TOMORROW starts here.
Better Network Management Through Network Programmability

BRKCDN-1005

Joe Clarke, Distinguished Services Engineer
Jason Davis, Distinguished Services Engineer
Abstract

- As we enter the age of network programmability the data models, protocols, and tools provided by a programmable network can greatly improve and simplify network management tasks.

- Configuration and operational data can be read and set regardless of the underlying device. Errors are properly reported to ensure reliable delivery of data. Connections are secure and robust. Data is more intelligently extracted. This session will explore how tools like NETCONF, YANG, and Cisco's Embedded Event Manager, onePK APIs, and embedded Python scripting can radically improve network management applications by offering visibility and provisioning power throughout the network stack.

- We will also provide an overview of the Cisco Process Orchestrator solution which provides an excellent way to interface disparate systems and achieve robust automation in your environment. Operational state checks and assisted troubleshooting are enabled through automation and orchestration.
Agenda

- Legacy Protocols and Methods
- NETCONF
- OnePK
- Python Scripting
- Embedded Event Manager (EEM)
- Cisco Process Orchestrator (CPO)
- Example Use Cases
- Cisco Live Orlando Case Study
And that’s the way it was…
Manual CLI

Slow, Error-prone and Does Not Scale

I need to get the CDP neighbors...
foreach device [array names devs] {
    set username [lindex $devs($device) 0]
    set password [lindex $devs($device) 1]
    ... 
    send "show cdp nei\r"
}
CLI Drawbacks

- Command Line Interface is not meant to be an API
  - CLI is for people!
  - No well-defined structure to output

- Using CLI forces use of a VTY (i.e., a terminal)
  - Slow character-by-character or line-by-line interface
  - Not ideal for large data transfers

- Changes to the CLI can break scripts
  (remember “show mac-address-table” ➔ “show mac address-table”?)

- Difficult to determine if an error actually occurred in the command sequence
CLI Drawbacks

- CLI output is different across different platforms:

  - `show ip route (IOS)`
  - `show ip route (NX-OS)`

---

BRKCDN-1005 © 2013 Cisco and/or its affiliates. All rights reserved. Cisco Public
SNMP

```
snmpTable -v2c -c notpublic 10.1.1.4
```
SNMP Structure

- SNMP uses the Structure of Management Interface (SMI) as its data modeling language.

- The SMI model is laid out as the Management Information Base (MIB) in a tree structure.

- Objects are accessed in lexicographical order, which may not be conducive for all data (e.g., large ARP tables).
SNMP Drawbacks

- While structured, SNMP MIBs can be difficult to understand
- SNMP uses UDP as a transport and thus is may not be reliable
- No clear separation of operational and configuration data
- Not ideal for holistic configuration management
- Not scalable from a programmatic standpoint
  - No transaction support
  - No rollback
- Different devices support different MIBs (sometimes for the same function)
Intro to Programmable Protocols and Solutions

That’s the way I like it…

That’s the way I heard it should be…
NETCONF
What is NETCONF?

- NETCONF is an IETF configuration management protocol.

- It addresses many of the short-comings in SNMP’s Structured Management Interface (SMI) specifically to aid in the management of configuration data:
  - Lack of backup and restore capabilities for configurations
  - No concept of transactions
  - Strange or legacy concepts (e.g., limited length of string labels)
  - Lack of reliable error reporting

- The “wherefore” of NETCONF can be found in RFC 3535, “Overview of the 2002 IAB Network Management Workshop”

- NETCONF was meant to ease configuration management burden across multiple vendors.
## Comparing NETCONF and SNMP

<table>
<thead>
<tr>
<th></th>
<th>SNMP</th>
<th>NETCONF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Management Operations</strong></td>
<td>SNMP</td>
<td>NETCONF</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>SNMPv1/2c – Community Strings / None</td>
<td>SSH subsystem</td>
</tr>
<tr>
<td></td>
<td>SNMPv3 – Authentication MD5/SHA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SNMPv3 – Encryption DES56, 3DES168, AES128, 192, 256</td>
<td></td>
</tr>
<tr>
<td><strong>RPC Encapsulation</strong></td>
<td>Basic Encoding Rules (BER)</td>
<td>XML</td>
</tr>
<tr>
<td><strong>Transport Protocol</strong></td>
<td>UDP</td>
<td>TCP (reliable transport)</td>
</tr>
</tbody>
</table>
# Comparing NETCONF and SNMP

<table>
<thead>
<tr>
<th></th>
<th>SNMP</th>
<th>NETCONF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Models</td>
<td>Defined in MIBs</td>
<td>Defined in YANG modules (or XML schema documents)</td>
</tr>
<tr>
<td>Data Modeling Language</td>
<td>Structure of Management Information (SMI)</td>
<td>YANG (and XML schema)</td>
</tr>
<tr>
<td>Management Operations</td>
<td>SNMP</td>
<td>NETCONF</td>
</tr>
<tr>
<td>Security</td>
<td>SNMPv1/2c – Community Strings / None</td>
<td>SSH subsystem</td>
</tr>
<tr>
<td></td>
<td>SNMPv3 – Authentication MD5/SHA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SNMPv3 – Encryption DES56, 3DES168, AES128, 192, 256</td>
<td></td>
</tr>
<tr>
<td>RPC Encapsulation</td>
<td>Basic Encoding Rules (BER)</td>
<td>XML</td>
</tr>
<tr>
<td>Transport Protocol</td>
<td>UDP</td>
<td>TCP (reliable transport)</td>
</tr>
</tbody>
</table>
Comparing NETCONF and SNMP

- SNMP is a simple “object = value” representation
  ```
  nocuser@nms$ snmpget -v2c 10.0.0.1 sysDescr.0
  RFC1213-MIB::sysDescr.0 = STRING: "Cisco IOS Software, 2800 Software (C2800NM-ADVIPSERVICESK9-M), Version 12.4(15)T4, RELEASE SOFTWARE (fc2)
  Technical Support: http://www.cisco.com/techsupport
  Copyright (c) 1986-2008 by Cisco Systems, Inc.
  Compiled Thu 13-Mar-08 03:04 by prod_rel_team"
  ```

- A Network Management System (NMS) must still be able to parse the results and correlate object and value
  - In this example, might the multiline response cause a problem? How about special characters?
The NETCONF Protocol

- NETCONF is defined in RFC 6241
- Provides multiple operations for interacting with configuration and operational data
  - get-config
  - get
  - edit-config
  - commit
  - validate
  - copy-config
  - discard-changes
  - delete-config
  - lock
  - unlock
  - create-subscription
- Includes multiple configuration data stores (candidate, running, startup)
- Offers clear indication of when errors occur in operations
The Protocol Layers

**Content**
- Configuration Data
- Operational Data

**Operations**
- `<get-config>`
- `<edit-config>`
- `<get>`
- `<…>`

**Messages**
- `<rpc>`
- `<notification>`
- `<…>`

**Transport**
- SSH
- BEEP
- SOAP
- TLS
<?xml version="1.0" encoding="ISO-8859-1"?>
<rpc message-id="11">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <config-format-xml/>
    </filter>
  </get-config>
</rpc>

]]>]]>
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="11" xmlns="urn:ietf:params:netconf:base:1.0">
  <data>
    <xml-config-data>
      <Device-Configuration xmlns="urn:cisco:xml-pi">
        <version>
          <Param>15.2</Param>
        </version>
        <service>
          <timestamps>
            <debug>
              <datetime>
                <msec/>
              </datetime>
            </debug>
            <timestamps>
            </timestamps>
          </service>
        </Device-Configuration>
      </xml-config-data>
    </data>
  </rpc-reply>
Example NETCONF Interaction

Get Config

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<rpc message-id="11">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <config-format xml/>
    </filter>
  </get-config>
</rpc>
```
Example NETCONF Interaction

Error Reporting

Client

<?xml version="1.0" encoding="UTF-8"?>
  <error-config>
    <error-config-target>
      <target name="running"/>
    </error-config-target>
    <error-config-error-type>
      <error-type>protocol</error-type>
      <error-tag>operation-failed</error-tag>
      <error-severity>error</error-severity>
    </error-config-error-type>
  </error-config>
</rpc-message>

Server

<?xml version="1.0" encoding="UTF-8"?>
  <error-config>
    <error-config-target>
      <target name="running"/>
    </error-config-target>
    <error-config-error-type>
      <error-type>protocol</error-type>
      <error-tag>operation-failed</error-tag>
      <error-severity>error</error-severity>
    </error-config-error-type>
  </error-config>
</rpc-reply>
OnePK
One Platform Kit (onePK)

Overview

- OnePK is a device-level API for Cisco’s core operating systems
- Enables one to create applications that run on or near Cisco devices
- Provides capabilities to...
  - Innovate
  - Extend
  - Automate
  - Customize
  - Enhance
  - Modify
- …the default behavior of Cisco devices
OnePK Architecture

Applications

Cisco onePK API’s

IOS | IOS-XE | NX-OS | IOS-XR

Hardware Silicon
<table>
<thead>
<tr>
<th>Service Set</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DATA PATH</strong></td>
<td>Packet delivery services to application – e.g. copy, punt, inject</td>
</tr>
<tr>
<td><strong>POLICY</strong></td>
<td>Filtering (NBAR), classification (class-map, policy-map), actions (marking, policing, queuing), applying policy to interfaces</td>
</tr>
<tr>
<td><strong>ROUTING</strong></td>
<td>Read RIB Routes, add/remove routes, receive RIB notifications</td>
</tr>
<tr>
<td><strong>ELEMENT</strong></td>
<td>CPU/Memory statistics, interface statistics, element and interface events</td>
</tr>
<tr>
<td><strong>DISCOVERY</strong></td>
<td>Layer 3 topology and local service discovery</td>
</tr>
<tr>
<td><strong>UTILITY</strong></td>
<td>Syslog event and path tracing capability</td>
</tr>
<tr>
<td><strong>DEVELOPER</strong></td>
<td>Debug capability and CLI extension (invoke CLI from application)</td>
</tr>
</tbody>
</table>
Supports applications written in C, Java, Python, or any language that supports REST.

Combine C API with SWIG (http://www.swig.org) to build bindings in other languages.
Application Deployment Models

- Runs locally on the device
- Runs on a service blade
- Runs on a remote server
Getting System Properties (in C)

```c
char *str = NULL;

onep_element_connect(elemA, user, pwd, NULL, &sh);
onep_element_get_property(elemA, &property);

if (property) {
    onep_element_to_string(elemA, &str);
    if (str) {
        fprintf(stderr, "\nElement Info: %s\n", str);
        free(str);
    }
}
```
Custom Routing
Routing For Dollars

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, D - OSPF, IA - OSPF inter-area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
I - IS-IS, S - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
L1/2 - IS-IS inter-area, * - candidate default, U - per-user static route
O - OSPF, O1 - periodic downloaded static route, H - HPNP, L - LISP
A - application route
+ - repalced route, % - next hop override

Gateway of last resort is not set

10.6.8.8/8 is variably subnetted, 6 subnets, 2 masks
C 10.1.1.1/24 is directly connected, Ethernet0/0
L 10.1.1.1/24 is directly connected, Ethernet0/0
D 10.40.8.8/24 [90/2801856] via 40.10.1.2, 2wd, Serial2/0
D 10.40.8.8/24 [90/3195385] via 40.10.1.2, 2wd, Serial2/0
D 10.40.8.8/24 [90/3706856] via 40.10.1.2, 2wd, Serial2/0
D 10.40.8.8/24 [90/4729856] via 40.10.1.2, 2wd, Serial2/0
D 10.70.8.8/24 [90/3195385] via 40.10.1.2, 2wd, Serial2/0

20.6.8.8/24 is subnetted, 10 subnets
D 20.10.1.0 [90/3705856] via 40.10.1.2, 2wd, Serial2/0
D 20.20.1.0 [90/4729856] via 40.10.1.2, 2wd, Serial2/0
D 20.20.1.0 [90/4729856] via 40.10.1.2, 2wd, Serial2/0
D 20.30.1.0 [90/3705856] via 40.10.1.2, 2wd, Serial2/0
D 20.40.1.0 [90/4729856] via 40.10.1.2, 2wd, Serial2/0
D 20.50.1.0 [90/4729856] via 40.10.1.2, 2wd, Serial2/0
D 20.60.1.0 [90/4729856] via 40.10.1.2, 2wd, Serial2/0
D 20.70.1.0 [90/4729856] via 40.10.1.2, 2wd, Serial2/0
D 20.80.1.0 [90/4729856] via 40.10.1.2, 2wd, Serial2/0
D 20.90.1.0 [90/4729856] via 40.10.1.2, 2wd, Serial2/0
D 20.100.1.0 [90/4729856] via 40.10.1.2, 2wd, Serial2/0

30.6.8.8/24 is subnetted, 5 subnets
D 30.10.1.0 [90/3241856] via 40.10.1.2, 2wd, Serial2/0
D 30.20.1.0 [90/3241856] via 40.10.1.2, 2wd, Serial2/0
D 30.30.1.0 [90/3241856] via 40.10.1.2, 2wd, Serial2/0
D 30.40.1.0 [90/3241856] via 40.10.1.2, 2wd, Serial2/0
D 30.50.1.0 [90/3241856] via 40.10.1.2, 2wd, Serial2/0

40.6.8.8/8 is variably subnetted, 4 subnets, 2 masks
C 40.10.1.0/24 is directly connected, Serial2/0
L 40.10.1.3/24 is directly connected, Serial2/0
C 40.20.1.0/24 is directly connected, Serial2/3
L 40.20.1.3/24 is directly connected, Serial2/3

108.8.8.8/24 is subnetted, 1 subnets
D 108.10.1.0 [90/3241856] via 90.0.0.16, 2wd, Serial2/0

bangalore# show ip route
Getting and Setting Routes (in Java)

```java
L3UnicastScope scope = new L3UnicastScope("", AFIType.IPv4, SAFIType.UNICAST, "");
NetworkPrefix prefix = new NetworkPrefix(InetAddress.getByName("0.0.0.0"), 0);
L3UnicastRIBFilter ribFilter = new L3UnicastRIBFilter(OwnerType.NONE, "NONE", prefix);
L3UnicastRouteRange range = new L3UnicastRouteRange(prefix, RouteRange.RangeType.EQUAL_OR_LARGER, 100);
List<TopoNode> mynodes = TopoNode.getAllNodes();
for (TopoNode thisnode : mynodes) {
    Routing routing = Routing.getInstance(thisnode.ne);
    RIB rib = routing.getRib();
    List<Route> routelist = rib.getRouteList(scope, ribFilter, range);
    for (Route route : routelist) {
        // Get Routes

L3UnicastRoute aRoute = new L3UnicastRoute(prefix, nextHopList);
aRoute.setAdminDistance(1);
RouteOperation op = new L3UnicastRouteOperation(RouteOperationType.ADD, aRoute);
List<RouteOperation> opList = new ArrayList<RouteOperation>();
opList.add(op);
AppRouteTable art = routing getAppRouteTable();
art.updateRoutes(scope, opList);
    } // for
} // for
```

Get Routes

Set Routes
Python Scripting
Python Scripting Engines

- In addition to the external onePK Python API, Nexus switches include embedded Python interpreter.
- The Nexus 3K and Nexus 9K switches include a Cisco-specific set of Python modules that provide API access to certain subsystems.
- Use the `python` command to enter the interpreter in interactive mode.
Getting Just-In-Time Help

- Python interpreters have built-in help that can be accessed using the `help()` command once in the Python shell.
- “Help” mode provides details about built-in Python modules as well as the Cisco NX-OS API:
  - modules
  - topics
  - cisco
Python Scripting
Create a VRF and Add an Interface

```python
import sys
from cisco import VRF

if len(sys.argv) < 3:
    print "usage: vrf_demo.py VRF_NAME INTERFACE"
    exit(1)

vname = sys.argv[1]
intf = sys.argv[2]

v = VRF(vname)
if not v.create():
    print "Failed to create VRF %s" % (vname)
    exit(1)

if not v.add_interface(intf):
    print "Failed to add interface %s to VRF %s" % (intf, vname)
    exit(1)

print "Successfully created VRF %s and added interface %s to it." % (vname, intf)
```

Create a script to simplify the creation of a new VRF and then adding an interface to it.

NX-OS on the Cat3K exposes an API for VRF management.
Python Scripting
Running the Script

```
n3k-one# copy scp://marcus@rms-server2.cisco.com/home/marcus/vrf_demo.py bootflash:
Warning: There is already a file existing with this name. Do you want to overwrite (y/n)[n] y
Enter vrf (If no input, current vrf 'default' is considered): management
Password:
vrf_demo.py
100% 442 0.4KB/s 00:00
Copy complete, now saving to disk (please wait)...n
n3k-one# python bootflash:vrf_demo.py
usage: vrf_demo.py VRF_NAME INTERFACE
n3k-one# python bootflash:vrf_demo.py cl_milan Ethernet1/4
Successfully created VRF cl_milan and added interface Ethernet1/4 to it.
n3k-one# show run int e1/4
!Command: show running-config interface Ethernet1/4
!Time: Sun Jan 5 18:13:45 2014
version 6.0(2)U1(1a)
interface Ethernet1/4
no switchport
vrf member cl_milan
```
Python Scripting
Getting More Help and Samples

- N3K Documentation - [http://tools.cisco.com/squish/4F41b](http://tools.cisco.com/squish/4F41b)
- GitHub repository for N9K scripts:
  - [https://github.com/datacenter/nexus9000/tree/master/nx-os/python](https://github.com/datacenter/nexus9000/tree/master/nx-os/python)
  - [https://github.com/datacenter/nexus9000/tree/master/nx-os](https://github.com/datacenter/nexus9000/tree/master/nx-os)
- Monitor framework for N3K - [https://github.com/datacenter/PyMonitor](https://github.com/datacenter/PyMonitor)
Embedded Event Manager (EEM)
What Is Embedded Event Manager (EEM)?

- An extremely flexible and powerful subsystem within Cisco Software
- Provides real-time network event detection and onboard automation
- Generate and react to events when certain conditions are met in devices
- Trigger the execution of custom modules written in CLI or Tcl
- Adapt device behavior and insert business logic without an OS upgrade
EEM Architecture

- Extremely flexible and powerful subsystem within Cisco IOS, IOS-XE, IOS-XR, and NX-OS Software
- Adapt device behavior and insert custom logic without OS upgrade
- 24 Event Detectors (ED) integrated with IOS modules for wide range of system event detection
- CLI and Tcl based policy provides consistent programmability interface
- Powerful event engine supporting multi-event correlation, advance scheduling and more
### Event Detectors

- **Cisco IOS CLI ED**
  Triggers policies based on commands entered via the CLI.

- **Cisco IOS Counter ED**
  Policies can be triggered based on a change of the designated Cisco IOS counter.

- **Cisco IOS Redundancy Facility ED**
  Provides for detection of hardware and software failures related to the Stateful Switchover service. This ED will trigger policies based on the RF state change. It is also used to initiate switchovers as a result of a policy action.

- **Cisco IOS Timer Services ED**
  Policies can be scheduled to occur at the designated time or interval.

- **Cisco IOS Watchdog / System Monitor ED**
  Triggers policies based on certain conditions relative to a certain Cisco IOS process or subsystem’s activity.

- **EEM Application Specific ED**
  Application specific events can be detected or set by a Cisco IOS subsystem or a policy script. This provides the ability for one policy to trigger another policy.

- **Interface Counter ED**
  Policies can be triggered based on the specific interface counter; includes thresholds.

- **Online Insertion and Removal (OIR) ED**
  Triggers policies based on hardware installation and removal activity.

- **Object Tracking ED**
  Triggers policies based on routing protocol events.

- **SNMP ED**
  Triggers policies based on the associated SNMP MIB variable; includes MIB variable threshold setting.

- **SNMP Notification ED**
  Intercept incoming and outgoing SNMP traps and publish events

- **Syslog ED**
  Triggers policies based on the regular expression match of a local Syslog message.
Event Detectors

- **Resource Thresholding ED**
  Triggers policies based on certain internal resource usage and conditions; interface to Embedded Resource Manager.

- **Generic Online Diagnostics ED**
  Triggers policies based on diagnostic results

- **XML-RPC**
  Listen to SOAP request over SSH and trigger EEM policy execution.

- **“None”ED**
  Triggers policies by command
EEM Applet Policy

- **Step 1 – Configure Environment Var**
  - Configure necessary EEM environment variable

- **Step 2 -- Register Policy**
  - register the policy with the applet policy engine

- **Step 3 -- Establish Event Trigger**
  - Define the event that triggers the policy

- **Step 4 -- Define Action**
  - Define the action to take upon detection of the event

---

**event manager environment**

```
_email_server
172.27.121.177
```

**event manager applet**

```
email_hsrp_state_change
```

**event syslog**

```
pattern ".*%HSRP-5-STATECHANGE.*"
```

**action 1.0 info**

```
type routename
```

**action 1.1 cli command**

```
"enable"
```

**action 1.3 cli command**

```
"show standby | append hsrp_state_change"
```

**action 1.4 cli command**

```
"show standby brief | append hsrp_state_change"
```

---

**action 1.8 mail**

```
server "$_email_server" to "$_email_to" from "$_email_from" subject "HSRP_STATE_CHANGE Alert from 
$_info_routename: $_syslog_msg" body "$_cli_result"
```
### EEM 4.0 Event Detectors

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
<th>Version</th>
<th>Node</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>application</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>2</td>
<td>rf</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>3</td>
<td>identity</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>4</td>
<td>neighbor-discovery</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>5</td>
<td>msp</td>
<td>03.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>6</td>
<td>routing</td>
<td>02.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>7</td>
<td>syslog</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>8</td>
<td>generic</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>9</td>
<td>nhrp</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>10</td>
<td>track</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>11</td>
<td>resource</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>12</td>
<td>cli</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>13</td>
<td>counter</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>14</td>
<td>interface</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>15</td>
<td>ioswsysmon</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>16</td>
<td>none</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>17</td>
<td>oir</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>18</td>
<td>snmp</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>19</td>
<td>snmp-object</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>20</td>
<td>ipsla</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>21</td>
<td>snmp-notification</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>22</td>
<td>timer</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>23</td>
<td>test</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>24</td>
<td>config</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>25</td>
<td>env</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>26</td>
<td>ds</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>27</td>
<td>crash</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>28</td>
<td>nf</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
<tr>
<td>29</td>
<td>rpc</td>
<td>01.00</td>
<td>node0/0</td>
<td>RP</td>
</tr>
</tbody>
</table>

### EEM 4.0 Applet Actions

<table>
<thead>
<tr>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
</tr>
<tr>
<td>append</td>
</tr>
<tr>
<td>break</td>
</tr>
<tr>
<td>cli</td>
</tr>
<tr>
<td>cnsets-event</td>
</tr>
<tr>
<td>comment</td>
</tr>
<tr>
<td>context</td>
</tr>
<tr>
<td>continue</td>
</tr>
<tr>
<td>counter</td>
</tr>
<tr>
<td>decrement</td>
</tr>
<tr>
<td>divide</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>elseif</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>exit</td>
</tr>
<tr>
<td>file</td>
</tr>
<tr>
<td>file operations</td>
</tr>
<tr>
<td>force-switchover</td>
</tr>
<tr>
<td>foreach</td>
</tr>
<tr>
<td>get</td>
</tr>
<tr>
<td>handle-error</td>
</tr>
<tr>
<td>help</td>
</tr>
<tr>
<td>if</td>
</tr>
<tr>
<td>increment</td>
</tr>
<tr>
<td>info</td>
</tr>
<tr>
<td>mail</td>
</tr>
<tr>
<td>multiply</td>
</tr>
<tr>
<td>policy</td>
</tr>
<tr>
<td>publishing-event</td>
</tr>
<tr>
<td>puts</td>
</tr>
<tr>
<td>reload</td>
</tr>
<tr>
<td>set</td>
</tr>
<tr>
<td>snmp-object-value</td>
</tr>
<tr>
<td>snmp-trap</td>
</tr>
<tr>
<td>string</td>
</tr>
<tr>
<td>subtract</td>
</tr>
<tr>
<td>syslog</td>
</tr>
<tr>
<td>track</td>
</tr>
<tr>
<td>wait</td>
</tr>
<tr>
<td>while</td>
</tr>
</tbody>
</table>

Extensible Event Manager (EEM) is a feature of Cisco IOS that provides a way to react to events in the network. EEM can be used to monitor and respond to events, such as network failures or changes, and can be used to implement event-driven policies. EEM allows network administrators to define policies that are triggered by specific events, and to execute actions in response to those events. This can be used to automate network management tasks, such as resetting links or sending alerts to network administrators. EEM is typically used in conjunction with Cisco ASA firewalls and Cisco Firepower Threat Defense devices.
EEM Tcl Policy

- **Step 1 – Register User Directories**
  - Register user policy directory and user library directory

- **Step 2 – Develop Policies Offline**
  - No online editor available.

- **Step 3 – Download Policy**
  - Download TCL policies using standard IOS file transfer mechanisms
  - Support script auto refresh from remote location

- **Step 4 – EEM Environment Variable Configuration**

- **Step 5 – Register Policy**
  - Register policy to TCL policy engine

```
mkdir disk2:/eem
event manager directory user policy disk2:/eem
event manager directory user library disk2:/eemlib

copy tftp disk2:/eem
Address or name of remote host []? 10.1.88.9
Source filename []? cl_acl_change.tcl
Destination filename [cl_acl_change.tcl]? eem/cl_acl_change.tcl
Accessing tftp://10.1.88.9/cl_acl_change.tcl...
1232 bytes copied in 0.620 secs (1987 bytes/sec)

event manager update user policy group "*.tcl" repository
tftp://2.2.2.2/users2/eem_1

event manager environment _email_server 172.27.121.177

event manager policy cl_acl_change.tcl type user
```
Say you want to suspend (i.e., shutdown) ports that haven’t been active in a week…

…Use EEM to watch for ports that become active…
EEM Tcl Policy
Suspending Inactive Ports (Cont.)

```tcl
set fd $result
set contents [read $fd]
close $fd

set contents [string trim $contents]
array set ports [split $contents]

if { [info exists ports($iface)] } {
    array unset ports $iface
    set fd [open $suspend_ports_config "w"]
    puts -nonewline $fd [array get ports]
close $fd
}
```

...Then delete those newly active ports from a log file tracking ports that are down.
EEM Tcl Policy
Track Ports That Are Down

```
::cisco::eem::event_register_timer cron cron_entry "0 0 * * *" queue_priority normal maxrun 600
if { ![info exists suspend_ports_days] } {
    set result "ERROR: Policy cannot be run: variable suspend_ports_days has not been set"
    error $result $errorInfo
}
if { ![info exists suspend_ports_config] } {
    set result "ERROR: Policy cannot be run: variable suspend_ports_config has not been set"
    error $result $errorInfo
}
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
proc run_cli { clist } { ... }
set SECS_IN_DAYS 86400
set DOWN 0
set UP 1
set ADMIN_DOWN 2
set now [clock seconds]
set susp_time [expr $suspend_ports_days * $SECS_IN_DAYS]
array set suspend_ports [list]
if { [catch {open $suspend_ports_config "r"} result] } {
    array set ports [list]
} else {
    set fd $result
    set contents [read $fd]
    close $fd
    set contents [string trim $contents]
    array set ports [split $contents]
}
```

Every night at midnight, another EEM policy runs that records each “down” port into a file...
set result [run_cli [list "show ip interface brief | include Ethernet"]]
foreach line [split $result "\n"]{
    set line [string trim $line]
    regsub -all {s+} "$line" " " line
    set elems [split $line]
    set iface [lindex $elems 0]
    if { ![regexp (Ethernet) $iface] || [llength $elems] < 6 } {
        continue
    }
    if { [lindex $elems 4] == "administratively" && [lindex $elems 5] == "down" } {
        set status $ADMIN_DOWN
    } elseif { [lindex $elems 4] == "down" } {
        set status $DOWN
    } elseif { [lindex $elems 4] == "up" && [lindex $elems 5] == "up" } {
        set status $UP
    } else {
        set status $DOWN
    }
    if { [info exists ports($iface)] } {
        if { $status == $UP || $status == $ADMIN_DOWN } {
            array unset ports $iface
        } elseif { [expr $now - $ports($iface)] >= $susp_time } {
            set suspend_ports($iface) $ports($iface)
        }
    } else {
        if { $status == $DOWN } {
            set ports($iface) $now
        }
    }
}...
EEM Tcl Policy
Track Ports That Are Down (Cont.)

```tcl
set fd [open $suspend_ports_config "w"]
puts -nonewline $fd [array get ports]
close $fd

set cli [list "config t"]
foreach port [array name suspend_ports] {
    if {[info exists suspend_quarantine_vlan]} {
        set cli [concat $cli [list "interface $port" "switchport access vlan $suspend_quarantine_vlan"]]
        action_syslog msg "Moving port $port into quarantine VLAN $suspend_quarantine_vlan since it was last used on [clock format $suspend_ports($port)]"
    } else {
        set cli [concat $cli [list "interface $port" "shut"]]
        action_syslog msg "Shutting down port $port since it was last used on [clock format $suspend_ports($port)]"
    }
}
lappend cli "end"
if { [catch {run_cli $cli} result] } {
    action_syslog priority err msg "Failed to shutdown ports: '$result'"
}
```

…For each port in the list, it is administratively shut down or moved to a quarantine VLAN (if said VLAN is defined).

Download the full version from https://supportforums.cisco.com/docs/DOC-39192
EEM For Real-World Challenges

- **Challenge 1:** Every few weeks a router is running low on memory around 2am, and I want to find out what’s happening

  **Solution:** EEM script could be triggered based on the memory utilization, capture the memory information and send the output with Syslog or Email

- **Challenge 2:** One of the (redundant) links has problems with occasional high error rates, but does not go down. It causes service interruption for my customer.

  **Solution:** EEM script could be triggered on the interface errors, remove the link to force the use of the stable link and send a notification by Syslog

- **Challenge 3:** My ACL configuration gets changed, I want to get notified, but I can’t sit there and monitor it all the time

  **Solution:** EEM script could be triggered by CLI command, take a snapshot of the logged in user, changed configuration, and send an email to you
Challenge 4: I want to save energy, but I can’t go around turn off everyone’s IP phone everyday

Solution: Timer ED can be used to trigger the execution of an EEM script to turn off your IP phone at 7pm everyday and turn it back on 7am the next day.

Challenge 5: My NMS system need this MIB that is missing from my router, and I can’t wait for the next IOS release

Solution: Using EEM SNMP Proxy ED, an EEM script can be configured to intercept and process SNMP requests, collect the information from other management interface, e.g., show commands, and simulate the MIB.
# EEM Version/Product Support Matrix

## CISCO ACCESS ROUTERS - Current Models

<table>
<thead>
<tr>
<th>EEM Version</th>
<th>Cisco 800 Series</th>
<th>Cisco 1800 Series</th>
<th>Cisco 2800 Series</th>
<th>Cisco 3800 Series</th>
<th>Cisco 1900 Series</th>
<th>Cisco 2900 Series</th>
<th>Cisco 3900 Series</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td></td>
<td>12.3(11)T</td>
<td>12.3(11)T</td>
<td>12.3(11)T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td></td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.4</td>
<td>12.4(20)T</td>
<td>12.4(20)T</td>
<td>12.4(20)T</td>
<td>12.4(20)T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>12.4(22)T</td>
<td>12.4(22)T</td>
<td>12.4(22)T</td>
<td>12.4(22)T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
</tr>
<tr>
<td>3.2</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
</tr>
<tr>
<td>4.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15.2(2)T</td>
<td>15.2(2)T</td>
<td>15.2(2)T</td>
</tr>
</tbody>
</table>

## CISCO ACCESS ROUTERS - Old Models

<table>
<thead>
<tr>
<th>EEM Version</th>
<th>Cisco 1700 Series</th>
<th>Cisco 2600 Series</th>
<th>Cisco 2600XM Series</th>
<th>Cisco 2691 Series</th>
<th>Cisco 3600 Series</th>
<th>Cisco 3700 Series</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>12.3(4)T</td>
<td>12.3(4)T</td>
<td>12.3(4)T</td>
<td>12.3(4)T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
</tr>
<tr>
<td>2.1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
</tr>
<tr>
<td>2.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Notes
- **Shipping**
- **Coming Soon**
- **Planning**
## EEM Version/Product Support Matrix

### CISCO SERVICE AGGREGATION/CORE ROUTERS

<table>
<thead>
<tr>
<th>EEM Version</th>
<th>Cisco ASR1000 Series</th>
<th>Cisco 7200 Series</th>
<th>Cisco 7301</th>
<th>Cisco 7304</th>
<th>Cisco 7600 Series</th>
<th>Cisco UBR 10000</th>
<th>Cisco UBR 7200</th>
<th>Cisco 12000 Series</th>
<th>Cisco XR 12000</th>
<th>Cisco CRS-1</th>
<th>Cisco ASR 9000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12.0(26)S</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FM</td>
<td></td>
<td>FM</td>
</tr>
<tr>
<td>2.1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td>12.2(28)SB</td>
<td>12.2(18)SXF5</td>
<td>12.2(28)SB</td>
<td>12.2(28)SB</td>
<td></td>
<td></td>
<td>FM</td>
<td></td>
<td>FM</td>
</tr>
<tr>
<td>2.2</td>
<td>12.4(2)T</td>
<td>12.4(2)T1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FM</td>
<td></td>
<td>FM</td>
</tr>
<tr>
<td>2.3</td>
<td>2.1XE</td>
<td>12.4(11)T</td>
<td>12.2(33)SB</td>
<td>12.2(33)SB</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SB</td>
<td>12.2(33)SB</td>
<td>FM</td>
<td></td>
<td>FM</td>
</tr>
<tr>
<td>2.4</td>
<td>12.2(33)XN RL57</td>
<td>12.4(20)T</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>FM</td>
<td></td>
<td>FM</td>
</tr>
<tr>
<td>3.0</td>
<td>12.2(33)XN RL57</td>
<td>12.4(22)T</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>FM</td>
<td></td>
<td>FM</td>
</tr>
<tr>
<td>3.1</td>
<td>15.1(3)S3</td>
<td>15.0(1)M</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td></td>
<td>Planning</td>
</tr>
<tr>
<td>3.2</td>
<td>Planning</td>
<td>15.1(3)T</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td></td>
<td>Planning</td>
</tr>
<tr>
<td>4.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Planning</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### CISCO CATALYST SWITCHES

<table>
<thead>
<tr>
<th>EEM Version</th>
<th>Catalyst 3000 Switches</th>
<th>Cisco 3400ME Switches</th>
<th>Catalyst 4500 Switches</th>
<th>Catalyst 4900 Switches</th>
<th>Catalyst 6500 Switches</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>105 w/o Modularity</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12.2(18)SXF5</td>
</tr>
<tr>
<td>2.1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>w/ Modularity</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12.2(18)SXF4</td>
</tr>
<tr>
<td>2.3</td>
<td>12.2(40)SE</td>
<td>12.2(40)SE</td>
<td>12.2(44)SG</td>
<td>12.2(44)SG</td>
<td>12.2(33)SXI</td>
</tr>
<tr>
<td>2.4</td>
<td>12.2(50)SE</td>
<td>12.2(50)SE</td>
<td>12.2(52)SG</td>
<td>12.2(52)SG</td>
<td>12.2(33)SXI</td>
</tr>
<tr>
<td>3.0</td>
<td>12.2(52)SE</td>
<td>12.2(52)SE</td>
<td>12.2(54)SG1</td>
<td>12.2(54)SG1</td>
<td>12.2(33)SXI</td>
</tr>
<tr>
<td>3.1</td>
<td>12.2(52)SE</td>
<td>12.2(52)SE</td>
<td>15.0(2)SG</td>
<td>15.0(2)SG</td>
<td>15.1(1)SY</td>
</tr>
<tr>
<td>4.0</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
</tr>
</tbody>
</table>

*Notes:*
- Total rows: 12
- Total columns: 12
Cisco Process Orchestrator (CPO)
Cisco Intelligent Automation

- **Cisco Process Orchestrator (CPO)**
- From Tidal acquisition in May 2009
- Provides Orchestration/Run book automation
- Provides ‘glue’ and data transformation among tools/apps and devices
- Automation of ongoing operations:
  - Monitoring, incident, fault mgmt
  - Metering and billing
  - Capacity & compliance
- OS & app provisioning
- Integration with ITIL tools & processes
Cisco Intelligent Automation

- **Cisco Prime Service Catalog**
  - From newScale acquisition in April 2011
  - Provides “pretty” User Interface/Portal and data-gathering front-end to supply user input to back-end automation tools (eg. CPO)
- **Service Request Management (Service Catalog)**
- **One-stop shop self-service portal**
- **IT service catalog w/ tiers and pricing**
- **Governance, policies, approvals, & controls**
- **Lifecycle management & tracking**
- **Resource management & reservations**
Some Solutions Have Many Moving Parts

Consider the applications, tools and devices necessary to administer BYOD

MS AD/LDAP Server
LMS
Certificate Authority
Cisco Security Manager
Cisco Prime Infrastructure

ISE
DNS/DHCP Server
WLAN Controller
IronPort Web Security Appliance
Mobility Services Engine

HOW many user interfaces is that?
Troubleshooting is Complex

“And this is where our customer and tech support staff went insane.”
A Common User Interface Can Integrate with Automation

Delete User
Provides functionality needed to delete a user from the BYOD solution

Pre-Provision User
Provides functionality needed to pre-provision a BYOD user – Webex, Jabber, etc.

Add/Change Policy
Provides functionality needed to add or delete a policy from the BYOD solution

Troubleshooting Wizard
Provides functionality to collect data for troubleshooting and Cisco TAC Service Requests
Solution

- Use Cisco Intelligent Automation to meld the workflow into simpler tasks
Assisted Troubleshooting - Automating Manual Tasks
From Manual Best Practices to Intelligent Automation

BEFORE
Manual Best Practices

Problem definition
This message indicates an error in the port ASIC. The first [dec] is the module
number. The second [dec] is the ASIC port number. This error message could
possibly be due to either software or a hardware issue.

Problem definition
(for SYS-3-
PKTBUFFERFAIL-ERDIDS)
This error message is caused because of a hardware problem (parity error) on one of
the ASICs, where the disabled ports are directly connected. The erdsabling of the
ports is a precautionary action of the switch after it receives a parity error message

Impact
Network connectivity to the ports of the module will be lost until this error is resolved

Troubleshooting steps
To recover the module or ports from coil ASIC errors, perform the following steps:
1. Disable and enable the ports; if the error is not resolved by
re-enabling the ports then more testing is required on the whole module. Depending
on module# and which ports are effected, use configuration mode and do the
following:
- Conf t
- interface range (module#)/(intf start) – (intf end)
- Shut
- No shut
2. Set the bootup online diagnostics level to complete using
the configuration mode command:
- Conf t
- diagnostic bootup level complete
3. Soft-reset the module by issuing the following command:
- hw-module module (module#) reset
4. Perform a Hard-reset of the module by physically reseating
the card. Alternatively, issue the following global configuration commands:
- conf t
- no power enable module (module#)
- power enable module (module#)
NOTE: Remember to set the diagnostics level back to minimal by issuing the
configuration mode command:
- conf t
- diagnostic bootup level minimal
5. If the above steps fail to recover the ports, consider
replacing the faulty module.
For more information, see Troubleshooting Hardware and Common Issues on
Catalyst 6500/6000 Series Switches Running Cisco IOS System Software.

AFTER
Automated Process executed
by Advanced Automation Tool
How Do We Do That?

**Most Common CPO Activities**
- Terminal (Open/Execute/Close) (SSH)
- Table Activities (Insert/Select/Update)
- String Activities
- SNMP (Get/Set/Trap)
- Email
- Match Regular Expression
- Set Variable
- Web HTTP Request
- Web Service Execute

**Useful Methods**
- Conditional Branch
- While Loop
- For-each Loop
Example Tasks and Methods
Consistent Interface Settings With onePK

Problem: Misconfigurations cause network outages, degrade performance, impact SLAs.
Value proposition: Get, set, and detect configuration changes via cross-platform API

1. Network begins with mismatched parameters on either side of link (e.g. MTU)
2. Application checks parameters on either side and identifies mismatches (red lines)
3. Application sets parameters to match (lines turn green)
4. Application registers for events related to parameters change.
5. Users logs into console and manually changes parameter. Topology indicates change.
MTU In Action

Device Info
Hostname: ASR9k
Address: 172.20.165.43
Processor: MPC8641D
Version: ASR-9006 AC Chassis
Serial Number: FOX1548GNC1
Neighbors: 2
MTU:
  Gi0/0/0/1:1500 to Eth1/1:1500(proctype)
  Gi0/0/0/0:1024 to Gi0/0/0/0:1514(CRS)
public void checkInterfacesMTU(NetworkElement ne) {
    for (NetworkInterface intf : ne.getInterfaceList(new InterfaceFilter(null,
                                NetworkInterface.Type.ONEP_IF_TYPE_Ethernet))) {
        InterfaceConfig ic = intf.getConfig();
        if (ic.getMTU() != STANDARD_MTU) {
            intf.setMTU(STANDARD_MTU);
            ne.createSyslogMessage(NetworkElement.OnepSyslogSeverity.ONEP_SYSLOG_INFO, "Changed MTU of " + intf.getName() + " from " + ic.getMTU() + " to " + STANDARD_MTU);
        }
    }
}

public void handleEvent(CLIEvent e, Object clientData) {
    checkInterfacesMTU(e.getNetworkElement());
}

CLIFilter cfilt = new CLIFilter("^mtu ");
cfilt.setSync(0);
cfilt.setSkip(0);
ne.addCLILeveler(new RouteTest(), cfilt, null);
checkInterfacesMTU(ne);

Iterate through each interface and verify MTU.
Callback to re-run the interface check whenever MTU is changed from the CLI.
Register a CLI handler for whenever anyone types a command starting with "mtu ".

MTU Code (in Java)
Customized Provisioning
Distributing Intelligence To the Device

- EEM XML-RPC can be used to split the intelligence between the device and the NMS
- The NMS can use a known API when talking to each device type
- The “driver” or device-specific code can be maintained within an EEM policy on each device
- EEM XML-RPC Perl API available at https://supportforums.cisco.com/docs/DOC-19523

On the NMS (Perl)

```perl
my $pol = new Cisco::EEM::RPC::Policy("add_customer.tcl");
$pol->setArgList('Acme Co.', '192.168.17.0/29', 'GOLD');
...
my $result = $rpc->invoke($pol);
```

On the Device (Tcl)

```tcl
... if { $arr_einfo(arg2) == {GOLD} } {
    cli_exec $cli(fd) "service-policy GOLD"
    cli_exec $cli(fd) " class acme_co"
    cli_exec $cli(fd) " bandwidth 1024"
}
...```
Getting Interface Parameters
With NETCONF

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="4" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source><running/></source>
    <filter type="cli"><config-format-xml options="interface GigabitEthernet0/0"></config-format-xml></filter>
  </get-config>
</rpc>

<?xml version="1.0" encoding="UTF-8"?><rpc-reply message-id="4" xmlns="urn:ietf:params:netconf:base:1.0"><data><xml-config-data>
Building configuration...
  <Device-Configuration>
    <interface><Param>GigabitEthernet0/0</Param>
    <ConfigIf-Configuration>
      <ip><address><dhcp/></address></ip>
      <duplex><auto/></duplex>
    </ConfigIf-Configuration>
  </interface>
</xml-config-data></data></rpc-reply>
```
Setting Interface Parameters
With NETCONF

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="edit-config addhosts" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target><running/></target>
    <config>
      <xml-config-data>
        <Device-Configuration>
          <ip>
            <host>
              <NameHost>shangrila</NameHost>
              <HostIPAddress>1.2.3.4</HostIPAddress>
            </host>
          </ip>
          <ip>
            <host>
              <NameHost>vallhala</NameHost>
              <HostIPAddress>1.2.3.5</HostIPAddress>
            </host>
          </ip>
        </Device-Configuration>
      </xml-config-data>
    </config>
  </edit-config>
</rpc>
```
Getting / Setting Interface Parameters
With EEM

- This one is built-in to many EEM releases! `tm_cli_cmd.tcl`
- Set a few variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_email_server</td>
<td>Hostname or IP address of SMTP server</td>
</tr>
<tr>
<td>_email_to</td>
<td>Recipient email address</td>
</tr>
<tr>
<td>_email_from</td>
<td>Sender email address</td>
</tr>
<tr>
<td>_email_cc</td>
<td>Carbon-copy recipient email address</td>
</tr>
<tr>
<td>_cron_entry</td>
<td>Defines the CRON specification when the policy will run</td>
</tr>
<tr>
<td>_show_cmd</td>
<td>Defined the command to be executed with the policy runs</td>
</tr>
</tbody>
</table>
Getting / Setting Interface Parameters
With EEM

- Config example – email trunk interface stats every 2 hours

```plaintext
hostname cpu
ip domain-name example.com
event manager environment _email_server smtp.example.com
event manager environment _email_to noc@example.com
event manager environment _email_from routera@example.com
event manager environment _cron_entry 0 0-23/2 * * 0-7
event manager environment _show_cmd show interface gig1/24
event manager policy tm_cli_cmd.tcl
```
In addition to email and syslog, EEM includes an HTTP API

- Data can be shipped off-box to a web service application for further processing
  - Database insertion
  - Analytics
  - Etc.

```
namespace import ::http::*

set output [cli_exec $cli(fd) "show mac address-table | include Total"]
if { ! [regexp {: (\d+)} $output -> tmacs] } {
    action_syslog priority err msg "Failed to get total MAC count: \'$output’"
    exit 1
}
::http::config -useragent "MAC Updater/1.0 (Cisco EEM)"
set query [::http::formatQuery count $tmacs device [info hostname]]
set tok [::http::geturl -query $query "http://www.company.com/rest/mac_addrs/total"]
if { [::http::error $tok] != "" } {
    puts "ERROR: Failed to upload total MAC count; \'[::http::error $tok]'"
    exit 1
}
action_syslog msg "Total MAC count uploaded successfully."
```
Interface Statistics Using onePK

- OnePK provides APIs to gather throughput, load, errors, and drops from interfaces
  - Most stats seen under “show interface” are available via the API
- API is event-based so changes can notify the application instead of relying on continuous polling

Live event-driven updates
Interface Statistics Using onePK

The Code (in Java)

Interface Selector Code

Register for stats change events

Callback on Stats Change

```java
public void handleEvent(InterfaceStatisticsEvent e, Object clientData) {
    String prevVal;
    switch (e.getParameter()) {
        case ONEPK_IF_STAT_RECEIVE_RATE_BPS:
            prevVal = IBPS.getText();
            IBPS.setText(e.getValue().toString());
            highlight(IBPS, prevVal, e.getValue());
            break;
        case ONEPK_IF_STAT_TRANSMIT_RATE_BPS:
            prevVal = DBPS.getText();
            DBPS.setText(e.getValue().toString());
            highlight(DBPS, prevVal, e.getValue());
            break;
        case ONEPK_IF_STAT_RECEIVE_RATE_PPS:
            prevVal = IPPS.getText();
            IPPS.setText(e.getValue().toString());
            highlight(IPPS, prevVal, e.getValue());
            break;
        case ONEPK_IF_STAT_TRANSMIT_RATE_PPS:
            prevVal = DPPS.getText();
            DPPS.setText(e.getValue().toString());
            highlight(DPPS, prevVal, e.getValue());
            break;
        case ONEPK_IF_STAT_INPUT_ERRORS:
            prevVal = Ierrs.getText();
            Ierrs.setText(e.getValue().toString());
            highlight(Ierrs, prevVal, e.getValue());
            break;
        case ONEPK_IF_STAT_OUTPUT_ERRORS:
            prevVal = Oerrs.getText();
            Oerrs.setText(e.getValue().toString());
            highlight(Oerrs, prevVal, e.getValue());
            break;
        case ONEPK_IF_STAT_INPUT_PACKETS DROPPED:
            prevVal = Drops.getText();
            Drops.setText(e.getValue().toString());
            highlight(Drops, prevVal, e.getValue());
            break;
        case ONEPK_IF_STAT_OUTPUT_PACKETS DROPPED:
            prevVal = Odrops.getText();
            Odrops.setText(e.getValue().toString());
            highlight(Odrops, prevVal, e.getValue());
            break;
        default:
            // Don’t care
    }
}
```
Getting / Setting Interface Parameters

With CPO
Getting / Setting Interface Parameters
With CPO
Getting / Setting Interface Parameters

With CPO
Getting / Setting Interface Parameters

With CPO
Getting / Setting Interface Parameters
With CPO
Assume we’d like to ensure another device or path is still operational using a router’s IPSLA feature – we can trigger EEM

track 100 rtr 50 reachability

ip sla 50
   icmp-echo 192.168.1.10
ip sla schedule 50 life forever start-time now

event manager applet watch-ipsla-down
   event track 100 state down
   action 1.0 syslog priority errors msg "IP SLA operation 50 to priority service is down"

event manager applet watch-ipsla-up
   event track 100 state up
   action 1.0 syslog msg "IP SLA operation 50 to priority service is up"
Service Validation
With onePK

- In addition to syslog messages and traps, EEM can generate application-specific events
- These events can be picked up by onePK apps
- Apps can form a bridge between systems like change management or trouble-ticketing

```plaintext
event manager applet watch-ipsla-up
  event track 100 state up
  action 1.0 syslog msg "IP SLA operation 50 to priority service is up"
  action 2.0 publish-event sub-system 798 type 3451 arg1 "Down"

Uplink down, opening case
```
Service Validation

The Code (in Java)

```java
/**
 * Listen for application events coming from our EEM policy.
 */
ApplFilter filter = new ApplFilter(798, 3541);
ne.addApplListener(this, filter, null);
...

public void handleEvent(ApplEvent event, Object clientData) {
    String status = "Up";
    try {
        status = event.getData(1);
    } catch (Exception e) {
        getLogger().error("Failed to get data element 1", e);
        return;
    }

    /**
     * Change the line color based on the results of our IP SLA collector.
     * Open case if need be.
     */
    if (status.equals("Down")) {
        this.lineColor = Color.red;
        this.case = Case.createCase();
    } else {
        this.lineColor = Color.green;
        this.case.updateCase(Case.Status.CLOSED);
    }

    this.repaint();
}
```
Device Health Monitoring
With EEM

- Assume we’d like to get an emailed list of processes running on an IOS-based device when the CPU goes above 75%

```plaintext
event manager applet snmp_cpu_ge75
event snmp oid 1.3.6.1.4.1.9.9.109.1.1.1.1.3.1 get-type exact entry-op ge entry-val 75 poll-interval 10
action 1.0 cli command "enable"
action 2.0 cli command "show process cpu"
action 3.0 mail server "192.168.1.10" to "noc@example.com" from "router-a@example.com"
  subject "CPU High Alert" body "$_cli_result"
```
Use onePK to build a live process monitor similar to UNIX `top`

The same app can connect to multiple devices to display the top processes across the entire network

```
while (!done) {
    List<ElementProcess> plist = ne.getProcessList();
    for (ElementProcess proc : plist) {
        displayProcess(proc);
    }
    try {
        Thread.sleep(delay);
    } catch (InterruptedException e) {
    }
}
```
Device Health Monitoring
With CPO

- “Operational State Checks”
  The automated collection of various ‘show’ command and other operational state data to ensure correct system operation.
  E.g. ‘show ntp status’ does **not** show ‘stratum 16’ or ‘insane time’

- “Assisted Troubleshooting”
  The automated collection of specific diagnostic data that is useful for troubleshooting. Ideally complemented by a first-order analysis that looks for good/bad output.

- Use ‘Match Regular Expression’ activities to parse good/bad output

- Report with ‘Email’ or ‘Generate Alert’ activities

- Also possible to Remediate by using ‘Terminal’ activities
Device Health Monitoring
With CPO

- Can use ‘Terminal’ activities with CPO direct to device
- Can use ‘Terminal’ or Web Services activities to interact with another NMS that interacts with the device (router, switch, etc.)
Case Study: Network Programmability at Cisco Live Orlando 2013
Embedded Automation
Using EEM To Set Port Descriptions

EEM dynamically sets descriptions on ports based on last connected CDP neighbor.

Applet available from https://supportforums.cisco.com/docs/DOC-24529

```
SDCC_IDF_1.19#show run | begin event manager
event manager session cli username "nmsuser"
    event manager applet update-port-description
        event neighbor-discovery interface regexp GigabitEthernet.* cdp add
            action 1.0 cli command "enable"
            action 2.0 cli command "config t"
            action 3.0 cli command "interface $_nd_local_intf_name"
            action 4.0 cli command "description $_nd_cdp_entry_name:$nd_port_id"
    end

SDCC_IDF_1.19#show int gi0/14
GigabitEthernet0/14 is up, line protocol is up (connected)
    Hardware is Gigabit Ethernet, address is 040c.3f1.9d8e (bia 040c.3f1.9d8e)
    Description: SDCC_DMS_12.show.ciscolive.com:GigabitEthernet0/10
```

Embedded Automation
Converting Access Switches From Stage to Production

Switch from stage to production VLAN

Ping production default GW, and verify reachability

If reachable, reconfigure switch for production...

...If not, then revert to stage.

Applet available from https://supportforums.cisco.com/docs/DOC-34529
Monitor an Interface for Errors

- Interface errors can be the silent killer
- As the interface takes errors, performance suffers but no alerts are seen
- Using EEM, we can monitor for increases in errors and send notifications to the NMS

```plaintext
event manager applet error_monitor_GigabitEthernet0/1
  event interface name GigabitEthernet0/1 parameter input_errors
    entry-op ge entry-val 5 entry-type increment poll-interval 10
    action 1.0 syslog priority errors msg "Interface GigabitEthernet0/1 has seen $interface_delta_value input errors in the past 10 seconds"
    action 2.0 cli command "enable"
    action 3.0 cli command "show int GigabitEthernet0/1 | inc error"
    action 4.0 syslog priority errors msg "$cli_result"
```

```
Jun 2 02:34:45.381: %HA_EM-3-LOG: error_monitor_GigabitEthernet0/1: Interface GigabitEthernet0/1 has seen 7 input errors in the past 10 seconds
Jun 2 02:34:45.417: %HA_EM-3-LOG: error_monitor_GigabitEthernet0/1: 59 input errors, 58 CRC, 1 frame, 0 overrun, 0 ignored
```
Recommended Reading
BRKCDN-1005


Cisco Beyond Has a New Home

Script repository
Over 130+ open source scripts, learn by example

Discussion forums
Ask questions, get answers

Video tutorials

http://www.cisco.com/go/ciscobeyond
Check out the onePK DevNet site to get started building apps

http://developer.cisco.com/web/onepk
Join Cisco Support Communities!

- **Free** for anyone with Cisco.com registration
- Get **timely** answers to your technical questions
- Find **relevant** technical documentation
- Engage with over 200,000 **top technical experts**
- **Seamless** transition from discussion to TAC Service Request (*Cisco customers and partners only*)

The Cisco Support Community is your one-stop community destination from Cisco for sharing current, real-world technical support knowledge with peers and experts.

supportforums.cisco.com
supportforums.cisco.mobi
...That's the way it IS.
Call to Action…

Visit the World of Solutions:-

- Cisco Campus
- Walk-in Labs
- Technical Solutions Clinics

- Meet the Engineer

- Lunch Time Table Topics, held in the main Catering Hall

- Recommended Reading: For reading material and further resources for this session, please visit www.pearson-books.com/CLMilan2014
Complete Your Online Session Evaluation

- Complete your online session evaluation
- Complete four session evaluations and the overall conference evaluation to receive your Cisco Live T-shirt