Supported Yang Models

Supported Yang Models

The following is the list of supported config, oper and act YANG models for NCS 1020:

Config Models	Oper Models	Action Models
Cisco-IOS-XR-osa-linesystem-cfg.yang	Cisco-IOS-XR-osa-hwmod-linesys-oper.yang	Cisco-IOS-XR-install-act.yang
Cisco-IOS-XR-controller-ots-cfg.yang	Cisco-IOS-XR-controller-ots-oper.yang	Cisco-IOS-XR-upgrade-fpd-ng-act.yang
Cisco-IOS-XR-ots-och-cfg.yang	Cisco-IOS-XR-controller-ots-och-oper.yang	Cisco-IOS-XR-system-reboot-act.yang
Cisco-IOS-XR-controller-oms-cfg	Cisco-IOS-XR-controller-oms-oper.yang	Cisco-IOS-XR-pmengine-clear-act.yang
Cisco-IOS-XR-controller-och-cfg	Cisco-IOS-XR-controller-och-oper.yang	Cisco-IOS-XR-olc-act.yang

Config Models	Oper Models	Action Models
Cisco-IOS-XR-controller-osc-cfg.yang	Cisco-IOS-XR-controller-osc-oper.yang	Cisco-IOS-XR-controller-ots-otdr-act.yang
Cisco-IOS-XR-controller-dfb-cfg.yang	Cisco-IOS-XR-controller-dfb-oper.yang	Cisco-IOS-XR-controller-ots-tone-pattern-act.yang
Cisco-IOS-XR-pmengine-cfg.yang	Cisco-IOS-XR-pmengine-oper.yang	Cisco-IOS-XR-controller-ots-tone-pattern-detect-act.yang
Cisco-IOS-XR-olc-cfg.yang	Cisco-IOS-XR-olc-oper.yang	
Cisco-IOS-XR-fpd-infra-cfg	Cisco-IOS-XR-show-fpd-loc-ng-oper	
Cisco-IOS-XR-osa-ct-cfg	Cisco-IOS-XR-alarmgr-server-oper.yang	
	Cisco-IOS-XR-platform-oper	

The supported Open Config model is: openconfig-optical-amplifier

Structure of Yang Models

YANG data models can be represented in a hierarchical, tree-based structure with nodes, which makes them more easily understandable. YANG defines four nodes types. Each node has a name, and depending on the node type, the node might either define a value or contain a set of child nodes. The nodes types (for data modeling) are:

- leaf node—Contains a single value of a specific type
- list node—Contains a sequence of list entries, each of which is uniquely identified by one or more keys leafs
- leaf-list node—Contains a sequence of leaf nodes
- container node—Contains a grouping of related nodes containing only child nodes, which can be any of the four node types

The following is the tree structure of the openconfig-optical-amplifier model.

Note Cisco NCS 1020 supports only the leaves that are highlighted as bold in the following open configuration models.

```
 +--rw optical-amplifier
+--rw amplifiers
| +--rw amplifier* [name]
| +--rw name -> ../config/name
| +--rw config
| | +--rw name? string
| | +--rw type? identityref
| | +--rw target-gain? decimal64
| | +--rw min-gain? decimal64
| | +--rw max-gain? decimal64
| | +--rw target-gain-tilt? decimal64
```

```
| | +--rw gain-range? identityref
| | +--rw amp-mode? identityref
| | +--rw target-output-power? decimal64
| | +--rw max-output-power? decimal64
| | +--rw enabled? boolean
| | +--rw fiber-type-profile? identityref
| +--ro state
| +--ro name? string
| +--ro type? identityref
| +--ro target-gain? decimal64
| +--ro min-gain? decimal64
| +--ro max-gain? decimal64
| +--ro target-gain-tilt? decimal64
| +--ro gain-range? identityref
| +--ro amp-mode? identityref
| +--ro target-output-power? decimal64
| +--ro max-output-power? decimal64
| +--ro enabled? boolean
| +--ro fiber-type-profile? identityref
| +--ro component? -> /oc-platform:components/component/name
| +--ro ingress-port? -> /oc-platform:components/component/name
| +--ro egress-port? -> /oc-platform:components/component/name
| +--ro actual-gain
| +--ro actual-gain-tilt
| +--ro input-power-total
| +--ro input-power-c-band
| +--ro input-power-l-band
| +--ro output-power-total
| +--ro output-power-c-band
| +--ro output-power-l-band
| +--ro laser-bias-current
| +--ro optical-return-loss
+--rw supervisory-channels
+--rw supervisory-channel* [interface]
+--rw interface -> ../config/interface
```

The following is a sample tree structure of Cisco-IOS-XR-controller-ots-oper model.

```
+--ro ots-oper
       +--ro ots-ports
          +--ro ots-port* [name]
             +--ro ots-info
             |  +--ro raman-tx-power
             |  |  +--ro raman-tx-power*
             |  |     +--ro raman-tx-power-instance?   uint32
             |  |     +--ro raman-tx-power-value?      uint32
             |  |     +--ro raman-tx-wavelength?       uint32
             |  +--ro transmit-n-power
             |  |  +--ro transmit-power*
             |  |     +--ro instance?   uint32
             |  |     +--ro value?      int32
             |  +--ro receive-n-power
             |  |  +--ro receive-power*
             |  |     +--ro instance?   uint32
             |  |     +--ro value?      int32
             |  +--ro ingress-channel-slice-attenuation
             |  |  +--ro ingress-channel-slice*
             |  |     +--ro ingress-channel-slice?               uint32
             |  |     +--ro ingress-channel-slice-attenuation?   int32
             |  +--ro egress-channel-slice-attenuation
             |  |  +--ro egress-channel-slice*
             |  |     +--ro egress-channel-slice?               uint32
             |  |     +--ro egress-channel-slice-attenuation?   int32
             |  +--ro raman-tx-power-config
```

```
|  |  +--ro raman-tx-power*
|  |     +--ro raman-tx-power-instance?   uint32
|  |     +--ro raman-tx-power-value?      uint32
|  +--ro ingress-channel-slice-attenuation-configured
|  |  +--ro ingress-channel-slice*
|  |     +--ro ingress-channel-slice?               uint32
|  |     +--ro ingress-channel-slice-attenuation?   int32
|  +--ro egress-channel-slice-attenuation-configured
|  |  +--ro egress-channel-slice*
|  |     +--ro egress-channel-slice?               uint32
|  |     +--ro egress-channel-slice-attenuation?   int32
|  +--ro channel-attenuation-info
|  |  +--ro total-channel-attenuation-slice-count?        uint32
|  |  +--ro channel-attenuation-slice-spacing?            uint32
|  |  +--ro channel-attenuation-first-slice-wavelength?   uint32
|  |  +--ro channel-attenuation-first-slice-frequency?    uint32
|  |  +--ro ingress-channel-attenuation-info*
|  |  |  +--ro slice-num?             uint32
|  |  |  +--ro ingress-attenuation?   uint32
|  |  +--ro egress-channel-attenuation-info*
|  |     +--ro slice-num?            uint32
|  |     +--ro egress-attenuation?   uint32
|  +--ro otdr-info-rx
|  |  +--ro scan-status?       Otdr-scan-status
|  |  +--ro tracepoint-file?   string
|  |  +--ro total-events?      uint32
|  |  +--ro scan-timestamp?    string
|  |  +--ro event-info*
|  |     +--ro event-number?     uint32
|  |     +--ro detected-event?   uint32
|  |     +--ro location?         int64
|  |     +--ro accuracy?         int64
|  |     +--ro magnitude?        int64
|  |     +--ro attenuation?      int64
|  +--ro otdr-info-tx
|  |  +--ro scan-status?       Otdr-scan-status
|  |  +--ro tracepoint-file?   string
|  |  +--ro total-events?      uint32
|  |  +--ro scan-timestamp?    string
|  |  +--ro event-info*
|  |     +--ro event-number?     uint32
|  |     +--ro detected-event?   uint32
|  |     +--ro location?         int64
|  |     +--ro accuracy?         int64
|  |     +--ro magnitude?        int64
|  |     +--ro attenuation?      int64
|  +--ro rx-los-p
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro rx-loc
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro tx-power-fail-low
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro ingress-auto-laser-shut
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro ingress-auto-pow-red
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro ingress-ampli-gain-low
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
```

```
|  +--ro ingress-ampli-gain-high
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro egress-auto-laser-shut
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro egress-auto-pow-red
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro egress-ampli-gain-low
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro egress-ampli-gain-high
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro high-tx-br-pwr
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro high-rx-br-pwr
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro span-too-short-tx
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro span-too-short-rx
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro raman-auto-pow-red
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro raman1-low-pwr
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro raman2-low-pwr
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro raman3-low-pwr
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro raman4-low-pwr
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro raman5-low-pwr
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro raman1-high-pwr
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro raman2-high-pwr
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro raman3-high-pwr
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro raman4-high-pwr
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro raman5-high-pwr
|  |  +--ro is-detected?   boolean
|  |  +--ro counter?       uint32
|  +--ro ots-och-alamr-info
|  |  +--ro rx-los-p
|  |  |  +--ro is-detected?   boolean
|  |  |  +--ro counter?       uint32
```

```
                |  |  +--ro tx-power-fail-low
                |  |     +--ro is-detected?   boolean
                |  |     +--ro counter?       uint32
                |  +--ro ots-tone-info
                |  |  +--ro tone-freq?          string
                |  |  +--ro tone-rate?          uint32
                |  |  +--ro pattern?            string
                |  |  +--ro pattern-expected?   string
                |  |  +--ro dectected-oob?      uint32
                |  |  +--ro state?              Conn-verfcn-state
                |  |  +--ro pattern-received?   string
                |  +--ro transport-admin-state?                          Ots-tas
                |  +--ro rx-pow-low-threshold?                           int32
                |  +--ro rx-pow-high-threshold?                          int32
                |  +--ro tx-pow-low-threshold?                           int32
                |  +--ro tx-pow-high-threshold?                          int32
                |  +--ro pm-enable?                                      uint32
                |  +--ro controller-state?                               Ots-controller-state
                |  +--ro rx-voa-attenuation?                             int32
                |  +--ro tx-voa-attenuation?                             int32
                |  +--ro channel-width?                                  uint32
                |  +--ro central-frequncy?                               uint32
                |  +--ro add-drop-channel?                               string
                |  +--ro line-channel?                                   string
                |  +--ro ingress-ampli-gain?                             int32
                |  +--ro ingress-ampli-tilt?                             int32
                |  +--ro ingress-amp-gain-deg-thres-low?                 uint32
                |  +--ro ingress-amp-gain-deg-thres-high?                uint32
                |  +--ro ingress-ampli-gain-range?                     Ots-amplifier-gain-range

                |  +--ro egress-ampli-gain?                              int32
                |  +--ro egress-ampli-tilt?                              int32
                |  +--ro egress-amp-gain-deg-thres-low?                  uint32
                |  +--ro egress-amp-gain-deg-thres-high?                 uint32
                |  +--ro egress-ampli-gain-range?                      Ots-amplifier-gain-range

                |  +--ro composite-raman-power?                          uint32
                |  +--ro wavelength?                                     uint32
                |  +--ro transmit-power?                                 int32
                |  +--ro receive-power?                                  int32
                |  +--ro total-cl-tx-power?                              int32
                |  +--ro total-cl-rx-power?                              int32
                |  +--ro receive-signal-power?                           int32
                |  +--ro transmit-signal-power?                          int32
                |  +--ro ingress-ampli-osri?                             boolean
                |  +--ro egress-ampli-osri?                              boolean
                |  +--ro ingress-ampli-force-apr?                        boolean
                |  +--ro egress-ampli-force-apr?                         boolean
                |  +--ro ingress-ampli-safety-control-mode?
Ots-amplifier-safety-control-mode
                |  +--ro egress-ampli-safety-control-mode?
Ots-amplifier-safety-control-mode
                |  +--ro ingress-ampli-safety-control-mode-configured?
Ots-amplifier-safety-control-mode
                |  +--ro egress-ampli-safety-control-mode-configured?
Ots-amplifier-safety-control-mode
                |  +--ro ingress-ampli-osri-configured?                  boolean
                |  +--ro egress-ampli-osri-configured?                   boolean
                |  +--ro ingress-ampli-force-apr-configured?             boolean
                |  +--ro egress-ampli-force-apr-configured?              boolean
                |  +--ro raman-safety-control-mode?
Ots-amplifier-safety-control-mode
                |  +--ro raman-safety-control-mode-configured?
Ots-amplifier-safety-control-mode
```

```
              |  +--ro raman-osri?                                    boolean
              |  +--ro raman-force-apr?                               boolean
              |  +--ro raman-osri-configured?                         boolean
              |  +--ro raman-force-apr-configured?                    boolean
              |  +--ro rx-pow-low-warning-threshold?                  int32
              |  +--ro rx-pow-high-warning-threshold?                 int32
              |  +--ro tx-pow-low-warning-threshold?                  int32
              |  +--ro tx-pow-high-warning-threshold?                 int32
              |  +--ro description?                                   string
              |  +--ro channel-attenuation?                           int32
              |  +--ro rx-voa-attenuation-config-val?                 int32
              |  +--ro tx-voa-attenuation-config-val?                 int32
              |  +--ro ampli-control-mode-config-val?
Ots-amplifier-control-mode
              |  +--ro rx-low-th-psd-config-val?                      int32
              |  +--ro total-rx-power?                                int32
              |  +--ro total-tx-power?                                int32
            |  +--ro ingress-ampli-gain-range-config-val?          Ots-amplifier-gain-range

              |  +--ro ingress-ampli-gain-config?                     uint32
              |  +--ro ingress-ampli-tilt-config?                     int32
              |  +--ro ingress-ampli-thr-deg-low-config?              uint32
              |  +--ro ingress-ampli-thr-deg-high-config?             uint32
            |  +--ro egress-ampli-gain-range-config-val?           Ots-amplifier-gain-range

              |  +--ro egress-ampli-gain-config?                      uint32
              |  +--ro egress-ampli-tilt-config?                      int32
              |  +--ro egress-ampli-gain-thr-deg-low-config?          uint32
              |  +--ro egress-ampli-gain-thr-deg-high-config?         uint32
              |  +--ro channel-attenuation-configured?                int32
              |  +--ro br-power?                                      int32
              |  +--ro raman-br-power?                                int32
              |  +--ro led-state?                                     Led-state
              +--ro ots-spectrum-info
              |  +--ro spectrum-info
              |     +--ro total-spectrum-slice-count?   uint32
              |     +--ro spectrum-slice-spacing?       uint32
              |     +--ro first-slice-wavelength?       uint32
              |     +--ro first-slice-frequency?        uint32
              |     +--ro spectrum-slice-power-info*
              |        +--ro slice-num?   uint32
              |        +--ro rx-power?    int16
              |        +--ro tx-power?    int16
              +--ro name                xr:Interface-name
```

The following is a sample tree structure of Cisco-IOS-XR-controller-ots-cfg model.

```
augment /a1:interface-configurations/a1:interface-configuration:
    +--rw ots
       +--rw ingress-channel-slice-attns
       |  +--rw ingress-channel-slice-attn* [ingress-channel-slice-attn]
       |     +--rw ingress-channel-slice-attn         uint32
       |     +--rw ingress-channel-slice-attnvalue    uint32
       +--rw raman-tx-power-disables
       |  +--rw raman-tx-power-disable* [raman-tx-power-disable-instance]
       |     +--rw raman-tx-power-disable-instance    uint32
       +--rw raman-tx-powers
       |  +--rw raman-tx-power* [raman-tx-power-instance]
       |     +--rw raman-tx-power-instance    uint32
       |     +--rw raman-tx-power-value       uint32
       +--rw ots-otdr
       |  +--rw ots-otdr-rx
       |  |  +--rw ots-otdr-rx-expert
       |  |  |  +--rw ots-otdr-rx-capture-start?   uint32
       |  |  |  +--rw ots-otdr-rx-scan-duration?   uint32
```

```
|  |  |  +--rw ots-otdr-rx-pulse-width?     uint32
|  |  |  +--rw ots-otdr-rx-capture-end?     uint32
|  |  +--rw ots-otdr-rx-auto
|  |  |  +--rw ots-otdr-rx-excess-reflection-threshold?   int32
|  |  |  +--rw ots-otdr-rx-splice-loss-threshold?         uint32
|  |  |  +--rw ots-otdr-rx-raman-setpoint?                uint32
|  |  |  +--rw ots-otdr-rx-reflectance-threshold?         int32
|  |  +--rw ots-otdr-rx-back-scattering?    int32
|  |  +--rw ots-otdr-rx-refractive-index?   uint32
|  +--rw ots-otdr-scan-mode
|  |  +--rw ots-otdr-scan-mode-expert?   empty
|  +--rw ots-otdr-tx
|     +--rw ots-otdr-tx-expert
|     |  +--rw ots-otdr-tx-capture-end?     uint32
|     |  +--rw ots-otdr-tx-scan-duration?   uint32
|     |  +--rw ots-otdr-tx-capture-start?   uint32
|     |  +--rw ots-otdr-tx-pulse-width?     uint32
|     +--rw ots-otdr-tx-auto
|     |  +--rw ots-otdr-tx-splice-loss-threshold?         uint32
|     |  +--rw ots-otdr-tx-excess-reflection-threshold?   int32
|     |  +--rw ots-otdr-tx-raman-setpoint?                uint32
|     |  +--rw ots-otdr-tx-reflectance-threshold?         int32
|     +--rw ots-otdr-tx-refractive-index?   uint32
|     +--rw ots-otdr-tx-back-scattering?    int32
+--rw egress-channel-slice-attns
|  +--rw egress-channel-slice-attn* [egress-channel-slice-attn]
|     +--rw egress-channel-slice-attn         uint32
|     +--rw egress-channel-slice-attnvalue    uint32
+--rw ots-egress-safety-control-mode?                      Ots-safety-control-mode
+--rw ots-ingress-amplifier-gain?                          uint32
+--rw ots-tone-pattern-expected?                           string
+--rw ots-ingress-osri?                                    boolean
+--rw ots-ingress-amplifier-gain-degrade-high-threshold?   uint32
+--rw ots-tx-voa-attenuation?                              uint32
+--rw ots-ingress-safety-control-mode?                     Ots-safety-control-mode
+--rw ots-tone-detect-oob?                                 empty
+--rw ots-ingress-force-apr?                               boolean
+--rw ots-raman-force-apr?                                 boolean
+--rw ots-egress-amplifier-gain-degrade-low-threshold?     uint32
+--rw ots-ingress-amplifier-gain-degrade-low-threshold?    uint32
+--rw ots-egress-amplifier-tilt?                           int32
+--rw ots-raman-safety-control-mode?                       Ots-safety-control-mode
+--rw ots-tone-frequency?                                  string
+--rw ots-egress-amplifier-gain?                           uint32
+--rw ots-tone-pattern?                                    string
+--rw ots-egress-amplifier-gain-degrade-high-threshold?    uint32
+--rw ots-raman-osri?                                      boolean
+--rw ots-egress-osri?                                     boolean
+--rw ots-egress-amplifier-gain-range?
Ots-ingress-egress-ampli-gain-range
+--rw ots-ingress-amplifier-gain-range?
Ots-ingress-egress-ampli-gain-range
+--rw ots-ingress-amplifier-tilt?                          int32
+--rw ots-tone-rate?                                       uint32
+--rw ots-egress-force-apr?                                boolean
```

Configure Flex Grid in OLT Card

Step 1 Use the Cisco-IOS-XR-osa-linesystem-cfg.yang Yang model to configure flex grid channel in the OLT card.

Yang Model	Example
Cisco-IOS-XR-osa-linesystem-cfg.yang	<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <active-nodes xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-config-mda-cfg"> <active-node> <node-name>0/0/NXR0</node-name> <terminal-amplifier xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-osa-linesystem-cfg"> <olt-grid-mode> <olt-channel-identifier-tables> <olt-channel-identifier-table> <channel-number>1</channel-number> <olt-channel-definition> <centre-frequency>191.425</centre-frequency> <channel-width>150</channel-width> </olt-channel-definition> </olt-channel-identifier-table> </olt-channel-identifier-tables> </olt-grid-mode> </terminal-amplifier> </active-node> </active-nodes> </config> </edit-config> </rpc>

Step 2 Use the Cisco-IOS-XR-osa-hwmod-linesys-oper.yang Yang model to get the operational data of the flex grid channel configured on the OLT card.

Yang Model	Example
Cisco-IOS-XR-osa-hwmod-linesys-oper.yang	<?xml version="1.0"?> <rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <data> <osa xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-osa-hwmod-linesys-oper"> <node-ids> <node-id> <node-name>0/0/NXR0</node-name> <terminal-ampli> <flexi-grid-info> <channel-number>1</channel-number> <centre-frequency-thz>191.425000</centre-frequency-thz> <channel-width-ghz>150.000</channel-width-ghz> <channel-status>active</channel-status> <overlapping-channel-info> <left-overlapping-channel>-</left-overlapping-channel> <right-overlapping-channel>-</right-overlapping-channel> </overlapping-channel-info> </flexi-grid-info> </terminal-ampli> </node-id> </node-ids> </osa> </data> </rpc-reply>

Configure Flex Grid in ILA Card

Step 1 Use the Cisco-IOS-XR-osa-linesystem-cfg.yang Yang model to configure the flex grid channel in the ILA card.

Yang Model	Example
Cisco-IOS-XR-osa-linesystem-cfg.yang	<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <active-nodes xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-config-mda-cfg"> <active-node> <node-name>0/0/NXR0</node-name> <inline-amplifier xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-osa-linesystem-cfg"> <ila-grid-mode> <ila-channel-identifiers> <ila-channel-identifier> <channel-number>1</channel-number> <centre-frequency>191.375</centre-frequency> <channel-width>75</channel-width> </ila-channel-identifier> </ila-channel-identifiers> </ila-grid-mode> </inline-amplifier> </active-node> </active-nodes> </config> </edit-config> </rpc>

Step 2 Use the Cisco-IOS-XR-osa-hwmod-linesys-oper.yang Yang model to get the operational data for the flex grid channel configured on the ILA card.

Yang Model	Example
Cisco-IOS-XR-osa-hwmod-linesys-oper.yang	<?xml version="1.0"?> <rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <data> <osa xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-osa-hwmod-linesys-oper"> <node-ids> <node-id> <node-name>0/0/NXR0</node-name> <inline-ampli> <flexi-grid-info> <channel-number>1</channel-number> <centre-frequency-thz>191.375000</centre-frequency-thz> <channel-width-ghz>75.000</channel-width-ghz> <overlapping-channel-info> <left-overlapping-channel>-</left-overlapping-channel> <right-overlapping-channel>-</right-overlapping-channel> </overlapping-channel-info> </flexi-grid-info> </inline-ampli> </node-id> </node-ids> </osa> </data> </rpc-reply>

Configure OTS Controller

Step 1 Use the Cisco-IOS-XR-controller-ots-cfg.yang Yang model to configure the OTS controller parameters.

Yang Model	Example
Cisco-IOS-XR-controller-ots-cfg.yang	<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg"> <interface-configuration> <active>act</active> <interface-name>Ots0/0/0/0</interface-name> <ots xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-controller-ots-cfg"> <ots-egress-safety-control-mode>auto</ots-egress-safety-control-mode> <ots-ingress-amplifier-gain>160</ots-ingress-amplifier-gain> <ots-ingress-osri>true</ots-ingress-osri> <ots-tx-voa-attenuation>200</ots-tx-voa-attenuation> <ots-ingress-force-apr>false</ots-ingress-force-apr> <ots-egress-amplifier-tilt>-40</ots-egress-amplifier-tilt> <ots-egress-amplifier-gain>180</ots-egress-amplifier-gain> <ots-egress-osri>false</ots-egress-osri> <ots-ingress-amplifier-gain-range>normal</ots-ingress-amplifier-gain-range> <ots-ingress-amplifier-tilt>50</ots-ingress-amplifier-tilt> <ots-egress-force-apr>true</ots-egress-force-apr> </ots> </interface-configuration> </interface-configurations> </config> </edit-config> </rpc>

Step 2 Use the Cisco-IOS-XR-controller-ots-oper.yang Yang model to view the parameters of the OTS controller.

Note In the current release, all the controller models are mapped to the OTS controller model. Hence the operational data of all the controllers display "ots-state-up" as the controller state, and "ots-tas-ui-is" as transport-admin-state, irrespective of the functionality.

Yang Model	Example
Cisco-IOS-XR-controller-ots-oper.yang	

Yang Model	Example
	<?xml version="1.0" ?> <rpc-reply message-id="urn:uuid:1ecef265-e94d-4b42-ad53-adb137a58efc" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"> <data> <ots-oper xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-controller-ots-oper"> <ots-ports> <ots-port> <name>Ots0/0/0/0</name> <ots-info> <transport-admin-state>ots-tas-ui-is</transport-admin-state> <controller-state>ots-state-up</controller-state> <tx-voa-attenuation>200</tx-voa-attenuation> <ingress-ampli-gain>160</ingress-ampli-gain> <ingress-ampli-tilt>50</ingress-ampli-tilt> <ingress-ampli-gain-range>ots-amplifier-gain-range-normal</ingress-ampli-gain-range> <egress-ampli-gain>180</egress-ampli-gain> <egress-ampli-tilt>-40</egress-ampli-tilt> <total-cl-tx-power>2000</total-cl-tx-power> <total-cl-rx-power>-1000</total-cl-rx-power> <receive-signal-power>2000</receive-signal-power> <transmit-signal-power>2000</transmit-signal-power> <ingress-ampli-osri>true</ingress-ampli-osri> <egress-ampli-osri>false</egress-ampli-osri> .... <tx-power>-105</tx-power> </spectrum-slice-power-info> <spectrum-slice-power-info> <slice-num>1546</slice-num> <rx-power>-105</rx-power> <tx-power>-105</tx-power> </spectrum-slice-power-info> <spectrum-slice-power-info> <slice-num>1547</slice-num> <rx-power>-105</rx-power> <tx-power>-105</tx-power> </spectrum-slice-power-info> <spectrum-slice-power-info> <slice-num>1548</slice-num> <rx-power>-105</rx-power> <tx-power>-105</tx-power> </spectrum-slice-power-info> </spectrum-info> </ots-spectrum-info> </ots-port> </ots-ports> </ots-oper>

Yang Model	Example
	</data> </rpc-reply>

Configure OCH Controller

Step 1 Use the Cisco-IOS-XR-controller-och-cfg.yang Yang model to configure the OCH controller parameters.

Yang Model	Example
Cisco-IOS-XR-controller-och-cfg.yang	<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg"> <interface-configuration> <active>act</active> <interface-name>Och0/3/0/31</interface-name> <och xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-controller-och-cfg"> <och-tone-pattern-expected>1234abcd</och-tone-pattern-expected> <och-tone-rate>20</och-tone-rate> </och> </interface-configuration> </interface-configurations> </config> </edit-config> </rpc>

Step 2 Use Cisco-IOS-XR-controller-och-oper.yang Yang model to view the OCH controller parameters.

Yang Model	Example
Cisco-IOS-XR-controller-och-oper.yang	<?xml version="1.0" ?> <rpc-reply message-id="urn:uuid:50be1a71-e729-442d-aec7-14f486cd6028" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"> <data> <och-oper xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-controller-och-oper"> <och-ports> <och-port> <name>Och0/3/0/31</name> <och-info> <rx-power>0</rx-power> <tx-power>-5000</tx-power> <channel-frequency>191375</channel-frequency> <channel-width>1500</channel-width> <channel-wavelength>156652</channel-wavelength> <controller-state>ots-state-up</controller-state> <led-state>off</led-state> <rx-los-p> <is-detected>false</is-detected> <counter>0</counter> </rx-los-p> <tx-power-fail-low> <is-detected>false</is-detected> <counter>0</counter> </tx-power-fail-low> <och-tone-info> <tone-rate>20</tone-rate> <pattern-expected>1234abcd</pattern-expected> <dectected-oob>0</dectected-oob> <state>conn-vrfcn-state-not-running</state> </och-tone-info> <transport-admin-state>ots-tas-ui-is</transport-admin-state> </och-info> </och-port> </och-ports> </och-oper> </data> </rpc-reply>

Configure Optical Cross-Connect

Step 1 Use the Cisco-IOS-XR-Ots-Och-cfg.yang Yang model to configure an optical cross-connect (OTS-OCH controller).

Yang Model	Example
Cisco-IOS-XR-Ots-Och-cfg.yang	<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg"> <interface-configuration> <active>act</active> <interface-name>Ots-Och0/0/0/0/1</interface-name> <ots-och xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-Ots-Och-cfg"> <add-drop-channel>Ots-Och0/0/0/2/1</add-drop-channel> </ots-och> </interface-configuration> </interface-configurations> </config> </edit-config>

Step 2 Use the Cisco-IOS-XR-controller-ots-och-oper.yang Yang model to view the parameters of the OTS-OCH controller.

Yang Model	Example
Cisco-IOS-XR-controller-ots-och-oper.yang	<?xml version="1.0" ?> <rpc-reply message-id="urn:uuid:71601b7f-caee-4e65-9627-b5043e66436d" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"> <data> <ots-och-oper xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-controller-ots-och-oper"> <ots-och-ports> <ots-och-port> <name>Ots-Och0/0/0/0/1</name> <ots-och-info> <transport-admin-state>ots-tas-ui-is</transport-admin-state> <controller-state>ots-state-up</controller-state> <add-drop-channel>Ots-Och0/0/0/2/1</add-drop-channel> <total-rx-power>-1050</total-rx-power> <total-tx-power>-1050</total-tx-power> </ots-och-info> </ots-och-port> </ots-och-ports> </ots-och-oper> </data> </rpc-reply>

Configure OMS Controller

Step 1 Use the Cisco-IOS-XR-controller-oms-cfg.yang Yang model to configure the OMS controller parameters.

Yang Model	Example
Cisco-IOS-XR-controller-oms-cfg.yang	<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg"> <interface-configuration> <active>act</active> <interface-name>Oms0/3/0/32</interface-name> <oms xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-controller-oms-cfg"> <oms-tone-rate>20</oms-tone-rate> <oms-tone-pattern-expected>abcd1234</oms-tone-pattern-expected> <oms-tone-detect-oob/> </oms> </interface-configuration> </interface-configurations> </config> </edit-config> </rpc>

Step 2 Use the Cisco-IOS-XR-controller-oms-oper.yang Yang model to view the parameters of the OMS controller.

Yang Model	Example
Cisco-IOS-XR-controller-oms-oper	<?xml version="1.0" ?> <rpc-reply message-id="urn:uuid:ba7b0faf-3762-4a8e-b9fe-e8d190a2dbe7" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"> <data> <oms-oper xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-controller-oms-oper"> <oms-ports> <oms-port> <name>Oms0/3/0/32</name> <oms-info> <rx-power>0</rx-power> <tx-power>0</tx-power> <controller-state>ots-state-up</controller-state> <led-state>off</led-state> <rx-los-p> <is-detected>false</is-detected> <counter>0</counter> </rx-los-p> <tx-power-fail-low> <is-detected>false</is-detected> <counter>0</counter> </tx-power-fail-low> <oms-tone-info> <tone-rate>20</tone-rate> <pattern-expected>abcd1234</pattern-expected> <dectected-oob>1</dectected-oob> <state>conn-vrfcn-state-not-running</state> </oms-tone-info> <transport-admin-state>ots-tas-ui-is</transport-admin-state> </oms-info> </oms-port> </oms-ports> </oms-oper> </data> </rpc-reply>

Configure DFB Controller

Step 1 Use the Cisco-IOS-XR-controller-dfb-cfg.yang Yang model to configure the DFB controller parameters.

Yang Model	Example
Cisco-IOS-XR-controller-dfb-cfg.yang	<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg"> <interface-configuration> <active>act</active> <interface-name>Dfb0/0/0/0</interface-name> <dfb xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-controller-dfb-cfg"> <dfb-tx-voa-attenuation>150</dfb-tx-voa-attenuation> </dfb> </interface-configuration> </interface-configurations> </config> </edit-config> </rpc>

Step 2 Use the Cisco-IOS-XR-controller-dfb-oper.yang Yang model to view the DFB controller parameters.

Yang Model	Example
Cisco-IOS-XR-controller-dfb-oper.yang	<?xml version="1.0" ?> <rpc-reply message-id="urn:uuid:41205dcf-f92f-4b73-bdf3-ba64438d15ac" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"> <data> <dfb-oper xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-controller-dfb-oper"> <dfb-ports> <dfb-port> <name>Dfb0/0/0/0</name> <dfb-info> <laser-state>on</laser-state> <controller-state>ots-state-up</controller-state> <transport-admin-state>ots-tas-ui-is</transport-admin-state> <total-rx-power>1000</total-rx-power> <total-tx-power>2000</total-tx-power> <tx-voa-attenuation>150</tx-voa-attenuation> <tx-voa-attenuation-config-val>150</tx-voa-attenuation-config-val> <rx-los-p> <is-detected>false</is-detected> <counter>0</counter> </rx-los-p> <tx-power-fail-low> <is-detected>false</is-detected> <counter>0</counter> </tx-power-fail-low> </dfb-info> </dfb-port> </dfb-ports> </dfb-oper> </data> </rpc-reply>

Configure OSC Controller

Step 1 Use the Cisco-IOS-XR-controller-osc-cfg.yang Yang model to configure the OSC controller parameters.

Yang Model	Example
Cisco-IOS-XR-controller-osc-cfg.yang	<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg"> <interface-configuration> <active>act</active> <interface-name>Osc0/0/0/0</interface-name> <osc xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-controller-osc-cfg"> <osc-transmit-power>20</osc-transmit-power> <osc-transmit-shutdown>false</osc-transmit-shutdown> </osc> </interface-configuration> </interface-configurations> </config> </edit-config> </rpc>

Step 2 Use Cisco-IOS-XR-controller-osc-oper.yang Yang model to view the OSC controller parameters.

Yang Model	Example
Cisco-IOS-XR-controller-osc-oper.yang	<?xml version="1.0" ?> <rpc-reply message-id="urn:uuid:57794a6c-fe5b-425e-8df7-7c09a789b757" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"> <data> <osc-oper xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-controller-osc-oper"> <osc-ports> <osc-port> <name>Osc0/0/0/0</name> <osc-info> <laser-state>off</laser-state> <controller-state>ots-state-up</controller-state> <transport-admin-state>ots-tas-ui-is</transport-admin-state> <total-rx-power>-5000</total-rx-power> <total-tx-power>-5000</total-tx-power> <rx-los-p> <is-detected>false</is-detected> <counter>0</counter> </rx-los-p> <tx-power-fail-low> <is-detected>false</is-detected> <counter>0</counter> </tx-power-fail-low> </osc-info> </osc-port> </osc-ports> </osc-oper> </data> </rpc-reply>

Configure FPD Package

Step 1 Use the Cisco-IOS-XR-fpd-infra-cfg.yang Yang model to configure FPD package.

Yang Model	Example
Cisco-IOS-XR-fpd-infra-cfg.yang	<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <fpd xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-fpd-infra-cfg"> <auto-upgrade>enable</auto-upgrade> </fpd> </config> </edit-config> </rpc>

Step 2 Use Cisco-IOS-XR-show-fpd-loc-ng-oper.yang Yang model to view the operational data for FPD package details

Yang Model	Example
Cisco-IOS-XR-show-fpd-loc-ng-oper.yang	<?xml version="1.0"?> <rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <data> <show-fpd xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-show-fpd-loc-ng-oper"> <locations> <location> <location-name>0-RP0-CPU0</location-name> <fpds> <fpd> <fpd-name>fpd_list</fpd-name> <upgrade-status>No upgrade in progress</upgrade-status> <fpd-info-detail> <location>0/RP0/CPU0</location> <card-name>NCS1010-CTR2-B-K9</card-name> <fpd-name>ADMCONFIG</fpd-name> <hw-version>0.1 </hw-version> <status>CURRENT</status> <running-version> 1.00 </running-version> <programd-version> 1.00 </programd-version> <reload-location>NOT REQ</reload-location> </fpd-info-detail> <fpd-info-detail> <location>0/RP0/CPU0</location> <card-name>NCS1010-CTR2-B-K9</card-name> <fpd-name>BIOS</fpd-name> <hw-version>0.1 </hw-version> . . . <fpd-pkg-data> <card-type>NCS1K4-AC-PSU-2</card-type> <fpd-desc>PO-SecMCU</fpd-desc> <upgrade-method>Toggle</upgrade-method> <fpd-ver> 1.05 </fpd-ver> <min-sw-ver> 1.05 </min-sw-ver> <min-hw-ver> 0.1 </min-hw-ver> <cap-bitmap>5</cap-bitmap> <reload-type>0</reload-type> </fpd-pkg-data> </package> </show-fpd> </data> </rpc-reply>

View NCS 1020 Platform Details

Use the Cisco-IOS-XR-platform-oper.yang Yang model to view the platform details of the NCS 1020 node.

Yang Models	Example
Cisco-IOS-XR-platform-oper.yang	<?xml version="1.0"?> <rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <data> <platform xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-platform-oper"> <racks> <rack> <rack-name>0</rack-name> <slots> <slot> <slot-name>RP0</slot-name> <instances> <instance> <instance-name>CPU0</instance-name> <state> <card-type>NCS1010-CTR2-B-K9</card-type> <card-redundancy-state>active</card-redundancy-state> <state>not-applicable</state> <admin-state>NSHUT,NMON</admin-state> <node-name>0/RP0/CPU0</node-name> <oper-state>IOS XR RUN</oper-state> </state> </instance> </instances> </slot> <slot> <slot-name>FT0</slot-name> <state> <card-type>NCS1010-FAN</card-type> <card-redundancy-state>red-state-none</card-redundancy-state> <state>not-applicable</state> <admin-state>NSHUT,NMON</admin-state> <node-name>0/FT0</node-name> <oper-state>OPERATIONAL</oper-state> </state> . . . </slot> <slot> <slot-name>PM1</slot-name> <state> <card-type>NCS1K4-AC-PSU-2</card-type> <card-redundancy-state>red-state-none</card-redundancy-state> <state>not-applicable</state> <admin-state>NSHUT,NMON</admin-state> <node-name>0/PM1</node-name> <oper-state>OPERATIONAL</oper-state> </state> </slot> </slots> </rack> </racks> </platform> </data> </rpc-reply>

View Performance Monitoring Parameters

Use Cisco-IOS-XR-pmengine-oper.yang Yang model to view the performance monitoring parameters on the controllers.

Yang Model	Example

Cisco-IOS-XR-pmengine-oper.yang	rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
	  <get>
	  <filter>
	  <performance-management xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-pmengine-oper">
	  <optics>
	  .
	  .
	  .
	  </performance-management>
	  </filter>
	  </get>
	</rpc>
	#####Response######
	<?xml version="1.0"?>
	<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
	 <data>
	  <performance-management xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-pmengine-oper">
	  <optics>
	  <optics-ports>
	  <optics-port>
	  <name>Ots0/0/0/0</name>
	  <optics-current>
	  <optics-second30>
	  <optics-second30-optics>
	  <optics-second30-optic>
	  <number>1</number>
	  <index>0</index>
	  <valid>true</valid>
	  .
	  .
	  .
	  </optics-second30-optics>
	  </optics-second30>
	  </optics-current>
	  </optics-port>
	  </optics-ports>
	  </optics>
	  </performance-management>
	 </data>
	</rpc-reply>

Configure Equipment Mismatch Alarm

Use the Cisco-IOS-XR-osa-ct-cfg.yang Yang model to configure the equipment mismatch alarm. For example, when the NCS 1020 node is loaded with the OLT- C card and if you try to configure the node with a different line card configuration, the equipment mismatch alarm rises.

Yang Model	Example
Cisco-IOS-XR-osa-ct-cfg.yang	`<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <resrv-cli xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-osa-ct-cfg"> <slot-info-cli> <lc-slot>0_0_NXR0</lc-slot> <card-type-cli>ncs1k-olt-r-c</card-type-cli> </slot-info-cli> </resrv-cli> </config> </edit-config> </rpc>`

View the List of Alarms on the NCS 1020 Node

Use the Cisco-IOS-XR-alarmgr-server-oper.yang Yang model to view the list of alarms generated on the NCS 1020 node.

Yang Model	Example
Cisco-IOS-XR-alarmgr-server-oper.yang	<?xml version="1.0" ?> <rpc-reply message-id="urn:uuid:518e2c10-c837-4b36-9bab-93f935148ce5" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"> <data> <alarms xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-alarmgr-server-oper"> <brief> <brief-system> <active> <alarm-info> <location>0/Rack</location> <severity>major</severity> <group>fpd-infra</group> <set-time>06/09/2022 06:26:48 UTC</set-time> <set-timestamp>1654756008</set-timestamp> <clear-time>-</clear-time> <clear-timestamp>0</clear-timestamp> <description>One Or More FPDs Need Upgrade Or Not In Current State</description> </alarm-info> <alarm-info> <location>0/RP0/CPU0</location> <severity>major</severity> <group>fpd-infra</group> <set-time>06/09/2022 06:26:49 UTC</set-time> <set-timestamp>1654756009</set-timestamp> <clear-time>-</clear-time> <clear-timestamp>0</clear-timestamp> <description>One Or More FPDs Need Upgrade Or Not In Current State</description> </alarm-info> <alarm-info> <location>0/0/NXR0</location> <severity>major</severity> <group>fpd-infra</group> <set-time>06/09/2022 06:26:51 UTC</set-time> <set-timestamp>1654756011</set-timestamp> <clear-time>-</clear-time> <clear-timestamp>0</clear-timestamp> <description>One Or More FPDs Need Upgrade Or Not In Current State</description> </alarm-info> <alarm-info> <location>0/0/NXR0</location> <severity>minor</severity> <group>software</group> <set-time>06/09/2022 06:27:13 UTC</set-time> <set-timestamp>1654756033</set-timestamp> <clear-time>-</clear-time> <clear-timestamp>0</clear-timestamp> <description>Ots0/0/0/0 - APC blocked</description> </alarm-info> </active> </brief-system> </brief> </alarms> </data> </rpc-reply>

Configure Optical Line Control Applications

Step 1 Use the Cisco-IOS-XR-olc-cfg.yang Yang model to configure various optical line applications such as link tuner, span loss, connector loss, fiber-type, span-length, gain-margin, Automatic Power Control (apc) and Power Spectral Densities (psd), and apc span mode tx and rx.

Table 1: Link tuner

Openconfig Model	Example
Cisco-IOS-XR-olc-cfg.yang	**Link tuner** <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <olc xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-olc-cfg"> <controller-rsips> <controller-rsip> <controller>Ots0/0/0/0</controller> <link-tuner> <spectrum-density>93</spectrum-density> <link-tuner-cfg-state>manual</link-tuner-cfg-state> </link-tuner> </controller-rsip> </controller-rsips> </olc> </config> </edit-config> </rpc>
	Span Loss <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <olc xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-olc-cfg"> <controller-rsips> <controller-rsip> <controller>Ots0/0/0/0</controller> <span-loss> <max-threshold>232</max-threshold> <min-threshold>192</min-threshold> </span-loss> </controller-rsip> </controller-rsips> </olc> </config> </edit-config> </rpc>

Openconfig Model	Example
	Gain Estimator The gain estimator can be set to Enable, Disable and Manual states. `<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <olc xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-olc-cfg"> <controller-rsips> <controller-rsip> <controller>Ots0/0/0/0</controller> <gain-estimator> <gain-estimator-cfg-state>manual</gain-estimator-cfg-state> </gain-estimator> </controller-rsip> </controller-rsips> </olc> </config> </edit-config> </rpc>`
	Connector loss, fiber-type, span-length, and gain-margin The supported fiber types are SMF, SMF-28e, TW-RS, TW-REACH, E-LEAF, FREE-LIGHT, METRO-CORE, TERA-LIGHT, TW-MINUS, TW-PLUS, and ULL-SMF28. `<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <olc xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-olc-cfg"> <controller-rsips> <controller-rsip> <controller>Ots0/0/0/0</controller> <rx-connector-loss>0.25</rx-connector-loss> <tx-connector-loss>0.25</tx-connector-loss> <fiber-type>smf</fiber-type> <gain-range-margin>30</gain-range-margin> <span-length>1000</span-length> </controller-rsip> </controller-rsips> </olc> </config> </edit-config> </rpc>`

Openconfig Model	Example
	APC and PSD <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <olc xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-olc-cfg"> <controller-rsips> <controller-rsip> <apc> <psds> <psd> <psd-index>1</psd-index> <psd-value>-46</psd-value> </psd> <psd> <psd-index>2</psd-index> <psd-value>-46</psd-value> </psd> </psds> <apc-cfg-state>manual</apc-cfg-state> <psd-min>-226</psd-min> </apc> </controller-rsip> </controller-rsips> </olc> </config> </edit-config> </rpc>
	APC span mode tx and rx <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <olc xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-olc-cfg"> <controller-rsips> <controller-rsip> <controller>Ots0/0/0/0</controller> <apc-span-mode-pause> <apc-span-mode-pause-dir-tx> <apc-span-mode-pause-enable/> </apc-span-mode-pause-dir-tx> <apc-span-mode-pause-dir-rx> <apc-span-mode-pause-enable/> </apc-span-mode-pause-dir-rx> </apc-span-mode-pause> </controller-rsip> </controller-rsips> </olc> </config> </edit-config> </rpc>

Step 2 Use the Cisco-IOS-XR-olc-oper.yang Yang model to retrieve span loss data, Gain estimator and Link tuner details, and alc status.

Openconfig Model	Example
Cisco-IOS-XR-olc-oper.yang	**Span loss data** **********RPC********** <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <get> <filter> <olc xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-olc-oper"> <span-loss-ctrlr-tables> <span-loss-ctrlr-table/> </span-loss-ctrlr-tables> </olc> </filter> </get> </rpc> **********Response********* <rpc-reply xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:aca06af6-9b52-4e9d-92b4-9255ced69c22"> <data> <olc xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-olc-oper"> <span-loss-ctrlr-tables> <span-loss-ctrlr-table> . . . <signal-tx-span-loss>209.4</signal-tx-span-loss> <osc-rx-span-loss>235.0</osc-rx-span-loss> <osc-tx-span-loss>231.0</osc-tx-span-loss> </span-loss-ctrlr-table> </span-loss-ctrlr-tables> </olc> </data> </rpc-reply>

Openconfig Model	Example
	Link tuner details **********RPC********** <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <get> <filter> <olc xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-olc-oper"> <link-tuner-table> <link-tuner-detail-ctrlr-tables> <link-tuner-detail-ctrlr-table/> </link-tuner-detail-ctrlr-tables> </link-tuner-table> </olc> </filter> </get> </rpc> **********Response********* <rpc-reply xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:0461c04b-642b-4a47-9e41-4d8a57d1d6c3"> <data> <olc xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-olc-oper"> <link-tuner-table> <link-tuner-detail-ctrlr-tables> <link-tuner-detail-ctrlr-table> <name>Ots0/0/0/0</name> <link-tuner-info> <status>manual</status> . . . <computed-total-noise>NA</computed-total-noise> </link-tuner-detail-ctrlr-table> </link-tuner-detail-ctrlr-tables> </link-tuner-table> </olc> </data> </rpc-reply>

Openconfig Model	Example
	Gain Estimator details **********RPC********** <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <get> <filter> <olc xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-olc-oper"> <gain-estimator-ctrlr-tables> <gain-estimator-ctrlr-table/> </gain-estimator-ctrlr-tables> </olc> </filter> </get> </rpc> **********Response********* <rpc-reply xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:48d7eca3-4382-4244-966b-598fb1d7cda4"> <data> <olc xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-olc-oper"> <gain-estimator-ctrlr-tables> <gain-estimator-ctrlr-table> <name>Ots0/0/0/0</name> <ingress-status> <status>manual</status> <last-gain-cmpt-time-stamp>2024-06-03 17:04:25</last-gain-cmpt-time-stamp> <computed-gain>261.0</computed-gain> <computed-gain-mode>Extended</computed-gain-mode> </ingress-status> </gain-estimator-ctrlr-table> </gain-estimator-ctrlr-tables> </olc> </data> </rpc-reply>

Openconfig Model	Example

	ALC Status
	`************RPC**************`
	`<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">`
	`<get>`
	`<filter>`
	`<olc xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-olc-oper">`
	`<alc-status-ctrlr-tables>`
	`<alc-status-ctrlr-table/>`
	`</alc-status-ctrlr-tables>`
	`</olc>`
	`</filter>`
	`</get>`
	`</rpc>`
	`************Response*********`
	`<rpc-reply xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:77b9dc9a-beb7-441f-ad7d-46c92682b0ea">`
	`<data>`
	`<olc xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-olc-oper">`
	`<alc-status-ctrlr-tables>`
	`<alc-status-ctrlr-table>`
	`<name>Ots0/0/0/0</name>`
	`<manager-status>idle</manager-status>`
	`<alc-start-timestamp>2024-06-03 16:40:18</alc-start-timestamp>`
	.
	.
	.
	`<node-rid>177.1.1.6</node-rid>`
	`<alc-state>complete</alc-state>`
	`</node-info>`
	`</alc-status-ctrlr-table>`
	`</alc-status-ctrlr-tables>`
	`</olc>`
	`</data>`
	`</rpc-reply>`

Configure Optical Amplifier on OLT Line Card Using Open Config Model

The openconfig-optical-amplifier Yang model uses the following naming convention for the preamplifier and the booster amplifier in the OLT line card:

R/S-<AMP TYPE><ID>

- *R*—Rack.
- *S*—Slot.
- *<AMP TYPE>*—AMP-PRE (for preamplifier) or AMP-BST (for booster amplifier).

- *ID*—The value is 0 in openconfig.

For example, the amplifiers are mentioned as 0/0-AMP-PRE0 or 0/0-AMP-BST0 which is a line port ots0/0/0/0 in the IOS-XR.

Step 1 Use the openconfig-optical-amplifier Yang model to configure the amplifier on the OLT line card.

Openconfig Model	Example

openconfig-optical-amplifier	<pre>{
	  "openconfig-optical-amplifier:optical-amplifier": {
	   "amplifiers": {
	    "amplifier": [
	     {
	      "name": "0/0-AMP-PRE0",
	      "config": {
	       "name": "0/0-AMP-PRE0",
	       "target-gain": "19.00",
	       "gain-range": "MID_GAIN_RANGE",
	       "target-gain-tilt": "3.90",
	       "enabled": true
	       }
	     },
	     {
	      "name": "0/0-AMP-BST0",
	      "config": {
	       "name": "0/0-AMP-BST0",
	       "target-gain": "19.00",
	       "target-gain-tilt": "-1.5",
	       "enabled": true
	       }
	     }
	    ]
	   }
	  }
	 }</pre>

Step 2 Get the operational data using GNMI.

```
{
  "openconfig-optical-amplifier": {
    "optical-amplifier": {
      "amplifiers": {
        "amplifier": {
          "0/0-AMP-BST0": {
            "state": {
              "enabled": true,
              "name": "0/0-AMP-BST0",
              "target-gain": 19.00,
              "target-gain-tilt": -1.5
            }
          },
          "0/0-AMP-PRE0": {
            "state": {
              "enabled": true,
              "gain-range": "MID_GAIN_RANGE",
              "name": "0/0-AMP-PRE0",
              "target-gain": 19.00,
              "target-gain-tilt": 3.90
            }
          }
```

```
            }
          }
        }
      }
    }
```

Configure Optical Amplifier on ILA Line Card Using Open Config Model

The openconfig-optical-amplifier Yang model uses the following naming convention for the two booster amplifiers in the ILA line card:

R/S-<AMP TYPE><ID>

- *R*—Rack.
- *S*—Slot.
- *<AMP TYPE>*—AMP-BST for the booster amplifier.
- *ID*—The value is 0 or 2 in openconfig.

For example, the amplifiers are mentioned as 0/0-AMP-BST0 and 0/0-AMP-BST2 which are the line ports ots0/0/0/0 and ots0/0/0/2 respectively in the IOS-XR.

Step 1 Use the openconfig-optical-amplifier Yang model to configure the amplifier on the ILA line card.

Openconfig model	Example

openconfig-optical-amplifier	{
	"openconfig-optical-amplifier:optical-amplifier": {
	"amplifiers": {
	"amplifier": [
	{
	"name": "0/0-AMP-BST0",
	"config": {
	"name": "0/0-AMP-BST0",
	"target-gain": "24.00",
	"target-gain-tilt": "-3.90",
	"enabled": false,
	"gain-range": "HIGH_GAIN_RANGE",
	}
	},
	{
	"name": "0/0-AMP-BST2",
	"config": {
	"name": "0/0-AMP-BST2",
	"target-gain": "24.00",
	"target-gain-tilt": "-3.20",
	"enabled": false,
	"gain-range": "HIGH_GAIN_RANGE"
	}
	}
	]
	}
	}
	}

Step 2 Get the operational data using GNMI.


```
{
  "openconfig-optical-amplifier": {
    "optical-amplifier": {
      "amplifiers": {
        "amplifier": {
          "0/0-AMP-BST0": {
            "state": {
              "enabled": false,
              "gain-range": "HIGH_GAIN_RANGE",
              "name": "0/0-AMP-BST0",
              "target-gain": 24.00,
              "target-gain-tilt": -3.90
            }
          },
          "0/0-AMP-BST2": {
            "state": {
              "enabled": false,
              "gain-range": "HIGH_GAIN_RANGE",
              "name": "0/0-AMP-BST2",
              "target-gain": 24.00,
              "target-gain-tilt": -3.20
            }
          }
        }
      }
    }
  }
}
```