OpenConfig YANG

- About OpenConfig YANG, on page 1
- Guidelines and Limitations for OpenConfig YANG, on page 1
- Understanding Deletion of BGP Routing Instance, on page 9
- Verifying YANG, on page 10
- Enabling OpenConfig Support, on page 10

About OpenConfig YANG

OpenConfig YANG supports modern networking principles, such as declarative configuration and model-driven management and operations. OpenConfig provides vendor-neutral data models for configuration and monitoring of the network. And, helping with moving from a pull model to a push model, with subscriptions and event update streaming.

Beginning with Cisco NX-OS Release 9.2(1), support is added across a broad range of functional areas. Those include BGP, OSPF, Interface L2 and L3, VRFs, VLANs, and TACACs.

For additional information about OpenConfig YANG, see About OpenConfig YANG.

For the OpenConfig models for Cisco NX-OS 9.2(1), see YANG Models 9.2(1). OpenConfig YANG models are grouped by Cisco NX-OS release, so when the Cisco NX-OS release number changes, the last digits in the URL change.

Guidelines and Limitations for OpenConfig YANG

OpenConfig YANG has the following guidelines and limitations:

- For IPv4 and IPv6 addresses, you must provide the same operation for remove and delete for the IP address field (**oc-ip:ip** and **oc-ip:prefix_length**).

  For example:

  ```
  oc-ip:ip: remove
  oc-ip:prefix_length: remove
  ```

- Configuring BGP actions with **set med** and OSPF actions with metric in the same route-map via OpenConfig NETCONF is not recommended as the OSPF actions metric takes precedence over BGP **set med** property.

Use two different route-maps to set metrics under OSPF actions. Use **set-med** under BGP actions using separate route-maps.

We recommended that you do not change the metric of BGP actions to OSPF actions or OSPF actions to BGP actions of a route-map in a single payload.

- In order to have a valid BGP instance, an autonomous system (AS) number must be provided. Since there cannot be a default value for an AS number, any attempt to delete in NETCONF/OPENCONFIG <asn> without removing the BGP instance, results in the following highlighted error message:

```
764
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
message-id="urn:uuid:1ea09de2-605e-46aa-984b-9dfdad03354d">
  <nc:edit-config>
    <nc:target>
      <nc:running/>
    </nc:target>
    <nc:config>
      <network-instances xmlns="http://openconfig.net/yang/network-instance">
        <network-instance>
          <name>default</name>
          <protocols>
            <protocol>
              <identifier>BGP</identifier>
              <name>bgp</name>
              <bgp>
                <global>
                  <config nc:operation="delete">
                    <as>100</as>
                  </config>
                </global>
                <neighbors>
                  <neighbor>
                    <neighbor-address>1.1.1.1</neighbor-address>
                    <enable-bfd xmlns="http://openconfig.net/yang/bfd">
                      <config>
                        <enabled>true</enabled>
                      </config>
                    </enable-bfd>
                  </neighbor>
                </neighbors>
              </bgp>
            </protocol>
          </protocols>
        </network-instance>
      </network-instances>
    </nc:config>
  </nc:edit-config>
</nc:rpc>

##
Received:
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
message-id="urn:uuid:1ea09de2-605e-46aa-984b-9dfdad03354d">
    <rpc-error>
        <error-type>protocol</error-type>
        <error-tag>operation-failed</error-tag>
        <error-severity>error</error-severity>
        <error-message xml:lang="en">invalid property value , for property asn, class
bgpInst</error-message>
        <error-path>/config/network-instances</error-path>
    </rpc-error>
    <rpc-error>
```

```
        <error-type>protocol</error-type>
        <error-tag>operation-failed</error-tag>
        <error-severity>error</error-severity>
        <error-message xml:lang="en">invalid property value , for property asn, class
bgpInst Commit Failed</error-message>
        <error-path>/config/network-instances</error-path>
    </rpc-error>
</rpc-reply>
```

- The following OpenConfig YANG limitations exist for OC-BGP-POLICY:
  - Action type is always `permit` for `community-set` and `as-path-set`, which applies to the following containers:
    - `/bgp-defined-sets/community-sets/community-set/`
    - `/bgp-defined-sets/as-path-sets/as-path-set/`

    In OpenConfig YANG, there is no action type concept as there is in the CLI for `community-set` and `as-path-set`. Therefore, the action type is always `permit` for `community-set` and `as-path-set`.
  - The following OpenConfig YANG limitation applies to this container: `/bgp-defined-sets/community-sets/community-set/`

    In the CLI, `community-list` can have two different types: standard and expanded. However, in the OpenConfig YANG model, `community-set-name` has no such differentiation.

    When you create the `community-set-name` through OpenConfig YANG, the following things happen internally:
    - The `_std` suffix will be appended after `community-set-name` if `community-member` is in the standard form (AS:NN).
    - The `_exp` suffix will be appended after `community-set-name` if `community-member` is in the expanded form (regex):

      ```
      <community-set>
          <community-set-name>oc_commset1d</community-set-name>
          <config>
              <community-set-name>oc_commset1d</community-set-name>
              <community-member>0:1</community-member>
              <community-member>_1_</community-member>
          </config>
      </community-set>
      ```

      The preceding OpenConfig YANG configuration is mapped to the following CLI:

      ```
      ip community-list expanded oc_commset1d_exp seq 5 permit "_1_"
      ip community-list standard oc_commset1d_std seq 5 permit 0:1
      ```

  - The following OpenConfig YANG limitation applies to this container: `/bgp-conditions/match-community-set/config/community-set/`

    OpenConfig YANG can only map to one `community-set`, while the CLI can match to multiple instances of the `community-set`:
    - In the CLI:

```
ip community-list standard 1-1 seq 1 permit 1:1
  ip community-list standard 1-2 seq 1 permit 1:2
  ip community-list standard 1-3 seq 1 permit 1:3
route-map To_LC permit 10
  match community 1-1 1-2 1-3
```

- The corresponding OpenConfig YANG payload follows:

```
<config>
  <routing-policy xmlns="http://openconfig.net/yang/routing-policy">
    <defined-sets>
      <bgp-defined-sets xmlns="http://openconfig.net/yang/bgp-policy">
        <community-sets>
          <community-set>
            <community-set-name>cs</community-set-name>
            <config>
              <community-set-name>cs</community-set-name>
              <community-member>1:1</community-member>
              <community-member>1:2</community-member>
              <community-member>1:3</community-member>
            </config>
          </community-set>
        </community-sets>
      </bgp-defined-sets>
    </defined-sets>
    <policy-definitions>
      <policy-definition>
        <name>To_LC</name>
        <statements>
          <statement>
            <name>10</name>
            <conditions>
              <bgp-conditions xmlns="http://openconfig.net/yang/bgp-policy">
                <match-community-set>
                  <config>
                    <community-set>cs</community-set>
                  </config>
                </match-community-set>
              </bgp-conditions>
            </conditions>
          </statement>
        </statements>
      </policy-definition>
    </policy-definitions>
  </routing-policy>
</config>
```

As a workaround, create one community with multiple statements through OpenConfig YANG:

```
ip community-list standard cs_std seq 5 permit 1:1
  ip community-list standard cs_std seq 10 permit 1:2
  ip community-list standard cs_std seq 15 permit 1:3
route-map To_LC permit 10
  match community cs_std
```

- The following OpenConfig YANG limitation applies to this container: `/bgp-conditions/state/next-hop-in`

  In OpenConfig YANG, the `next-hop-in` type is an IP address, but in the CLI, it is an IP prefix.

While creating the `next-hop-in` through OpenConfig YANG, the IP address is converted to a "/32" mask prefix in the CLI configuration. For example:

- Following is an example of `next-hop-in` in the OpenConfig YANG payload:

```
<policy-definition>
    <name>sc0</name>
    <statements>
        <statement>
            <name>5</name>
            <conditions>
                <bgp-conditions xmlns="http://openconfig.net/yang/bgp-policy">

                    <config>
                        <next-hop-in>2.3.4.5</next-hop-in>
                    </config>
                </bgp-conditions>
            </conditions>
        </statement>
    </statements>
</policy-definition>
```

- Following is an example of the same information in the CLI:

```
ip prefix-list IPV4_PFX_LIST_OPENCONFIG_sc0_5 seq 5 permit 2.3.4.5/32
route-map sc0 permit 5
  match ip next-hop prefix-list IPV4_PFX_LIST_OPENCONFIG_sc0_5
```

- The following NX-OS limitations exist for OC-BGP-POLICY:
  - `/bgp-actions/set-community/config/method` enum "REFERENCE" is not supported.
  - enum "SELF", which is supported in the OpenConfig YANG model for `/bgp-actions/config/set-next-hop`, is not supported.
- For OC-BGP-POLICY, `/bgp-conditions/match-community-set/config/community-set` get mapped only to `match community <community-set>_std`, so only standard community is supported. Match to expanded community set is not supported.
- There is a limitation in replacing `match-tag-set` because defined sets for `tag-sets` are not currently implemented.

  Currently, replacing `match-tag-set` appends the values. To replace `match-tag-set`, delete it, then create it again.
- The following guidelines and limitations apply to OSPF OpenConfig YANG:
  - If you configure and remove an area configuration in OSPF, the deleted areas (stale entries) are still shown in DME. Those stale area entries are shown in the GETCONFIG/GET output in OpenConfig YANG.
  - Only one area is supported in OpenConfig YANG in the OSPF policy `match ospf-area` configuration. In the CLI, you can configure to match multiple areas, such as `match ospf-area 100 101`. However, in OpenConfig YANG, you can configure only one area (for example, `match ospf-area 100`).

- The area virtual-link and area interface configurations payload cannot go under the same area list. Split the area container payload as a Virtual link area and interface area in the same payload.
- The MD5 authentication string cannot be configured in OSPF OpenConfig YANG.

  In the OSPF model, `Authentication-type` is defined for the Authentication:

  ```
  leaf authentication-type {
    type string;
    description
      "The type of authentication that should be used on this
      interface";
  }
  ```

  OSPF OpenConfig YANG does not support an option for authentication password.
- The OSPF area authentication configuration is not supported. For example, `area 0.0.0.200 authentication message-digest` cannot be configured from OpenConfig YANG.
- The OSPF/BGP instance configuration that falls under default VRF (for example, **router ospf 1/router bgp 1**) is not deleted when you delete the Protocols container with the default network instance.

- The following are guidelines and limitations for VLAN configuration between the OpenConfig payload and the Cisco Nexus 9000 interfaces:
  - When you attempt to simultaneously configure a trunk-mode interface and trunk VLANs in the same OpenConfig payload, the configuration does not complete successfully. However, when you split the payload so that the trunk-mode interface is sent first, then the trunk VLANs are sent, the configuration completes successfully.

    On Cisco NX-OS interfaces, the default interface mode is **access**. To implement any trunk-related configurations, you must first change the interface mode to **trunk**, then configure the trunk VLAN ranges. Do these configurations in separate payloads.

    The following examples show the separate payloads for the configuring trunk mode and VLAN ranges.

    Example 1, payload configuring the interface to trunk mode.

    ```
    <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
      <edit-config>
        <target>
          <running/>
        </target>
        <config>
          <interfaces xmlns="http://openconfig.net/yang/interfaces">
            <interface>
              <name>eth1/47</name>
              <subinterfaces>
                <subinterface>
                  <index>0</index>
                  <config>
                    <index>0</index>
                  </config>
                </subinterface>
              </subinterfaces>
              <ethernet xmlns="http://openconfig.net/yang/interfaces/ethernet">
                <switched-vlan xmlns="http://openconfig.net/yang/vlan">
                  <config>
                    <interface-mode>TRUNK</interface-mode>
                  </config>
    ```

```
            </switched-vlan>
          </ethernet>
        </interface>
      </interfaces>
    </config>
  </edit-config>
</rpc>
```

Example 2, payload configuring the VLAN ranges.

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <interfaces xmlns="http://openconfig.net/yang/interfaces">
        <interface>
          <name>eth1/47</name>
          <subinterfaces>
            <subinterface>
              <index>0</index>
              <config>
                <index>0</index>
              </config>
            </subinterface>
          </subinterfaces>
          <ethernet xmlns="http://openconfig.net/yang/interfaces/ethernet">
            <switched-vlan xmlns="http://openconfig.net/yang/vlan">
              <config>
                <native-vlan>999</native-vlan>
                <trunk-vlans xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
nc:operation="delete">1..4094</trunk-vlans>
                <trunk-vlans>401</trunk-vlans>
                <trunk-vlans>999</trunk-vlans>
              </config>
            </switched-vlan>
          </ethernet>
        </interface>
      </interfaces>
    </config>
  </edit-config>
</rpc>
```

- Because of the design of OpenConfig YANG, when you configure VLANs, there must be no overlap between the VLANs in the payload and the VLANs already configured on an interface. If an overlap exists, the configuration through OpenConfig is not successful. Make sure that the VLANs configured on an interface are different from the VLANs in the OpenConfig payload. Pay particular attention to the starting and ending VLANs in a range.

- Queueing stats for Hig (ii) ports is not supported.
- You do not see the tx-packets, or bytes, and drop-packets per unicast, multicast, or broadcast queue. The stats that display in the OC response are a sum of the ucast, mcast, and bcast queues per qos-group.
- OpenConfig YANG does not support stats for a QoS policy that is applied at the VLAN level.
- The ingress queue drop count that can be retrieved through OC can be displayed at the slice/port/queue level depending on the platform.
- The following is the guideline and limitation for OpenConfig configurations for switchport, shut/no shut, MTU, and mac-address:

- An ascii reload is required when configuring switchport, shut/no shut, MTU, and mac-address. Using a binary reload results in the configuration being lost.

- The following state containers are implemented for the OpenConfig ACL at interface-ref level:
  - `/acl/interfaces/interface/interface-ref/state` for `acl/interfaces/state` container.
  - `acl/interfaces/interface/interface-ref/state/interface` for `read-onlyoc-if:interface` leaf.
  - `acl/interfaces/interface/interface-ref/state/subinterface` for `read-onlyoc-if:subinterface` leaf.

- The following system config containers are implemented for domain-name, login-banner, and motd-banner models:
  - `/system/config/domain-name for /top:System/top:dns-items/top:prof-items/top:Prof-list/top:dom-items/top:name container`
  - `system/config/login-banner for /top:System/top:userext-items/top:postloginbanner-items/top:message container`
  - `/system/config/motd-banner for /top:System/top:userext-items/top:preloginbanner-items/top:message container`

- The following new operational state OpenConfig paths are supported. Some paths have extra guidelines and limitations as mentioned below:
  - `/network-instances/network-instance/fdb/l2rib/mac-table`
    - Parent level queries for l2rib are supported at l2rib level. For example, you can query until `network-instances/network-instance/fdb/l2rib` but not at fdb level `network-instances/network-instance/fdb`.
  - `/interfaces/interface/routed-vlan/ipv4/neighbors/neighbor/state`
  - `/interfaces/interface/routed-vlan/ipv6/neighbors/neighbor/state`
    - For parent queries, the infrastructure retrieves all the keys for all the list items and a request is sent to the back end to populate the rest of the data for each of these list items. This means that the infrastructure must have the same view of the tree as the back end.

      For example, if the infrastructure only sees static entries, while the back end has static and dynamic entries, then for the list walk the infrastructure will only send requests for each static entry which will result in incomplete data. The paths with this limitation in the current release are `/interfaces/interface/routed-vlan/ipv6/neighbors/neighbor/state` and `/interfaces/interface/routed-vlan/ipv4/neighbors/neighbor/state`.

The data contains both dynamic and static ARP and ND entries if the exact path is given but would only contain the static entries if the parent path given.

- /network-instances/network-instance/protocols/protocol/bgp/rib/afi-safis/afi-safi/l2vpn-evpn/loc-rib/routes
- /network-instances/network-instance/protocols/protocol/bgp/rib/attr-sets
- /network-instances/network-instance/protocols/protocol/bgp/rib/communities
- /network-instances/network-instance/protocols/protocol/bgp/rib/ext-communities
- /network-instances/network-instance/connection-points/connection-point/endpoints/endpoint/vxlan/endpoint-peers
- /network-instances/network-instance/connection-points/connection-point/endpoints/endpoint/vxlan/endpoint-vnis

Understanding Deletion of BGP Routing Instance

With OpenConfig YANG network-instance (OCNI), when attempting to delete only the BGP configuration of the default VRF instead of deleting the entire BGP routing instance, BGP information might not be deleted at the protocols/BGP level. In this situation, when the delete is at the protocols or BGP level with the autonomous system number in the payload, only the configuration of the default VRF is deleted instead of removing the entire BGP routing instance.

Following is an example payload that would be used to delete the configuration under the default VRF in BGP.

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <network-instances xmlns="http://openconfig.net/yang/network-instance">
        <network-instance>
          <name>default</name>
          <protocols>
            <protocol>
              <identifier>BGP</identifier>
              <name>bgp</name>
             <bgp xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" nc:operation="delete">

                 <global>
                   <config>
                     <as>100</as>
                   </config>
                 </global>
               </bgp>
             </protocol>
           </protocols>
         </network-instance>
       </network-instances>
    </config>
  </edit-config>
</rpc>
```

Expected Behavior: The BGP routing instance itself should be deleted, which is the equivalent to **no router bgp 100**.

Actual Behavior: Only the BGP configuration under the default VRF is deleted, and there is no equivalent single CLI configuration.

Following is the running configuration before the delete operation:

```
router bgp 100
  router-id 1.2.3.4
  address-family ipv4 unicast
  vrf abc
    address-family ipv4 unicast
      maximum-paths 2
```

And following is the running configuration after the delete operation:

```
router bgp 100
  vrf abc
    address-family ipv4 unicast
      maximum-paths 2
```

Verifying YANG

Use the following commands to verify YANG settings: :

Table 1: YANG Verification

Command	Description
show telemetry yang direct-path cisco-nxos-device	Displays the paths which are supported.

Enabling OpenConfig Support

To enable or disable OpenConfig support on the programmability agents (NETCONF, RESTCONF and gRPC), configure "[no] feature openconfig". For example:

```
switch(config)# feature netconf
switch(config)# feature restconf
switch(config)# feature grpc
switch(config)# feature openconfig
```

Note In previous releases, mtx-openconfig-all RPM was downloaded separately and installed. This method is deprecated in 10.2(2) release.