

NETCONF Usage Guide

APIs

- Interface
- VLAN
- SSH/Telnet
- Routing
- IP Interface
- Port Configuration
- STP Global
- STP Per-Port

This guide provides practical NETCONF XML examples to help you manage and configure your network device. NETCONF (Network Configuration Protocol) is a standardized protocol for managing network devices, using XML-based data encoding for configuration and operational data.

These examples cover common tasks such as:

- Retrieving current configurations and operational state using simplified RPCs.
- Modifying device settings for interfaces, VLANs, remote access (SSH/Telnet), IP routing, IP addressing, port-specific features, Port Channels (LAGs), and Spanning Tree Protocol (STP) using simplified RPCs.

Key NETCONF Concepts Used in This Guide:

- **<rpc>**: The root element for every NETCONF request. For the simplified RPCs shown, **message-id** and the base NETCONF namespace on the **<rpc>** tag are often omitted for brevity, though standard clients might still send them. The server will internally track message IDs.
 - **<get>**: Retrieves operational state data and configuration data.
 - **<config>**: Used within **<edit-config>** to enclose the configuration data you want to apply.
 - **xmlns (XML Namespace)**: An attribute used to qualify XML elements and attributes, preventing naming conflicts. Each data model (e.g., for VLANs, interfaces) will have its own namespace.
 - **operation attribute**: Used within **<edit-config>** on specific data nodes to indicate the action to perform (e.g., **create**, **delete**, **merge**, **replace**).
-

1. Interface Information (Custom GET)

This section covers how to retrieve detailed information about all network interfaces on the device. The response format for this specific `<get>` operation is a custom XML structure, where interface data is directly under an `<rpc-reply>` root element, with each interface name as a dynamic tag.

Get Interface Information

Retrieves operational status and configuration for all interfaces.

```
<rpc>
  <get>
    <interfaces xmlns="yang:get_interface"/>
  </get>
</rpc>
]]>]]>
```

2. VLAN Configuration

Virtual Local Area Networks (VLANs) allow you to segment your network. This section shows how to view and manage VLANs. The response for GET operations will be an `<rpc-reply>` with a `<data>` wrapper, and the `<vlans>` element within will use `xmlns="yang:vlan"`. The response for edit operations will be a simple `<rpc-reply><ok/></rpc-reply>`.

Get VLANs

Retrieves a list of all configured VLANs and their names. The response format for this specific `<get>` operation is a custom XML structure, where VLAN data is directly under a `<vlans>` element.

```
<rpc>
  <get>
    <vlans xmlns="yang:get_vlan"/>
  </get>
</rpc>
]]>]]>
```

Set VLAN

Creates a new VLAN or modifies an existing one.

```
<rpc>
  <edit-config>
    <config>
```

```

    <vlans xmlns="yang:set_vlan">
      <vlan>
        <id>79</id>
        <name>vlan_79</name>
      </vlan>
    </vlans>
  </config>
</edit-config>
</rpc>
]]>]]>

```

3. SSH Server Configuration

Secure Shell (SSH) provides secure remote access to the device. The response for GET operations will be an `<rpc-reply>` with the `<ssh-server-config>` data directly under it, using `xmlns="yang:ssh"`. The response for edit operations will be a simple `<rpc-reply><ok/></rpc-reply>`.

Get SSH Status

Checks if the SSH server is currently enabled or disabled.

```

<rpc>
  <get>
    <ssh-server-config xmlns="yang:get_ssh"/>
  </get>
</rpc>
]]>]]>

```

Enable SSH

Turns on the SSH server.

```

<rpc>
  <edit-config>
    <config>
      <ssh-server-config xmlns="yang:set_ssh">
        <enabled>true</enabled>
      </ssh-server-config>
    </config>
  </edit-config>
</rpc>
]]>]]>

```

Disable SSH

Turns off the SSH server.

```

<rpc>
  <edit-config>
    <config>
      <ssh-server-config xmlns="yang:set_ssh">
        <enabled>false</enabled>
      </ssh-server-config>
    </config>
  </edit-config>
</rpc>
]]>]]>

```

4. Telnet Server Configuration

Telnet provides remote access, but it's less secure than SSH as data is sent in clear text. The response for GET operations will be an `<rpc-reply>` with the `<telnet-server-config>` data directly under it, using `xmlns="yang:telnet"`. The response for edit operations will be a simple `<rpc-reply><ok/></rpc-reply>`.

Get Telnet Status

Checks if the Telnet server is currently enabled or disabled.

```

<rpc>
  <get>
    <telnet-server-config xmlns="yang:get_telnet"/>
  </get>
</rpc>
]]>]]>

```

Enable Telnet

Turns on the Telnet server.

```

<rpc >
  <edit-config>
    <config>
      <telnet-server-config xmlns="yang:set_telnet">
        <enabled>true</enabled>
      </telnet-server-config>
    </config>
  </edit-config>
</rpc>
]]>]]>

```

Disable Telnet

Turns off the Telnet server.

```
<rpc >
  <edit-config>
    <config>
      <telnet-server-config xmlns="yang:set_telnet">
        <enabled>false</enabled>
      </telnet-server-config>
    </config>
  </edit-config>
</rpc>
]]>]]>
```

5. Static Routing

Static routes manually define paths for IP traffic. The response for GET operations will be an `<rpc-reply>` with the `<routing>` data directly under it, using `xmlns="yang:route"`. The response for edit operations will be a simple `<rpc-reply><ok/></rpc-reply>`.

Get All Static Routes

Retrieves all configured static IP routes.

```
<rpc>
  <get>
    <routing xmlns="yang:get_route"/>
  </get>
</rpc>
]]>]]>
```

Add Static Route

Adds a new static IP route to the device's routing table. The `operation="create"` attribute indicates that a new route entry should be created.

```
<rpc>
  <edit-config>
    <config>
      <routing xmlns="yang:set_route">
        <static-routes>
          <route operation="create">
            <prefix>131.108.1.27</prefix>
            <mask>255.255.255.255</mask>
            <next-hop>131.108.1.28</next-hop>
          </route>
        </static-routes>
      </routing>
    </config>
  </edit-config>
</rpc>
```

```

    </routing>
  </config>
</edit-config>
</rpc>
]]>]]>

```

Delete Static Route

Removes an existing static IP route. The `operation="delete"` attribute specifies that the matching route entry should be removed.

```

<rpc>
  <edit-config>
    <config>
      <routing xmlns="yang:set_route">
        <static-routes>
          <route operation="delete">
            <prefix>131.108.1.27</prefix>
            <mask>255.255.255.255</mask>
          </route>
        </static-routes>
      </routing>
    </config>
  </edit-config>
</rpc>
]]>]]>

```

6. IP Interface

This section deals with configuring IP addresses and subnet masks on network interfaces.

Get All IP Interfaces

Retrieves the IP address configuration for all interfaces that have an IP address assigned.

```

<rpc>
  <get>
    <ip-interfaces xmlns="yang:get_ip_interface"/>
  </get>
</rpc>
]]>]]>

```

Set IP Interface

Assigns an IP address and subnet mask to a specified interface. The `operation="create"` attribute is used here to define a new IP address configuration on the interface. If an IP configuration already exists, this

might update it or add a secondary address depending on the device's behavior (often, **merge** or **replace** operations are used for updates).

```
<rpc>
  <edit-config>
    <config>
      <ip-interfaces xmlns="yang:set_ip_interface">
        <interface>
          <name>te1/0/1</name>
          <ip-address>131.108.1.27</ip-address>
          <mask-prefix>255.255.255.0</mask-prefix>
        </interface>
      </ip-interfaces>
    </config>
  </edit-config>
</rpc>
]]>]]>
```

7. Port Configuration

This section covers various settings for physical switch ports, such as administrative status, speed, description, and VLAN membership modes (access/trunk).

Enable/Disable Ports

Sets the administrative status of specified ports to 'up' (enabled) or 'down' (disabled).

```
<rpc>
  <edit-config>
    <config>
      <port-configurations xmlns="yang:set_port_config">
        <port><name>te1/0/1</name><admin-status>up</admin-status></port>
        <port><name>te1/0/2</name><admin-status>up</admin-status></port>
      </port-configurations>
    </config>
  </edit-config>
</rpc>
]]>]]>
```

Set Description and Speed

Configures the administrative status, adds a descriptive label, and sets the speed for a port.

```
<rpc>
  <edit-config>
    <config>
```

```

    <port-configurations xmlns="yang:set_port_config">
      <port>
        <name>te1/0/1</name>
        <admin-status>up</admin-status>
        <description>PC</description>
        <speed>10000</speed>
      </port>
      <port>
        <name>te1/0/2</name>
        <admin-status>up</admin-status>
        <description>Core_Switch</description>
      </port>
    </port-configurations>
  </config>
</edit-config>
</rpc>
]]>]]>

```

Configure Access VLAN

Sets a port to 'access' mode and assigns it to a specific VLAN. Packets on an access port are untagged and belong to this single VLAN.

```

<rpc>
  <edit-config>
    <config>
      <port-configurations xmlns="yang:set_port_config">
        <port>
          <name>te1/0/9</name>
          <switchport>
            <mode>access</mode>
            <access><vlan-id>100</vlan-id></access>
          </switchport>
        </port>
      </port-configurations>
    </config>
  </edit-config>
</rpc>
]]>]]>

```

Configure Trunk VLANs

Sets a port to 'trunk' mode, allowing it to carry traffic for multiple VLANs. You can specify which VLANs are allowed and set a native VLAN (for untagged traffic).

```

<rpc>
  <edit-config>
    <config>
      <port-configurations xmlns="yang:set_port_config">

```



```

    <port>
      <name>te1/0/5</name>
      <switchport>
        <mode>trunk</mode>
        <trunk>
          <allowed-vlans>10, 20, 30-35, 40</allowed-vlans>
          <native-vlan-id>1</native-vlan-id>
        </trunk>
      </switchport>
    </port>
  </port-configurations>
</config>
</edit-config>
</rpc>
]]>]]>

```

Enable/Disable STP on Port

Configures Spanning Tree Protocol (STP) on a specific port. This is typically for per-port STP settings if the device supports it, distinct from global STP. (Note: This example assumes per-port STP control. Global STP is covered in the next section.)

```

<rpc>
  <edit-config>
    <config>
      <port-configurations xmlns="yang:set_port_config">
        <port>
          <name>te1/0/6</name>
          <stp><enabled>true</enabled></stp>
        </port>
      </port-configurations>
    </config>
  </edit-config>
</rpc>
]]>]]>

```

8. Spanning Tree Protocol (STP) Global

Spanning Tree Protocol prevents broadcast storms and loop issues in a switched network. This section covers global STP settings.

Get Global STP Status

Checks if STP is globally enabled or disabled on the device.

```

<rpc>
  <get>

```

```
<stp-global-config xmlns="yang:get_stp"/>
</get>
</rpc>
]]>]]>
```

Enable Global STP

Turns on STP for the entire device.

```
<rpc>
  <edit-config>
    <target><running/></target>
    <config>
      <stp-global-config xmlns="yang:set_stp">
        <enabled>true</enabled>
      </stp-global-config>
    </config>
  </edit-config>
</rpc>
]]>]]>
```

Disable Global STP

Turns off STP for the entire device.

```
<rpc>
  <edit-config>
    <target><running/></target>
    <config>
      <stp-global-config xmlns="yang:set_stp">
        <enabled>false</enabled>
      </stp-global-config>
    </config>
  </edit-config>
</rpc>
]]>]]>
```