Original and OpenConfig YANG

About YANG

Yet Another Next Generation (YANG) is a data modeling language for the definition of data that are transferred over network management protocols. YANG defines a primary construct to define a tree data structure and various commonly used types to represent devices configuration and operational status.

YANG is a textual language, it can be used to define a model. Cisco NX-OS supports the following YANG models:

- **Cisco NX-OS-device**

  This is a Cisco NX-OS proprietary YANG model, which reflects the DME REST definitions. This model is referred as 'Device' or 'Original' model in this document. If you prefer to have full access over NX-OS device, it is suggested to use the 'Original' model. For more information, see the NX-API REST guide.

- **Open Config**

  The primary purpose of this model is to abstract the common functionalities across networking management. The output is a collection of YANG models, which is referred as the Open Config model. For more information, see About OpenConfig YANG.

  See Yang NX Repository guide to view models supported for NX-OS. Open Config YANG models are grouped based on Cisco NX-OS release.

This chapter describes the usage and limitations of Original and Open Config models.

Cisco Device YANG

Cisco NX-OS retains a single DME database to represent a centralized view of the configuration and state. You can access DME through a secured NX-API REST interface. If you prefer to operate based on standard YANG semantics, you can access the device through below Original YANG namespace:

Cisco-NX-OS-device - http://cisco.com/ns/yang/cisco-nx-os-device.

Note The device YANG URL is used to define a unique namespace identifier. The above URL is an example and not the accessible HTTP URL.

Guidelines and Limitations

The original YANG represents the DME database and inherits some behavior which is not in accordance with standard YANG operation. The following are the guidelines and limitations for YANG model:

- If a property is a numeric value, DME supports to define string aliases for specific values. You can use an alias or the numeric value to configure the property and DME converts the alias to a numeric value. If you prefer to adhere to the YANG standards, you can avoid using an alias.
- If a property is bitmask, DME supports special keywords (+,-) to gradually change the number in the bitmask range. This syntax is shown in Device YANG and it is not complaint to YANG. If you prefer to adhere to YANG standards, you can avoid using special keywords.
- In Device YANG, ephemeral paths are identified with comment **Ephemeral data**. These paths are NX-OS specific nonpersistent high volume data and are managed differently from the rest of the models. You can retrieve only when `<GET> query's <FILTER>` parameters marked specific to the elements marked with the comment. For more information, see *Ephemeral data support*.
- Beginning with Cisco NX-OS Release 10.4(3)F, YANG is supported on 92348GC-X.
- Beginning with Cisco NX-OS Release 10.5(2)F, following properties are not supported under the corresponding level from CLI.
  - **Bgp/neighbors/neighbor/config/send-community**

    Send-community is not supported under neighbor through CLI
  - **Bgp/neighbors/neighbor/timers/config/minimum-advertisement-interval**

    minimum-advertisement-interval is not supported under neighbor through CLI
  - **bgp/neighbors/neighbor/route-reflector/config/route-reflector-client**

    route-reflector-client is not supported under neighbor through CLI
  - **bgp/peer-groups/peer-group/config/send-community**

    Send-community is not supported under template peer through CLI
  - **bgp/neighbors/neighbor/as-path-options/config/allow-own-as**
  - **gp/neighbors/neighbor/as-path-options/config/disable-peer-as-filter**

- **bgp/peer-groups/peer-group/route-reflector/config/route-reflector-client**

  route-reflector-client is not supported under template peer through CLI
- **bgp/peer-groups/peer-group/as-path-options/config/allow-own-as**
- **bgp/peer-groups/peer-group/as-path-options/config/disable-peer-as-filter**

Migration from DME to Device YANG

If you are familiar with Cisco NX-OS DME interfaces, you can migrate existing DME to Device YANG. As the switches maintain one to one correspondence between DME and Device YANG models.

The following table shows representation of VRFs in DME and the Device YANG.

DME	Device YANG
sys +- name +- inst | +- name = default +- inst +- name = management	System +- name +- inst-items +- Inst-list | +- name = default +- Inst-list +- name = management

Make sure that you follow the below mentioned guidelines for migrating DME model to Device YANG model:

- The root element 'sys' MO is represented as 'System' root element in the YANG model.
- The DME MO is represented with the suffix '- items' in the YANG model.
- If DME MO has children of the same type, the Device YANG model adds an intermediate parent node for each child and use the suffix '-list' to represent the lists. This behavior is the same as OpenConfig.
- The DME property name remains the same in the Device YANG model.

About OpenConfig YANG

OpenConfig YANG model supports advanced networking standards, such as declarative configuration and model-driven management and operations. OpenConfig is a business data model for configuring and monitoring the network. This data model helps with moving from a pull model to a push model, with subscriptions and event update streaming.

Cisco NX-OS supports a wide range of functional areas, such as BGP, OSPF, Interface L2 and L3, VRFs, VLANs, and TACACs.

Guidelines and Limitations

The following are the guidelines and limitations of the OpenConfig YANG model:

Still guidelines and limitations required

- In networking protocol, you can add L2 MAC in a different way. When you use MAC leaf switch property values as input to perform an `NETCONF GET` operation, it is recommended to input the letters in MAC addresses in the uppercase in the forma of (AA:AA:AA:AA:AA:AA)". For an example, source-mac: 0A:0B:0C:0D:0E:0F.

OpenConfig Paths

Table 1: OpenConfig IP

Path	Description

`…/oc-ip:ip`
`…/oc-ip:prefix_length`
	For IPv4 and IPv6 addresses, you must provide the same operation to remove and delete the IP address field (**oc-ip:ip** and **oc-ip:prefix_length**)
	For example:
	`oc-ip:ip: remove`
	`oc-ip:prefix_length: remove`

Table 2: OpenConfig Network Instance

Path	Description
`bgp`	

Path	Description

	With OpenConfig YANG network-instance (OCNI), and if you want to delete only BGP configuration of default VRF instead of BGP routing instance, you cannot delete BGP information at protocol or BGP level. You can enter an autonomous system number in the payload and delete only the configuration of the default VRF and not the BGP routing instance.
	
	The following is an example for payload to delete the configuration in the default VRF OF 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="urn:ietf:params:xml:ns:netconf:base:1.0"`
	
	`operation="delete">`
	`<global>`
	`<config>`
	`<as>100</as>`
	`</config>`
	`</global>`
	`</bgp>`
	`</protocol>`
	`</protocols>`
	`</network-instance>`
	`</network-instances>`
	`</config>`
	`</edit-config>`
	`</rpc>`
	
	Expected Behavior: The BG routing instance is deleted, which is similar to configuring `no route bgp 100`.
	
	Actual Behaviour: Only the BGP configuration in default VRF is deleted, and there is no similar CLI configuration.
	
	The following mentioned below is the running configuration before configuring delete operation:

Path	Description

	`router bgp 100`
	`  router-id 1.2.3.4`
	`  address-family ipv4 unicast`
	`  vrf abc`
	`  address-family ipv4 unicast`
	`  maximum-paths 2`
	The following mentioned below is the running configuration after configuring delete operation:
	`router bgp 100`
	`  vrf abc`
	`  address-family ipv4 unicast`
	`  maximum-paths 2`

`.../set-med`	Make sure that you don't configure BGP actions with `set-med` and OSPF commands with metric in the same route-map through OpenCofig NETCONF. As the OSPF command metrics succeed over BGP `set-med` properties. You must use two different route maps to set metrics in OSPF actions. Use `set-med` in BGP actions using separate route maps. Cisco recommends you not to change the metric of BGP actions to OSPF actions or OSPF actions to BGP actions of a route map in a single payload.
`…/is-reachability`	When you search for IS-IS `is-reachability` with system-id as key, the original DME returns all the entries. But, OpenConfig returns one entry with the highest metric value.

Path	Description
…/bgp/global/config/as	

Path	Description

	You must enter an autonomous system (AS) number to have a valid BGP instance. As there is no default value for an AS number, deleting NETCONF/OPENCONFIG<asn> without removing BGP instance results in the following **highlighted** error message:
	`764`
	`<rpc`
	`xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"`
	`message-id="urn:uuid:1ea09de2-605e-46aa-984b-9dfdad03354d">`
	`<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>`
	`<global>`
	`<config 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>`
	`</config>`
	`</edit-config>`
	`</rpc>`

Path	Description
	## 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>

Path	Description
…/protocol/ospf	

Path	Description

	• If you configure and remove an area configuration in OSPF, the deleted area shows in DME. These areas are shown in `GETCONFIG/GET` output in OpenConfig YANG.
	• You can configure one are in OpenConfig YANG in the OSPF policy `match ospf-area` configuration. Though you can configure multiple areas, such as `match ospf-area 100 101`. But in OpenConfig YANG, you can configure only one area.
	For an example, `match ospf-area 100`.
	• You cant configure both area virtual-link and area interface configurations payload in the same area list. Ensure that you split the area container payload as Virtual ink area and interface area in the same payload.
	• You cannot configure the MD5 authentication string 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 doesn't support an option for password authentication.
	• The OSPF area authentication configuration is not supported. For example, `area 0.0.0.200 authentication message-digest` cannot be configured from OpenConfig YANG.
	• You cannot delete the OSPF or BGP instance configuration, which is in default VRF. For example, **router ospf 1/router bgp 1**, you cannot delete protocols container with the default network instance.
	• When you add an interface through OpenConfig YANG, OSPFv2 can send an error message. If there is an issue, you cannot add an interface, and the RPC reply has list merge failed errors as the following:
	`<rpc-reply`

Path	Description
	xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:39507023-8569-4cf8-869c-e19aaf76a260"> <rpc-error> <error-type>protocol</error-type> <error-tag>operation-failed</error-tag> <error-severity>error</error-severity> <error-message xml:lang="en">List Merge Failed: operation-failed</error-message> <error-path>/network-instances/network-instance[name='default']/protocols/protocol[identifier='BGP'][name='BGP']/bgp/neighbors/neighbor[neighbor-address='10.1.1.1']/afi-safis/afi-safi[afi-safi-name='IPV4_UNICAST']/ipv4-unicast/prefix-limit/config/max-prefixes</error-path> </rpc-error> </rpc-reply>

Table 3: OpenConfig Routing Policy

Path	Description
/bgp-defined-sets/community-sets/community-set/ /bgp-defined-sets/as-path-sets/as-path-set/	Action type is always `permit` for `community-set` and `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`.
/bgp-defined-sets/community-sets/community-set/ /bgp-defined-sets/as-path-sets/as-path-set/	Action type is always `permit` for `community-set` and `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`.

Path	Description
/bgp-defined-sets/community-sets/community-set/	In the CLI, community-list has two different types, such as standard and expanded. In the OpenConfig YANG model, community-set-name has no differentiation. When you create the community-set-name through OpenConfig YANG, the following changes are configured: • The _std suffix is appended after community-set-name if community-member is in the standard form (AS:NN). • The _exp suffix is 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

Path	Description
/bgp-defined-sets/community-sets/community-set/	In the CLI, community-list has two different types, such as standard and expanded. In the OpenConfig YANG model, community-set-name has no differentiation. When you create the community-set-name through OpenConfig YANG, the following changes are configured: • The _std suffix is appended after community-set-name if community-member is in the standard form (AS:NN). • The _exp suffix is 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

Path	Description
/bgp-conditions/match-community-set/config/community-set/	

Path	Description

	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">`

Path	Description
	<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, you can 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

Path	Description

/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. The 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>`
	The 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`

/bgp-actions/set-community/config/method	enum "REFERENCE" is not supported
/bgp-actions/config/set-next-hop	enum "SELF" is not supported
/bgp-conditions/match-community-set/config/community-set	Get mapped only to match community <community-set>_std, only a standard community is supported. Match to an expanded community set is not supported.
…/match-tag-set	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.

Table 4: OpenConfig Interfaces

Path	Description
interfaces/interface/ethernet/switched-vlan/config/interface-mode	

Path	Description

	When you attempt to simultaneously configure a trunk-mode interface and trunk related config in the same OpenConfig payload, the configuration does not complete successfully.
	
	This is because for Cisco NX-OS interfaces, the default interface mode is **access**. To implement any trunk specific configurations, you must first change the interface mode to **trunk**, then only the trunk related configuration could become valid. NX-OS does not support to change them in the same request.
	
	The recommendation is to send the requests in two steps :
	
	• The first step to change the interface to the trunk mode
	
	• The second step to change trunk specific config
	
	1. First step, configure the interface to trunk mode.
	
	`<rpc`
	
	`xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"`
	
	`message-id="101">`
	`<edit-config>`
	`<config>`
	`<interfaces`
	
	`xmlns="http://openconfig.net/yang/interfaces">`
	
	`<interface>`
	`<name>eth1/2</name>`
	`<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>`
	
	2. Second step, configure the trunk specific config. The following show two separate examples for the illustration purpose. If necessary, this step can combine these two examples in the same payload if necessary.

Path	Description
	• Example: change “native-vlan” <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <config> <interfaces xmlns="http://openconfig.net/yang/interfaces"> <interface> <name>eth1/2</name> <ethernet xmlns="http://openconfig.net/yang/interfaces/ethernet"> <switched-vlan xmlns="http://openconfig.net/yang/vlan"> <config> <native-vlan>999</native-vlan> </config> </switched-vlan> </ethernet> </interface> </interfaces> </config> </edit-config> </rpc>

Path	Description

	• Example: change “trunk-vlans"
	Please note that “trunk-vlans” is a leaf-list in which each range is a unique list entry. In Cisco NX-OS, when an interface is first moved to the trunk mode, NX-OS would automatically create a default list entry “1..4094” to allow the whole range for convenience. To customize the ranges, the user needs to remove the default entry, and add the customized non-overlap ranges.
	`<rpc`
	`xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"`
	`message-id="101">`
	`<edit-config>`
	`<config>`
	`<interfaces`
	`xmlns="http://openconfig.net/yang/interfaces">`
	`<interface>`
	`<name>eth1/2</name>`
	`<ethernet`
	`xmlns="http://openconfig.net/yang/interfaces/ethernet">`
	`<switched-vlan`
	`xmlns="http://openconfig.net/yang/vlan">`
	`<config>`
	`<trunk-vlans`
	`<xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"`
	`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>`

Path	Description
interfaces/interface/ethernet/switched-vlan/config/trunk-vlans	Due to restriction of OpenConfig YANG, ensure that configuring VLANs must not overlap between VLANs in the payload and the VLANs configured on an interface. If there is an overlap, 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 and ensure about appropriate about the VLAN range.
	The following for switch port, shut/no shut, MTU, and mac-address: You must reload ASCII, while configuring switch port, shut/no shut, MTU, and mac-address. If you reload a binary result, the configuration is lost.

Table 5: OpenConfig LACP

Path	Description
lacp/interfaces/interface/config/lacp-mode	OC-LACP enables configuring the port-channel mode on the port-channel interface. But, for the NXOS-CLI, the port-channel mode is configured on the member interface using channel-group mode active or passive. Though OC-LACP explicitly configures the port-channel mode on a port-channel interface, issuing the NX-OS **show running-config** command on a port-channel interface does not show the port-channel mode configuration for either empty or nonempty port-channels. When you add a member to the port-channel, **show running interface ethernet** <> shows the port-channel mode configuration as a channel-group mode active or passive. **Note** All port-channels that you create through OpenConfig must continue to be managed by OpenConfig.

Path	Description
lacp/interfaces/interface/config/interval	You can change port channel interval only when members are in `shut` state. The OC-LACP interval is per port-channel. The NX-OS LACP interval is per port-channel member. Due to this difference, the following behavior is expected: • If you configure the port-channel interval through OpenConfig, all members in the port-channel get the same configuration applied to them. If you configure the port-channel interval through OpenConfig and then you add a member. Ensure that you configure the interval through OpenConfig for applying configuration to the new member.
lacp/interfaces/interface/config/system-id-mac	does not support `system-id-mac` per port-channel
LACP	The following has member-state data only when a port is in admin `up` state: • LACP • Interface • Interfaces • Member • State

The following state Containers are implemented for OpenConfig ACL at the interface-ref level.

Table 6: OpenConfig ACL

Path	Description
acl/interfaces/interface/interface-ref/stateforacl/interfaces/state	N/A
acl/interfaces/interface/interface-ref/state/interface	read-only
acl/interfaces/interface/interface-ref/state/subinterface	read-only

Table 7: OpenConfig QoS

Path	Description
QoS	• 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 the sum of the ucast, mcast, and bcast queues per qos-group. • 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 system config containers are implemented for domain-name, login-banner, and motd-banner.

Table 8: OpenConfig System

Path	Description
system/config/domain-name	System/dns-items/ prof-items/Prof-list/dom-items/name
system/config/login-banner	System/userext-items/postloginbanner-items/message
system/config/motd-banner	System/userext-items/preloginbanner-items/message

Guidelines and Limitations for High Scale Data

Cisco NX-OS supports a new set of operational state OpenConfig path which has high scale data. The following are the guidelines and limitations:

- For optimal performance, you must retrieve data by providing the exact path. Parent-level path queries will not provide the same performance.
- The high scale paths are supported only for gNMI and not for RESTCONF or NETCONF.
- The high-scale paths do not support suppress-redundant.
- The high scale paths do not support the gNMI ON_CHANGE subscription.

Path	Description
/network-instances/network-instance/fdb/l2rib/mac-table	Parent level queries for l2rib are supported at l2rib level. For example, you can query till `network-instances/network-instance/fdb/l2rib` but not at fdb level `network-instances/network-instance/fdb`.

Path	Description
/interfaces/interface/routed-vlan/ipv4/neighbors/neighbor/state	
/interfaces/interface/routed-vlan/ipv6/neighbors/neighbor/state	For parent-level queries, the infrastructure retrieves all the keys for the list items and a request is sent 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 has a track of static entries, while the back end has static and dynamic entries. For the list, walk the infrastructure sends requests for each static entry which results 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 provided, it contains 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	

Configuring OpenConfig Support

To enable or disable OpenConfig support on the programmability agents such as (NETCONF, RESTCONF and gRPC).

SUMMARY STEPS

1. **configureterminal**
2. **featureopenconfig**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configureterminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**featureopenconfig** **Example:**	Enters global configuration mode. `switch(config)# feature netconf`

Upgrading or Downgrading YANG Version

YANG models are a collection of logical supported configuration or states. This model doesn't support earlier or future compatibility releases.

Cisco NX-OS doesn't support multiple YANG versions for a single release. Each NX-OS release supports a specific and authorized YANG collection. See Yang NX Repository, for respective NX-OS releases.

To upgrade or downgrade NX-OS releases, there exist few modifications in the supported YANG models. You must review the YANG models and perform appropriate actions.

Changes	Recommendation
Paths added	You can evaluate the way to handle the additional YANG paths
Paths modified	This is due to the property type changes, from integer to float. You must update
Paths removed	In OpenConfig, the community decides to remove paths which are not applica if you use the corresponding functionality, access through other interfaces, lik

RBAC for YANG

Cisco NX-OS supports write permission to YANG paths for non-admin user roles.

Guidelines and Limitations

The following are the guidelines and limitations for RBAC of YANG:

- A user with role network-admin has write access. This user can provide other edit permission to nonadmin groups.
- User role network-admin only has the read access.
- RBAC doesn't support providing root system access to nonadmin user roles.

- RBAC doesn't support providing access to paths *System/yangrbacdb-items* or *System/rbacdb-items* to nonadmin user roles. These user roles are restricted for modifying the access for their roles.
- RBAC Honors is supported in running configuration, not for candidate. Candidate configuration is supported in NETCONF and not in RESTCONF or gNMI. Cisco recommends configuring RBAC on the running configuration in a single transaction. If you want to change RBAC through candidate configuration, it is recommended to change and commit RBAC itself. Do not change RBAC configuration and regular configuration simultaneously, in such cases RBAC changes in the candidate configuration will not change.
- Make sure that you provide access to the last element in the path and its subtree. For an example, for path "x/y/z", to provide access to the user to modify the last path element and the child tree.
- If you want to provide write access to specific parent or child nodes in the same path, there is no syntax to merge rules in a single rule. You must configure it separately.

  For example, the following mentioned below rule provides access to priv-9, to change description for all interfaces. But, it can change two properties of a particular SVI interface. These rules cannot be merged.

  - interfaces/interface/description (priv-9)
  - interfaces/interface[id=vlan100]/type (priv-9)
  - interfaces/interface[id=vlan100]/enabled (priv-9)

- RBAC supports a maximum of 512 rules.
- RBAC user role supports forward reference.
  - The user role can specify a role which doesn't exist.
  - The user role can specify a yang path which doesn't exist.
- Make sure that you check the following syntax:
  - Don't add specific wildcard characters.
  - Make sure that you add the namespace string at the first element only.
  - Don't add leading '/' in the path.
  - Path with string only 'system' is not allowed.
- RBAC has no impact on the existing limitations for NETCONF, none operation.

Configuring YANG RBAC

To configure yang path to a specific user role.

SUMMARY STEPS

1. **cofnigureterminal**
2. **rbacyang-rule***yang-pathrole*
3. **allow-writes**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**cofnigureterminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

Step 2	**rbacyang-rule***yang-pathrole*
	Example:
	`switch(config)# rbac yang-rule System/intf-items/aggr-items/AggrIf-list[id=po101] network-operator`
	Example:
	`switch(config)# rbac yang-rule openconfig-interface:interfaces/interface[name=vlan100]/desc priv-9`
		Configures yang path to the user role. The path can be either a Device or OpenConfig YANG path.

Step 3	**allow-writes** **Example:** `switch(config)# allow-writes`	Enables the write permission.

Troubleshooting YANG

You can use the following commands to verify YANG configurations. Specifically to verify the enablement of “feature OpenConfig”.

For issues related to read or write operations for specific agents, see troubleshooting guide for respective agents.

Command	Description
`show running-config openconfig`	To check whether OpenConfig is enabled.
`show openconfig nxsdk event-history { event | errors }`	To check the debug of the OpenConfig feature
`show telemetry yang direct-path cisco-nxos-device`	To display the paths which are supported.