Configuring the Cisco APIC Using the REST API

- Expanding the APIC Cluster Using the REST API, on page 1
- Contracting the APIC Cluster Using the REST API, on page 2
- Reducing the APIC Cluster Size, on page 4
- Switching Over Active APIC with Standby APIC Using REST API, on page 4
- Registering an Unregistered Switch Using the REST API, on page 5
- Adding a Switch Before Discovery Using the REST API, on page 6
- Removing a Switch to Maintenance Mode Using the REST API, on page 6
- Inserting a Switch to Operational Mode Using the REST API, on page 7
- Configuring a Remote Location Using the REST API, on page 7
- Sending an On-Demand Tech Support File Using the REST API, on page 8
- Finding Your Switch Inventory Using the REST API, on page 8

Expanding the APIC Cluster Using the REST API

The cluster drives its actual size to the target size. If the target size is higher than the actual size, the cluster size expands.

Procedure

Step 1 Set the target cluster size to expand the APIC cluster size.

Example:

```
POST
https://<IP address>/api/node/mo/uni/controller.xml
<infraClusterPol name='default' size=3/>
```

Step 2 Physically connect the APIC controllers that you want to add to the cluster.

Contracting the APIC Cluster Using the REST API

Use this procedure to shrink the cluster size by removing controllers. For more information about contracting the cluster size, see Contracting the Cisco APIC Cluster.

Note Beginning with Cisco APIC Release 6.0(2), two additional properties are added to the API command to allow forcing the decommission operation. The new object properties are:

- `infraClusterPol:shrink`
  - `false`: (default) If the target cluster size (`infraClusterPol:size`) is less than the current operational cluster size, the APICs to be removed must be decommissioned manually, as in earlier releases.
  - `true`: If the target cluster size is less than the current operational cluster size, a cluster shrink decommission is triggered. The APICs to be removed are decommissioned automatically, beginning with the APIC with the highest controller ID number.
- `infraWiNode:force`
  - `false`: (default) The decommission proceeds only if the cluster is not in an unhealthy or upgrade state, where a decommission may not be proper.
  - `true`: The decommission proceeds regardless of the cluster state.

This example shows how to contract the cluster from three APIC controllers to a single controller. To achieve a target size of one, APIC3 and APIC2 must be decommissioned, in that order.

Procedure

Step 1 Set the target cluster size so as to contract the APIC cluster size.

When the cluster size is reduced with `shrink='true'`, the APICs to be removed are decommissioned automatically. Otherwise, they must be decommissioned manually.

Example:

Cisco APIC Release 6.0(1) and earlier:

```
POST
https://<IP address>/api/node/mo/uni/controller.xml
<infraClusterPol name='default' size=1 />
```

You must now manually decommission the APICs to be removed, as shown in the next steps.

Example:

Cisco APIC Release 6.0(2) and later, using 'shrink' property:

```
POST
https://<IP address>/api/node/mo/uni/controller.xml
<infraClusterPol name='default' size=1 shrink='true' />
```

With `shrink='true'`, the following steps can be skipped. The APICs to be removed are decommissioned automatically.

```
POST
https://<IP address>/api/node/mo/uni/controller.xml
<infraClusterPol name='default' size=1 shrink='false' />
```

With `shrink='false'`, you must now manually decommission the APICs to be removed, as shown in the next steps.

Step 2 Decommission APIC3 on APIC1 for cluster contraction.

Example:

Cisco APIC Release 6.0(1) and earlier:

```
POST
https://<IP address>/api/node/mo/topology/pod-1/node-1/av.xml
<infraWiNode id=3 adminSt='out-of-service'/>
```

The decommission proceeds only if the cluster is in a healthy state.

Example:

Cisco APIC Release 6.0(2) and later, using 'force' property:

```
POST
https://<IP address>/api/node/mo/topology/pod-1/node-1/av.xml
<infraWiNode id=3 adminSt='out-of-service' force='true' />
```

With `force='true'`, the decommission proceeds regardless of the cluster state.

Step 3 Decommission APIC2 on APIC1 for cluster contraction.

Example:

Cisco APIC Release 6.0(1) and earlier:

```
POST
https://<IP address>/api/node/mo/topology/pod-1/node-1/av.xml
<infraWiNode id=2 adminSt='out-of-service'/>
```

The decommission proceeds only if the cluster is in a healthy state.

Example:

Cisco APIC Release 6.0(2) and later, using 'force' property:

```
POST
https://<IP address>/api/node/mo/topology/pod-1/node-1/av.xml
<infraWiNode id=2 adminSt='out-of-service' force='false' />
```

With `force='false'`, the decommission proceeds only if the cluster is in a healthy state.

The operation cluster size shrinks only after the last appliance is decommissioned, and not after the administrative size is changed. Verify after each controller is decommissioned that the operational state of the controller is unregistered and that the controller is no longer in service in the cluster.

Note If a decommissioned APIC controller is not promptly removed from the fabric, it might be rediscovered, which could cause problems. In that case, follow the instructions in Reducing the APIC Cluster Size to remove the controller.

Reducing the APIC Cluster Size

Follow these guidelines to reduce the APIC cluster size and decommission the APIC controllers that are removed from the cluster:

Note Failure to follow an orderly process to decommission and power down APIC controllers from a reduced cluster can lead to unpredictable outcomes. Do not allow unrecognized APIC controllers to remain connected to the fabric.

- Reducing the cluster size increases the load on the remaining APIC controllers. Schedule the APIC controller size reduction at a time when the demands of the fabric workload will not be impacted by the cluster synchronization.
- If one or more of the APIC controllers' health status in the cluster is not "fully fit", remedy that situation before proceeding.
- Reduce the cluster target size to the new lower value. For example if the existing cluster size is 6 and you will remove 3 controllers, reduce the cluster target size to 3.
- Starting with the highest numbered controller ID in the existing cluster, decommission, power down, and disconnect the APIC controller one by one until the cluster reaches the new lower target size.

  Upon the decommissioning and removal of each controller, the APIC synchronizes the cluster.

Note After decommissioning an APIC controller from the cluster, power it down and disconnect it from fabric. Before returning it to service, do a wiped clean back to factory reset.

- Cluster synchronization stops if an existing APIC controller becomes unavailable. Resolve this issue before attempting to proceed with the cluster synchronization.
- Depending on the amount of data the APIC must synchronize upon the removal of a controller, the time required to decommission and complete cluster synchronization for each controller could be more than 10 minutes per controller.

Note Complete the entire necessary decommissioning steps, allowing the APIC to complete the cluster synchronization accordingly before making additional changes to the cluster.

Switching Over Active APIC with Standby APIC Using REST API

Use this procedure to switch over an active APIC with standby APIC using REST API.

Procedure

Switch over active APIC with standby APIC.

```
URL for POST: https://ip
address/api/node/mo/topology/pod-initiator_pod_id/node-initiator_id/av.xml
Body: <infraWiNode id=outgoing_apic_id targetMbSn=backup-serial-number/>
where initiator_id = id of an active APIC other than the APIC being replaced.
pod-initiator_pod_id = pod ID of the active APIC
backup-serial-number = serial number of standby APIC
```

Example:

```
https://ip address/api/node/mo/topology/pod-1/node-1/av.xml
<infraWiNode id=2 targetMbSn=FCH1750V00Q/>
```

Registering an Unregistered Switch Using the REST API

Use this procedure to register a switch from the **Nodes Pending Registration** tab on the **Fabric Membership** work pane using the REST API.

Note This procedure is identical to "Adding a Switch Before Discovery Using the REST API". When you apply the code, the system determines if the node exists and, if not, adds it. If the node does exist, the system registers it.

Procedure

Add a switch description.

Example:

```
POST
https://<IP address>/api/policymgr/mo/uni.xml

<!-- /api/policymgr/mo/uni.xml -->
<polUni>
<ctrlrInst>
    <fabricNodeIdentPol>
        <fabricNodeIdentP  nodeType="tier-2-leaf" podId="1" serial="XXXXXXXXX"
        name="tier-2-leaf-leaf1" nodeId="101"/>

    </fabricNodeIdentPol>
</ctrlrInst>
</polUni>
```

Adding a Switch Before Discovery Using the REST API

Use this procedure to add a switch to the **Nodes Pending Registration** tab on the **Fabric Membership** work pane using the REST API.

Note This procedure is identical to "Registering an Unregistered Switch Using the REST API". When you apply the code, the system determines if the node exists and, if not, adds it. If the node does exist, the system registers it.

Procedure

Add a switch description.

Example:

```
POST
https://<IP address>/api/policymgr/mo/uni.xml

<!-- /api/policymgr/mo/uni.xml -->
<polUni>
<ctrlrInst>
    <fabricNodeIdentPol>
        <fabricNodeIdentP  nodeType="tier-2-leaf" podId="1" serial="XXXXXXXXX"
        name="tier-2-leaf1" nodeId="101"/>

    </fabricNodeIdentPol>
</ctrlrInst>
</polUni>
```

Removing a Switch to Maintenance Mode Using the REST API

Use this procedure to remove a switch to maintenance mode using the REST API.

Procedure

Remove a switch to maintenance mode.

Example:

```
POST
https://<IP address>/api/node/mo/uni/fabric/outofsvc.xml

<fabricOOServicePol
    descr=""
    dn=""
    name="default"
```

```
    nameAlias=""
    ownerKey=""
    ownerTag="">
  <fabricRsDecommissionNode
      debug="yes"
      dn=""
      removeFromController="no"
      tDn="topology/pod-1/node-102"/>
</fabricOOServicePol>
```

Inserting a Switch to Operational Mode Using the REST API

Use this procedure to insert a switch to operational mode using the REST API.

Procedure

Insert a switch to operational mode.

Example:

```
POST
https://<IP address>/api/node/mo/uni/fabric/outofsvc.xml

<fabricOOServicePol
    descr=""
    dn=""
    name="default"
    nameAlias=""
    ownerKey=""
    ownerTag="">
  <fabricRsDecommissionNode
      debug="yes"
      dn=""
      removeFromController="no"
      tDn="topology/pod-1/node-102"
      status="deleted"/>
</fabricOOServicePol>
```

Configuring a Remote Location Using the REST API

This procedure explains how to create a remote location using the REST API.

```
<fileRemotePath name="local" host="host or ip" protocol="ftp|scp|sftp" remotePath="path to
 folder" userName="uname" userPasswd="pwd" />
```

Sending an On-Demand Tech Support File Using the REST API

Procedure

Step 1 Set the remote destination for a technical support file using the REST API, by sending a POST with XML such as the following example:

Example:

```
<fileRemotePath userName="" remotePort="22" remotePath="" protocol="sftp" name="ToSupport"
 host="192.168.200.2"
dn="uni/fabric/path-ToSupport" descr="">

<fileRsARemoteHostToEpg tDn="uni/tn-mgmt/mgmtp-default/oob-default"/>

</fileRemotePath>
```

Step 2 Generate an on-demand technical support file using the REST API by sending a POST with XML such as the following:

Example:

```
<dbgexpTechSupOnD upgradeLogs="no" startTime="unspecified" name="Tech_Support_9-20-16"
  exportToController="no" endTime="unspecified" dn="uni/fabric/tsod-Tech_Support_9-20-16"
descr=""
  compression="gzip" category="forwarding" adminSt="untriggered">
    <dbgexpRsExportDest tDn="uni/fabric/path-ToSupport"/>
    <dbgexpRsTsSrc tDn="topology/pod-1/node-102/sys"/>
    <dbgexpRsTsSrc tDn="topology/pod-1/node-103/sys"/>
    <dbgexpRsTsSrc tDn="topology/pod-1/node-101/sys"/>
    <dbgexpRsData tDn="uni/fabric/tscont"/>
</dbgexpTechSupOnD>
<fabricFuncP>
    <fabricCtrlrPGrp name="default">
        <fabricRsApplTechSupOnDemand tnDbgexpTechSupOnDName=" Tech_Support_9-20-16"/>
    </fabricCtrlrPGrp>
</fabricFuncP>
```

Finding Your Switch Inventory Using the REST API

This section explains how to find your switch model and serial numbers using the REST API

Procedure

Find your switch inventory as follows:

Example:

```
GET
https://192.0.20.123/api/node/mo/topology/pod-1.json?query-target=children&target-subtree-class=fabricNode
```

The following response is returned:

```
response:
  {
     "totalCount":"8",
     "imdata":
     [{
         "fabricNode":{
           "attributes":{
              "adSt":"on",
              "childAction":"",
              "delayedHeartbeat":"no",
              "dn":"topology/pod-1/node-103",
              "fabricSt":"active",
              "id":"103",
              "lcOwn":"local",
              "modTs":"2016-10-08T14:49:35.665+00:00",
              "model":"N9K-C9396PX",
              "monPolDn":"uni/fabric/monfab-default",
              "name":"leaf3",
              "nameAlias":"",
              "role":"leaf",
              "serial":"TEP-1-103",
              "status":"","uid":"0",
              "vendor":"Cisco Systems, Inc",
              "version":""}
            }
     },{
         "fabricNode":{
           "attributes":{
             "adSt":"on",
             "childAction":"",
             "delayedHeartbeat":"no",
             "dn":"topology/pod-1/node-105",
             "fabricSt":"active",
             "id":"105",
             "lcOwn":"local",
             "modTs":"2016-10-08T14:47:52.011+00:00",
             "model":"N9K-C9508",
             "monPolDn":"uni/fabric/monfab-default",
             "name":"spine2",
             "nameAlias":"",
             "role":"spine",
             "serial":"TEP-1-105","status":"",
             "uid":"0",
             "vendor":"Cisco Systems, Inc",
             "version":""
          ...
       [TRUNCATED]
          ...
     }
```