Using NETCONF and RESTCONF

NETCONF uses a simple RPC-based (Remote Procedure Call) mechanism to facilitate communication between a client and a server. The client can be a script or an application running as part of a network manager. The server is typically a network device (switch or router).

NETCONF uses Secure Shell Version 2(SSHv2) as the transport layer across network devices and RESTCONF uses HTTP.

NETCONF and RESTCONF also support capability discovery and model downloads. Supported models are discovered using the ietf-netconf-monitoring model. Revision dates for each model are shown in the capabilities response. Data models are available for optional download from a device using the get-schema rpc. You can use these YANG models to understand or export the data model.

To use NETCONF and RESTCONF you must complete all the required tasks as per the Configuring Programmability, page 1-4 section. The following shows examples of the RPCs you can send and the kind of action that is performed.

- Examples for NETCONF RPCs, page 3-1
- Examples for RESTCONF RPCs:, page 3-2

Examples for NETCONF RPCs

Get the running-configuration of the switch by sending the following RPC:

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <native xmlns="http://cisco.com/ns/yang/ned/ios"/>
    </filter>
  </get-config>
</rpc>
```

Change the description of an interface by sending the following RPC

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <native xmlns="http://cisco.com/ns/yang/ned/ios">
        <interface>
```

```
          <TenGigabitEthernet>
            <name>4/1</name>
            <description>to_distribution</description>
           </TenGigabitEthernet>
        </interface>
      </native>
    </config>
  </edit-config>
</rpc>
```

Remove the description from an interface by sending the following RPC

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <native xmlns="http://cisco.com/ns/yang/ned/ios">
        <interface>
          <TenGigabitEthernet>
            <name>4/1</name>
            <description xc:operation="delete"/>
           </TenGigabitEthernet>
        </interface>
      </native>
    </config>
  </edit-config>
</rpc>
```

Examples for RESTCONF RPCs:

Get the TFTP source interface by sending the following RPC:

```
GET http://10.106.30.33:80/restconf/api/running/native/ip/tftp/source-interface
```

Configure the TFTP source interface by sending the following RPC:

```
PATCH
http://10.106.30.33:80/restconf/api/running/native/ip/tftp/source-interface/GigabitEth
ernet
payload = "{\n    \"GigabitEthernet\": \"2/2\"\n}"
```

Enter a HTTP delete request by sending the following RPC:

```
DELETE http://10.106.30.33:55080/api/running/native/ip/tftp/source-interface/
```

Note For the HTTP delete request do not use:
`http://10.106.30.33:80/restconf/api/running/native/ip/tftp/source-interface/`