.5 cli command "config t"
```

```
action 2.0 cli command "interface fa1/0"
```

```
action 2.5 cli command "no shutdown"
```

```
action 3.0 cli command "end"
```

```
action 3.5 cli command "who"
```

```
action 4.0 mail server "192.168.1.1" to ".engineer@cisco.com." from ".EEM@cisco.com." subject  
".ISP1_Interface_fa1/0_SHUT." body "Current users $_cli_result"
```

Enabling debug detector to get closer view and shutdown the interface fa1/0.

```
R1#debug event manager action cli
```

Debug EEM action cli debugging is on

```
R1#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
R1(config-if)#int fa1/0
```

```
R1(config-if)#sh
```

```
*Nov 17 23:06:24.215: %LINK-5-CHANGED: Interface FastEthernet1/0, changed state to administratively down
```

```
*Nov 17 23:06:24.283: %HA_EM-6-LOG: interface_Shutdown : DEBUG(cli_lib) : : CTL : cli_open called.
```

```
*Nov 17 23:06:24.295: %HA_EM-6-LOG: interface_Shutdown : DEBUG(cli_lib) : : OUT : R1>
```

```
*Nov 17 23:06:24.299: %HA_EM-6-LOG: interface_Shutdown : DEBUG(cli_lib) : : IN :  
R1>enable
```

```
*Nov 17 23:06:24.319: %HA_EM-6-LOG: interface_Shutdown : DEBUG(cli_lib) : : OUT : R1#
```

```
*Nov 17 23:06:24.319: %HA_EM-6-LOG: interface_Shutdown : DEBUG(cli_lib) : : IN :  
R1#config t
```

```
*Nov 17 23:06:24.343: %HA_EM-6-LOG: interface_Shutdown : DEBUG(cli_lib) : : OUT : Enter configuration  
commands, one per line. End with CNTL/Z.
```

```
R1(config-if)#
```

```
*Nov 17 23:06:24.347: %HA_EM-6-LOG: interface_Shutdown : DEBUG(cli_lib) : : OUT : R1(config)#
```

```
*Nov 17 23:06:24.347: %HA_EM-6-LOG: interface_Shutdown : DEBUG(cli_lib) : : IN :  
R1(config)#interface fa1/0
```

```
*Nov 17 23:06:24.467: %HA_EM-6-LOG: interface_Shutdown : DEBUG(cli_lib) : : OUT : R1(config-if)#
```

```
*Nov 17 23:06:24.467: %HA_EM-6-LOG: interface_Shutdown : DEBUG(cli_lib) : : IN :  
R1(config-if)#no shutdown
```

```
*Nov 17 23:06:24.587: %HA_EM-6-LOG: interface_Shutdown : DEBUG(cli_lib) : : OUT : R1(config-if)#
```

\*Nov 17 23:06:24.587: %HA\_EM-6-LOG: interface\_Shutdown : DEBUG(cli\_lib) : : IN :  
R1(config-if)#end

\*Nov 17 23:06:24.643: %SYS-5-CONFIG\_I: Configured from console by on vty0 (EEM:interface\_Shutdown)

\*Nov 17 23:06:24.655: %HA\_EM-6-LOG: interface\_Shutdown : DEBUG(cli\_lib) : : OUT : R1#

\*Nov 17 23:06:24.655: %HA\_EM-6-LOG: interface\_Shutdown : DEBUG(cli\_lib) : : IN :  
R1#who

\*Nov 17 23:06:24.727: %HA\_EM-6-LOG: interface\_Shutdown : DEBUG(cli\_lib) : : OUT : Line User

R1(config-if)#Host(s) Idle Location

\*Nov 17 23:06:24.731: %HA\_EM-6-LOG: interface\_Shutdown : DEBUG(cli\_lib) : : OUT : 0 con 0  
idle 00:00:02

\*Nov 17 23:06:24.731: %HA\_EM-6-LOG: interface\_Shutdown : DEBUG(cli\_lib) : : OUT : \* 2 vty 0  
idle 00:00:00 EEM:interface\_Shutdown

\*Nov 17 23:06:24.735: %HA\_EM-6-LOG: interface\_Shutdown : DEBUG(cli\_lib) : : OUT :

\*Nov 17 23:06:24.735: %HA\_EM-6-LOG: interface\_Shutdown : DEBUG(cli\_lib) : : OUT : Interface  
User Mode Idle Peer Address

\*Nov 17 23:06:24.735: %HA\_EM-6-LOG: interface\_Shutdown : DEBUG(cli\_lib) : : OUT :

\*Nov 17 23:06:24.739: %HA\_EM-6-LOG: interface\_Shutdown : DEBUG(cli\_lib) : : OUT : R1#

R1(config-if)#

\*Nov 17 23:06:26.487: %LINK-3-UPDOWN: Interface FastEthernet1/0, changed state to up

## B) CLI Event detector:

In this example applet has created to stop "debugging" command on the CLI. If anyone enter the debug command it will have no effect, also clock and user's information will be stored in flash.

## Configuration:

```
event manager applet Stop_Debug
```

```
event cli pattern "debug" sync no skip yes
```

```
action 1.0 cli command "enable"
```

```
action 2.0 cli command "sh users | append flash:Debug"
```

```
action 3.0 cli command "sh clock | append flash:Debug_clock"
```

Verification:

Checking the registered applet with following command:

```
R1#sh event manager policy registered
```

No.	Class	Type	Event Type	Trap	Time Registered	Name
1	applet	user	cli	Off	Fri Mar 1 00:08:05 2002	Stop_Debug

```
pattern {debug} sync no skip yes
```

```
maxrun 20.000
```

```
action 1.0 cli command "enable"
```

```
action 2.0 cli command "sh users | append flash:Debug"
```

```
action 3.0 cli command "sh clock | append flash:Debug_clock"
```

```
R1#debug all
```

```
R1#debug event manager action cli
```

R1#sh debugging.....(Not showing any debug running on device)

R1#sh flash:

System CompactFlash directory:

File	Length	Name/status
------	--------	-------------

1	1614	Debug.....(User info is append to flash)
---	------	--

2	204	Debug_clock.....(Clock time when debug entered is also append to flash)
---	-----	---

R1#more flash:Debug

Line	User	Host(s)	Idle	Location
0 con 0		idle	00:00:00	
* 66 vty 0		idle	00:00:00	XYZ

Interface	User	Mode	Idle	Peer Address
-----------	------	------	------	--------------

R2#more flash:Debug\_Clock

12:43:15.263 UTC Sun Nov 18 2012

R2#

### C) Interface Events:

In this example applet is configure to measure rxload on the interface.

Two applets are configure:

1) When Rx load is above 50% then router will apply access-list on the interface fa0/1 to allow only important traffic and block the unwanted traffic.

2) When Rx load goes below 25, the router will remove applied access-list from the interface fa0/1 to allow all traffic.

### Configuration:

```
event manager applet Interface_Load_UP
```

```
event tag 1 interface name FastEthernet1/0 parameter rxload entry-op gt entry-val 127 entry-type value poll-  
interval 30
```

```
action 1.0 cli command "enable"
```

```
action 1.1 cli command "conf t"
```

```
action 1.2 cli command "int fa1/0"
```

```
action 1.3 cli command "ip access-group IMP_TRAFFIC in "
```

```
action 1.4 cli command "end"
```

```
event manager applet Interface_Load_Down
```

```
event tag 1 interface name FastEthernet1/0 parameter rxload entry-op lt entry-val 25 entry-type value poll-  
interval 30
```

```
action 1.0 cli command "enable"
```

```
action 1.1 cli command "conf t"
```

```
action 1.2 cli command "int fa1/0"
```

```
action 1.3 cli command "no ip access-group IMP_TRAFFIC in "
```

```
action 1.4 cli command "end"
```

### Verification:

```
R1#sh int fa1/0 | in load
```

```
reliability 255/255, txload 1/255, rxload 1/255..(Rxload is 1)
```

```
R1#debug event manager action cli
```

```
Debug EEM action cli debugging is on
```

```
R1#sh int fa1/0 | in load
```

```
reliability 255/255, txload 139/255, rxload 139/255....(Rxload is increase more 50% i.e.139)
```

```
R1#(As soon Rxload increase by 50%, causing to triggered 1 st event)
```

```
*Nov 18 02:19:55.403: %HA_EM-6-LOG: Interface_Load : DEBUG(cli_lib) : : CTL : cli_open called.
```

```
*Nov 18 02:19:55.423: %HA_EM-6-LOG: Interface_Load : DEBUG(cli_lib) : : OUT : R1>
```

```
*Nov 18 02:19:55.427: %HA_EM-6-LOG: Interface_Load : DEBUG(cli_lib) : : IN :  
R1>enable
```

```
*Nov 18 02:19:55.451: %HA_EM-6-LOG: Interface_Load : DEBUG(cli_lib) : : OUT : R1#
```

```
*Nov 18 02:19:55.451: %HA_EM-6-LOG: Interface_Load : DEBUG(cli_lib) : : IN : R1#conf t
```

```
*Nov 18 02:19:55.575: %HA_EM-6-LOG: Interface_Load : DEBUG(cli_lib) : : OUT : Enter configuration  
commands, one per line. End with CNTL/Z.
```

```
*Nov 18 02:19:55.579: %HA_EM-6-LOG: Interface_Load : DEBUG(cli_lib) : : OUT : R1(config)#
```

```
*Nov 18 02:19:55.579: %HA_EM-6-LOG: Interface_Load : DEBUG(cli_lib) : : IN :  
R1(config)#int fa1/0
```

```
*Nov 18 02:19:55.703: %HA_EM-6-LOG: Interface_Load : DEBUG(cli_lib) : : OUT : R1(config-if)#
```

```
*Nov 18 02:19:55.707: %HA_EM-6-LOG: Interface_Load : DEBUG(cli_lib) : : IN : R1(config-  
if)#ip access-group IMP_TRAFFIC in....(Access-list apply to interface)
```

```
*Nov
```

```
R1#18 02:19:55.827: %HA_EM-6-LOG: Interface_Load : DEBUG(cli_lib) : : OUT : R1(config-if)#
```

```
*Nov 18 02:19:55.831: %HA_EM-6-LOG: Interface_Load : DEBUG(cli_lib) : : IN : R1(config-  
if)#end
```



## Cisco EEM Basic Overview and Sample Configurations

\*Nov 18 02:19:55.839: %SYS-5-CONFIG\_I: Configured from console by on vty0 (EEM:Interface\_Load)

\*Nov 18 02:19:55.851: %HA\_EM-6-LOG: Interface\_Load : DEBUG(cli\_lib) : : OUT : R1#

\*Nov 18 02:19:55.855: %HA\_EM-6-LOG: Interface\_Load : DEBUG(cli\_lib) : : CTL : cli\_close called.

R1#sh ip access-lists

Extended IP access list IMP\_TRAFFIC

10 permit tcp any any eq www

20 permit tcp any any eq telnet

30 deny icmp any any (10 matches).....(Router Start dropping ICMP packets)

R1#sh run int fa1/0

Building configuration...

Current configuration : 159 bytes

!

interface FastEthernet1/0

bandwidth 64

ip address 10.1.1.1 255.255.255.0

ip access-group IMP\_TRAFFIC in

load-interval 30

duplex full

speed 100

!

end

R1#sh int fa1/0 | in load

reliability 255/255, txload 19/255, rxload 19/255(As soon as Rxload goes below 25, R1 trigger 2nd event)

R1#

\*Nov 18 02:21:31.367: %HA\_EM-6-LOG: Interface\_Load\_Down : DEBUG(cli\_lib) : : CTL : cli\_open called.

\*Nov 18 02:21:31.375: %HA\_EM-6-LOG: Interface\_Load\_Down : DEBUG(cli\_lib) : : OUT : R1>

\*Nov 18 02:21:31.379: %HA\_EM-6-LOG: Interface\_Load\_Down : DEBUG(cli\_lib) : : IN :  
R1>enable

\*Nov 18 02:21:31.395: %HA\_EM-6-LOG: Interface\_Load\_Down : DEBUG(cli\_lib) : : OUT : R1#

\*Nov 18 02:21:31.399: %HA\_EM-6-LOG: Interface\_Load\_Down : DEBUG(cli\_lib) : : IN :  
R1#conf t

\*Nov 18 02:21:31.423: %HA\_EM-6-LOG: Interface\_Load\_Down : DEBUG(cli\_lib) : : OUT : Enter configuration  
commands, one per line. End with CNTL/Z.

R1#

\*Nov 18 02:21:31.423: %HA\_EM-6-LOG: Interface\_Load\_Down : DEBUG(cli\_lib) : : OUT : R1(config)#

\*Nov 18 02:21:31.427: %HA\_EM-6-LOG: Interface\_Load\_Down : DEBUG(cli\_lib) : : IN :  
R1(config)#int fa1/0

\*Nov 18 02:21:31.455: %HA\_EM-6-LOG: Interface\_Load\_Down : DEBUG(cli\_lib) : : OUT : R1(config-if)#

\*Nov 18 02:21:31.459: %HA\_EM-6-LOG: Interface\_Load\_Down : DEBUG(cli\_lib) : : IN :  
R1(config-if)#no ip access-group IMP\_TRAFFIC in....(Removing Access-list)

\*Nov 18 02:21:31.579: %HA\_EM-6-LOG: Interface\_Load\_Down : DEBUG(cli\_lib) : : OUT : R1(config-if)#

\*Nov 18 02:21:31.579: %HA\_EM-6-LOG: Interface\_Load\_Down : DEBUG(cli\_lib) : : IN :  
R1(config-if)#end

\*Nov 18 02:21:31.587: %SYS-5-CONFIG\_I: Configured from console by on vty0 (EEM:Interface\_Load\_Down)

\*Nov 18 02:21:31.599: %HA\_EM-6-LOG: Interface\_Load\_Down : DEBUG(cli\_lib) : : OUT : R1#

\*Nov 18 02:21:31.603: %HA\_EM-6-LOG: Interface\_Load\_Down : DEBUG(cli\_lib) : : CTL : cli\_close called.

#### D) SNMP event Detector:

In this example EEM applet reads the SNMP variable and it will be triggered when the CPU goes above 60 %.you can customized applet according to the need. However, caution should be exercised to consider the file system free space before deploying the applet.

For SNMP OID you can use SNMP Object Navigator

#### Configuration:

```
event manager applet HIGH-CPU
```

```
!
```

```
event snmp oid 1.3.6.1.4.1.9.9.109.1.1.1.1.3 get-type next entry-op gt entry-val 60 poll-interval 5
```

```
!
```

```
action 0.1 cli command "enable"
```

```
action 0.2 syslog msg "Capturing high cpu information to flash:"
```

```
action 0.3 cli command "term length 0"
```

```
action 1.1 cli command "show process cpu sorted | append flash:EEM_CPU"
```

```
action 1.4 cli command "show ip traffic | append flash:EEM_TRAFFIC"
```

```
action 1.4 cli command "show logging | append flash:EEM_LOGS"
```

```
action 4.2 syslog msg "Removing EEM APPLET from running_config"
```

```
action 9.2 cli command "configure terminal"
```

action 9.3 cli command "no event manager applet HIGH-CPU"

action 9.4 cli command "end"

action 9.5 cli command "term default length"

Verification:

R2#

\*Nov 18 12:30:53.139: %HA\_EM-6-LOG: HIGH-CPU: Capturing high cpu information to flash:

\*Nov 18 12:30:53.371: %HA\_EM-6-LOG: HIGH-CPU: Removing EEM APPLET from running\_config

R2#sh flash:

System CompactFlash directory:

File	Length	Name/status
------	--------	-------------

1	22477	EEM_CPU
---	-------	---------

2	2665	EEM_LOGS
---	------	----------

E) TCL (Tool Control Language) is a scripting language used extensively by Cisco to facilitate the testing and automating of various functions in the IOS.

In this example, small TCL script configured to check reachability of Branch devices from the Core router.

R1#tclsh

```
R1(tcl)#
```

```
R1(tcl)#foreach VAR {
```

```
+>(tcl)#192.168.1.1
```

```
+>(tcl)#192.168.2.1
```

```
+>(tcl)#192.168.3.1
```

```
+>(tcl)#192.168.4.1
```

```
+>(tcl)#} { puts [exec "ping $VAR"] }
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/58/120 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/40/104 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/72/128 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:

U.U.U

Success rate is 0 percent (0/5)

R1(tcl)#

## Related Information

[Writing Embedded Event Manager Policies Using the Cisco IOS CLI](#)

[Cisco EEM Best Practices](#)

[Embedded Event Manager Overview](#)

[emm through event manager scheduler script](#)

[Writing Embedded Event Manager Policies Using Tcl](#)

[EEM Script For Periodically Clearing Interface Counters Using Regular Expression](#)

Base Initial configuration: