Supported YANG Models in NCS 1014

Supported Yang Models

The following is the list of supported config, oper, and act YANG models for NCS 1014:

Table 1: Native Models

Config Models	Oper Models
Cisco-IOS-XR-osa-linesystem-cfg.yang	Cisco-IOS-XR-osa-hwmod-linesys-oper.yang
Cisco-IOS-XR-controller-ots-cfg.yang	Cisco-IOS-XR-controller-ots-oper.yang
Cisco-IOS-XR-ots-och-cfg.yang	Cisco-IOS-XR-controller-ots-och-oper.yang
Cisco-IOS-XR-controller-oms-cfg	Cisco-IOS-XR-controller-oms-oper.yang
Cisco-IOS-XR-controller-och-cfg	Cisco-IOS-XR-controller-och-oper.yang
Cisco-IOS-XR-controller-osc-cfg.yang	Cisco-IOS-XR-controller-osc-oper.yang
Cisco-IOS-XR-controller-dfb-cfg.yang	Cisco-IOS-XR-controller-dfb-oper.yang
Cisco-IOS-XR-pmengine-cfg.yang	Cisco-IOS-XR-pmengine-oper.yang
Cisco-IOS-XR-olc-cfg.yang	Cisco-IOS-XR-olc-oper.yang
Cisco-IOS-XR-fpd-infra-cfg.yang	
Cisco-IOS-XR-osa-ct-cfg.yang	Cisco-IOS-XR-osa-controller-optics-oper.yang
Cisco-IOS-XR-osa-sp-cfg.yang	Cisco-IOS-XR-osa-hwmod-linesys-oper.yang
Cisco-IOS-XR-osa-cfg.yang	Cisco-IOS-XR-osa-oper.yang
Cisco-IOS-XR-ikev2-cfg.yang	Cisco-IOS-XR-ikev2-oper.yang
Cisco-IOS-XR-controller-optics-cfg.yang	Cisco-IOS-XR-controller-optics-oper.yang
Cisco-IOS-XR-ethernet-lldp-cfg.yang	Cisco-IOS-XR-ethernet-lldp-oper.yang

Config Models	Oper Models
Cisco-IOS-XR-telemetry-model-driven-cfg.yang	Cisco-IOS-XR-telemetry-model-driven-oper.yang
Cisco-IOS-XR-ifmgr-cfg.yang	Cisco-IOS-XR-envmon-oper.yang
Cisco-IOS-XR-fpd-infra-cfg.yang	Cisco-IOS-XR-show-fpd-loc-ng-oper.yang
Cisco-IOS-XR-invproxy-hwmodule-cfg.yang	Cisco-IOS-XR-procfind-oper.yang
Cisco-IOS-XR-syncc-controller-cfg.yang	Cisco-IOS-XR-procthreadname-oper.yang
Cisco-IOS-XR-drivers-media-eth-gl-pfc-wd-cfg.yang	Cisco-IOS-XR-invmgr-oper.yang
Cisco-IOS-XR-drivers-media-eth-cfg.yang	Cisco-IOS-XR-plat-chas-invmgr-ng-oper.yang
Cisco-IOS-XR-drivers-icpe-ethernet-cfg.yang	Cisco-IOS-XR-platform-oper.yang
Cisco-IOS-XR-drivers-media-eth-cfg.yang	Cisco-IOS-XR-invmgr-diag-oper.yang
Cisco-IOS-XR-drivers-media-eth-gl-pfc-wd-cfg.yang	Cisco-IOS-XR-nto-misc-oper.yang
Cisco-IOS-XR-sysmgr-cfg.yang	Cisco-IOS-XR-wd-oper.yang
Cisco-IOS-XR-um-ncs-hw-module-osa-cfg.yang	Cisco-IOS-XR-ledmgr-oper.yang
	Cisco-IOS-XR-shellutil-filesystem-oper.yang
	Cisco-IOS-XR-shellutil-oper.yang
	Cisco-IOS-XR-drivers-media-eth-oper.yang
	Cisco-IOS-XR-mediasvr-linux-oper.yang
	Cisco-IOS-XR-drivers-media-eth-oper.yang
	Cisco-IOS-XR-sysmgr-oper.yang
	Cisco-IOS-XR-procmem-oper.yang
	Cisco-IOS-XR-procfind-oper.yang
Act Models	
Cisco-IOS-XR-upgrade-fpd-ng-act.yang	
Cisco-IOS-XR-shellutil-delete-act.yang	
Cisco-IOS-XR-drivers-media-eth-act.yang	
Cisco-IOS-XR-shellutil-copy-act.yang	
Cisco-IOS-XR-shellutil-copy-act.yang	
Cisco-IOS-XR-drivers-media-eth-act.yang	
Cisco-IOS-XR-sysmgr-act.yang	
Cisco-IOS-XR-system-reboot-act	

Config Models	Oper Models
Cisco-IOS-XR-install-act.yang	
Cisco-IOS-XR-install-augmented-act	

The following is the list of supported Open Config models:

Table 2: OpenConfig Models

openconfig-platform.yang
openconfig-platform-transceiver.yang
openconfig-terminal-device.yang
openconfig-interfaces.yang
openconfig-system.yang
openconfig-network-instance
openconfig-procmon.yang

See https://cfnng.cisco.com/ios-xr/yang-explorer/view-data-model for the list of Yang models supported by NCS 1014.

Extending Cisco Native Models for OpenConfig Support

Table 3: Feature History

Feature Name	Release Information	Feature Description
Extending Cisco Native Models for OpenConfig Support	Cisco IOS XR Release 24.3.1	The OpenConfig model is completely supported on NCS 1014 chassis, by extending the existing Cisco native model configuration. It supports the Q-margin and Enhanced Q-margin parameters as part of the extended OpenConfig model.

From R24.3.1 onwards, OpenConfig augmentation support is available for NCS 1014. For any configuration parameters that exist in the Cisco native model but are not included in the OpenConfig model, then such parameters can be added to the OpenConfig model by extending the existing Cisco native model.

Benefits of Using OpenConfig Augmentation

OpenConfig augmentation allows NCS 1014 chassis to be fully managed using OpenConfig models.

Purpose of OpenConfig Augmentation for Cisco Native Models

The purpose is to identify errors in signal transmission by including Q margin and Enhanced Q margin values in the OpenConfig model, which requires adding them as extended leaves within the existing open terminal model configuration.

Enabling OpenConfig Augmentation Support for Cisco Native Models

To enable OpenConfig Augmentation support for Cisco native models in the terminal device refer to the given example.

The entry highlighted in bold shows the newly added OpenConfig model as an extension to the existing Cisco native model.

```
module: openconfig-terminal-device
  +--rw terminal-device
     +--rw config
     +--ro state
     +--rw logical-channels
     |  +--rw channel* [index]
     |     +--rw index                          -> ../config/index
     |     +--rw config
     |     |  +--rw index?                  uint32
     |     |  +--rw description?            string
     |     |  +--rw admin-state?            oc-opt-types:admin-state-type
     |     |  +--rw rate-class?             identityref
     |     |  +--rw trib-protocol?          identityref
     |     |  +--rw logical-channel-type?   identityref
     |     |  +--rw loopback-mode?          oc-opt-types:loopback-mode-type
     |     |  +--rw test-signal?            boolean
     |     +--ro state
     |     |  +--ro index?                  uint32
     |     |  +--ro description?            string
     |     |  +--ro admin-state?            oc-opt-types:admin-state-type
     |     |  +--ro rate-class?             identityref
     |     |  +--ro trib-protocol?          identityref
     |     |  +--ro logical-channel-type?   identityref
     |     |  +--ro loopback-mode?          oc-opt-types:loopback-mode-type
     |     |  +--ro test-signal?            boolean
     |     |  +--ro link-state?             enumeration
     |     +--rw otn
     |     |  +--rw config
     |     |  |  +--rw tti-msg-transmit?             string
     |     |  |  +--rw tti-msg-expected?             string
     |     |  |  +--rw tti-msg-auto?                 boolean
     |     |  |  +--rw tributary-slot-granularity?   identityref
     |     |  +--ro state
     |     |  |  +--ro tti-msg-transmit?             string
     |     |  |  +--ro tti-msg-expected?             string
     |     |  |  +--ro tti-msg-auto?                 boolean
     |     |  |  +--ro tributary-slot-granularity?   identityref
     |     |  |  +--ro tti-msg-recv?                 string
     |     |  |  +--ro rdi-msg?                      string
     |     |  |  +--ro errored-seconds?              yang:counter64
     |     |  |  +--ro severely-errored-seconds?     yang:counter64
     |     |  |  +--ro unavailable-seconds?          yang:counter64
     |     |  |  +--ro code-violations?              yang:counter64
     |     |  |  +--ro errored-blocks?               yang:counter64
     |     |  |  +--ro fec-uncorrectable-blocks?     yang:counter64
     |     |  |  +--ro fec-uncorrectable-words?      yang:counter64
     |     |  |  +--ro fec-corrected-bytes?          yang:counter64
     |     |  |  +--ro fec-corrected-bits?           yang:counter64
     |     |  |  +--ro background-block-errors?      yang:counter64
     |     |  |  +--ro pre-fec-ber
     |     |  |  |  +--ro instant?    decimal64
     |     |  |  |  +--ro avg?        decimal64
     |     |  |  |  +--ro min?        decimal64
     |     |  |  |  +--ro max?        decimal64
     |     |  |  |  +--ro interval?   oc-types:stat-interval
     |     |  |  |  +--ro min-time?   oc-types:timeticks64
     |     |  |  |  +--ro max-time?   oc-types:timeticks64
     |     |  |  +--ro post-fec-ber
     |     |  |  |  +--ro instant?    decimal64
     |     |  |  |  +--ro avg?        decimal64
     |     |  |  |  +--ro min?        decimal64
     |     |  |  |  +--ro max?        decimal64
     |     |  |  |  +--ro interval?   oc-types:stat-interval
     |     |  |  |  +--ro min-time?   oc-types:timeticks64
     |     |  |  |  +--ro max-time?   oc-types:timeticks64
     |     |  |  +--ro q-value
     |     |  |  |  +--ro instant?    decimal64
     |     |  |  |  +--ro avg?        decimal64
     |     |  |  |  +--ro min?        decimal64
     |     |  |  |  +--ro max?        decimal64
     |     |  |  |  +--ro interval?   oc-types:stat-interval
     |     |  |  |  +--ro min-time?   oc-types:timeticks64
     |     |  |  |  +--ro max-time?   oc-types:timeticks64
     |     |  |  +--ro esnr
     |     |  |     +--ro instant?    decimal64
     |     |  |     +--ro avg?        decimal64
     |     |  |     +--ro min?        decimal64
     |     |  |     +--ro max?        decimal64
     |     |  |     +--ro interval?   oc-types:stat-interval
     |     |  |     +--ro min-time?   oc-types:timeticks64
     |     |  |     +--ro max-time?   oc-types:timeticks64
     |     |  +--ro oc-opt-ext:extended
     |     |     +--ro oc-opt-ext:state
     |     |        +--ro oc-opt-ext:enhanced-q-margin?   decimal64
     |     |        +--ro oc-opt-ext:q-margin
     |     |           +--ro oc-opt-ext:instant?    decimal64
     |     |           +--ro oc-opt-ext:avg?        decimal64
     |     |           +--ro oc-opt-ext:min?        decimal64
     |     |           +--ro oc-opt-ext:max?        decimal64
     |     |           +--ro oc-opt-ext:interval?   Decimal64
     |     |           +--ro oc-opt-ext:min-time?   decimal64
     |     |           +--ro oc-opt-ext:max-time?   decimal64
     |     +--rw ethernet
     |     |  +--rw config
     |     |  |  +--rw client-als?   enumeration
     .................
     .................

  augment /oc-platform:components/oc-platform:component:
    +--rw optical-channel
       +--rw config
       |  +--rw frequency?             oc-opt-types:frequency-type
       |  +--rw target-output-power?   decimal64
       |  +--rw operational-mode?      uint16
       |  +--rw line-port?             -> /oc-platform:components/component/name
       +--ro state
       |  +--ro frequency?                                   oc-opt-types:frequency-type
       |  +--ro target-output-power?                         decimal64
       |  +--ro operational-mode?                            uint16
       |  +--ro line-port?                                   ->
/oc-platform:components/component/name
       |  +--ro group-id?                                    uint32
       .................
       .................
```

Augmented Model

This is the augmented model configuration.

The entry highlighted in bold shows the newly added OpenConfig model as an extension to the existing Cisco native model.

```
module Cisco-IOS-XR-openconfig-terminal-device-ext{

  namespace "http://cisco.com/ns/yang/"+
        "Cisco-IOS-XR-openconfig-terminal-device-ext";

  prefix oc-opt-ext;

  import openconfig-platform {
          prefix oc-platform;
    }
   import openconfig-terminal-device {
          prefix oc-opt-term;
    }

   organization "Cisco Systems, Inc.";

   contact
    "Cisco Systems, Inc.
     Customer Service

     Postal: 170 West Tasman Drive
     San Jose, CA 95134

     Tel: +1 800 553-NETS

     E-mail: cs-yang@cisco.com";

    description
   "This module is an extension of optical terminal device model
     and contains the definition of extended parameters for Optical
     Channels in order to optimize the AC1200 settings to get the highest
     performance and spectral efficiency.

     This module contains definitions for the following management objects:
     General Parameters
     Submarine Parameters

     Copyright (c) 2013-2023 by Cisco Systems, Inc.
     All rights reserved.";

  revision 2023-05-08 {
      description
          "Addition of baud-rate configuration";
  }

revision 2023-02-28 {
      description
          "Addition of input power lower and upper thresholds parameters and
           Fastpoll enable configuration";
      reference "7.8.1";
  }
 revision 2020-08-30 {
      description
          "IOS XR 6.0 revision";
      reference "7.3.1";
  }

grouping  terminal-device-optical-channel-ext-info {
     description "Submarine parameters for optical channel";
     leaf optics-cd-min {
         type int32 {
             range "-350000..350000";
         }
         description
             "Select min chromatic dispersion (in units of
             ps/nm)";
     }
     leaf optics-cd-max {
         type int32 {
             range "-350000..350000";
         }
         description
             "Select max chromatic dispersion (in units of
             ps/nm)";
     ................
     ................

augment "/oc-platform:components/oc-platform:component/oc-opt-term:optical-channel" {
    container extended {
        description
            "Enclosing container for the list of Subsea parameters";

        container config {
      description
                "Extended Configuration parameters";
            leaf rx-voa-target-power {
               type int32 {
               range "-190..30";
            }
            description "Receive Target Power in increments of 0.1 dBm
                 Default value is -5 dBm";
            }
            leaf rx-voa-fixed-ratio {
               type int32 {
               range "100..1700";
            }
            description "Receive Ratio of Optical Attenuation in
                 increments of 0.01 dB Default value is 15 dB";
            }
            leaf fastpoll-sop-enable {
               type boolean {
            }
            description "Receive Fastpoll status if it is enabled or
                 disabled";
            }
            uses terminal-device-optical-channel-ext-info;
        }

        container state {

            config false;

            description
                "Extended Operational parameters";
     ................

augment
"/oc-opt-term:terminal-device/oc-opt-term:logical-channels/oc-opt-term:channel/oc-opt-term:otn"
 {
                container extended {
augment
"/oc-opt-term:terminal-device/oc-opt-term:logical-channels/oc-opt-term:channel/oc-opt-term:otn"
 {
                container extended {
                  config false;
                  container state {
            config false;

        leaf enhanced-q-margin {
           type decimal64 {
           fraction-digits 2;
                   }
           description
             "Enhanced Instantaneous Q-Margin Value";
                }

                container q-margin {
            description
                "Q-Margin value in dB with two decimal precision. Values
                include the instantaneous, average, minimum, and maximum
                statistics. If avg/min/max statistics are not supported,
                the target is expected to just supply the instant value";

                        leaf instant {
               type decimal64 {
                  fraction-digits 2;
            }
            description
               "The instantaneous value of the q-margin (in units of dB).";
            }

                        leaf avg {
               type decimal64 {
                  fraction-digits 2;
            }
            description
               "The arithmetic mean value of the q-margin over the time interval of 10 or
30 sec. (in units of dB).";
            }

                        leaf min {
               type decimal64 {
                  fraction-digits 2;
            }
            description
               "The minimum value of the q-margin over the time interval of 10 or 30 sec.
(in units of dB).";
            }

                        leaf max {
               type decimal64 {
                  fraction-digits 2;
            }
            description
               "The maximum value of the q-margin over the time interval of 10 or 30 sec.
(in units of dB).";
            }
  leaf min-time {
               type decimal64 {
                  fraction-digits 2;
                           }
            description
               "The absolute time at which the minimum value occurred.
                            The value is the timestamp in nanoseconds relative to
                            the Unix Epoch (Jan 1, 1970 00:00:00 UTC).";
            }
```

```
                        leaf max-time {
               type decimal64 {
                  fraction-digits 2;
                           }
            description
               "The absolute time at which the maximum value occurred.
                            The value is the timestamp in nanoseconds relative to
                            the Unix Epoch (Jan 1, 1970 00:00:00 UTC).";
            }
                        leaf interval {
               type decimal64 {
                  fraction-digits 2;
                           }
            description
                "If supported by system, this reports the time interval
                           over which the min/max/average statistics are computed by the
system.";
            }
                }
      }
      description
       "This augment extends the operational data of
                         'terminal-otn-protocol-state'";
   }
description
      "This augment extends the operational data of
                        'terminal-otn-protocol-top'";

}
}
```

Note The CLI appearing in bold is the augmented code added to the existing Cisco native model.

OpenConfig Support for FEC Data

Table 4: Feature History

Feature Name	Release Information	Feature Description
OpenConfig Support for FEC Data	Cisco IOS XR Release 24.3.1	OpenConfig model support is added for Forward Error Correction (FEC) data on the NCS 1014 chassis. It helps in avoiding deviations in the **pre-fec-ber** and **post-fec-ber** leaves for pluggables.

In the NCS 1014 chassis, Forward Error Correction (FEC) is enabled on all 400G pluggables and in 100G deployments. The OpenConfig model allows monitoring of FEC statistics through NetConf and telemetry for the NCS1K14-2.4T-X-K9 card.

Purpose of OpenConfig Support for FEC Data in NCS 1014 Chassis

In the NCS 1014 chassis, Cisco native models do not support the **pre-fec-ber** and **post-fec-ber** containers which are needed for monitoring FEC statistics on the NCS1K14-2.4T-X-K9 card. The OpenConfig model can be used for this purpose by extending the existing native commands.

Benefits of Providing OpenConfig Support for FEC Data in NCS 1014

Enabling OpenConfig model support for FEC data in NCS 1014 avoids deviations in the **pre-fec-ber** and **post-fec-ber** leaves for transceivers across all routing and optical platforms.

Supported FEC Parameters

The instant data and performance monitoring statistics parameters are enabled for supporting FEC on transceivers are:

- min,
- max,
- avg,
- min-time and
- max-time,
- instant.

Enabling OpenConfig Support for FEC Data

To enable OpenConfig operations and telemetry support for FEC data, refer to the example.

The entry highlighted in bold shows the newly added OpenConfig model telemetry support for FEC data.

```
module: openconfig-platform-transceiver
 augment /oc-platform:components/oc-platform:component:
  +--rw transceiver
     +--rw config
      | +--rw enabled? boolean
      | +--rw form-factor-preconf? identityref
      | +--rw ethernet-pmd-preconf? identityref
      | +--rw fee-mode? identityref
     +--ro state
      |  +--ro enabled? boolean
      |  +--ro form-factor-preconf? identityref
      |  +--ro ethernet-pmd-preconf? identityref
      |  +--ro fee-mode? identityref
      |  +--ro present? enumeration
      |  +--ro form-factor?  identityref
      |  +--ro connector-type? identityref
      |  +--ro vendor?    string
      |  +--ro vendor-part?    string
      |  +--ro vendor-rev? string
      |  +--ro ethernet-pmd? identityref
      |  +--ro sonet-sdh-compliance-code? identityref
      |  +--ro otn-compliance-code? identityref
      |  +--ro serial-no? string
      |  +--ro date-code? oc-yang:date-and-time
      |  +--ro fault-condition? boolean
      |  +--ro fee-status? identityref
      |  +--ro fee-uncorrectable-blocks?  yang:counter64
      |  +--ro fee-uncorrectable-words? yang:counter64
      |  +--ro fee-corrected-bytes? yang:counter64
      |  +--ro fee-corrected-bits? yang:counter64
  +--ro pre-fec-ber
      | +--ro instant? decimal64
      | +--ro avg? decimal&
      | +--ro min? decimal64
      | +--ro max? decimal&
      | +--ro interval? oc-types:stat-interval
      | +--ro min-time? oc-types:timeticks64
      | +--ro max-time?  oc-types:timeticks64
  +--ro post-fec-ber
      | +--ro instant? decimal64
      | +--ro avg? decimal64
      | +--ro min? decimal64
      | +--ro max? decimal64
      | +--ro interval? oc-types:stat-interval
      | +--ro min-time? oc-types:timeticks64
      | +--ro max-time?  oc-types:timeticks64
   +--ro output-power
      | +--ro instant? decimal64
      | +--ro avg? decimal64
      | +--ro max-time? oc-types:timeticks64
   +--ro input-power
      | +--ro instant? decimal64
      | +--ro avg? decimal64
      | +--ro min? decimal64
      | +--ro max? decimal64
      | +--ro interval? oc-types:stat-interval
      | +--ro min-time? oc-types:timeticks64
      | +--ro max-time? oc-types:timeticks64
   +--ro laser-bias-current
      .......
+--ro laser-bias-current
  +--ro instant? decimal64
  +--ro avg? decimal64
  +--ro min? decimal64
  +--ro max? decimal64
  +--ro interval? oc-types:stat-interval
  +--ro min-time? oc-types:timeticks64
  +--ro max-time? oc-types:timeticks64
augment / oc-if:interfaces/oc-if:interface/oc-if:state:
+--ro transceiver? -> /oc-platform:components/component[oc
platform:name=current()/../oc-port:hardware-port]/ocplatform:subcomponents/subcomponent/name
augment / oc-if:interfaces/oc-if:interface/oc-if:state:
+--ro physical-channel* -> /oc-platform:components/component[oc
platform:name=current()/../oc-transceiver:transceiver]/transceiver/physical►
channels/channel/index
```

Instant value example:

```
instant value:
RP/0/RP0/CPU0:N112#sh controllers fourHundredGigEctrlr 0/2/0/1
Operational data for interface FourHundredGigECtrlr0/2/0/1:

State:
    Administrative state: enabled
    Operational state: Up
    LED state: Green On
    Maintenance: Disabled
    AINS Soak: None
      Total Duration: 0 hour(s) 0 minute(s)
      Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
    Laser Squelch: Disabled
    Insert Idle Ingress: Disabled
    Insert Idle Egress: Disabled

Phy:
    Media type: Not known
    Statistics:
        FEC:
            Corrected Codeword Count: 35409983               Valid: True       Start time:
 14:45:53 Mon Aug 19 2024
            Uncorrected Codeword Count: 56                   Valid: True       Start time:
 14:45:53 Mon Aug 19 2024
        PCS:
            Total BIP errors: 0                              Valid: True       Start time:
 14:45:53 Mon Aug 19 2024
            Total frame errors: 0                            Valid: False      Start time:
 14:45:53 Mon Aug 19 2024
            Total Bad SH: 0                                  Valid: False      Start time:
 14:45:53 Mon Aug 19 2024

Autonegotiation disabled.

Operational values:
    Speed: 400Gbps
    Duplex: Full Duplex
    Flowcontrol: None
    Loopback: None (or external)
    Pre FEC BER: 1.5E-09
    Post FEC BER: 0.0E+00
    BER monitoring:
        Not supported
    Forward error correction: Standard (Reed-Solomon)
    Holdoff Time: 0ms
```

Output CLI example of pre-fec and post-fec ber parameters:

```
min/max/avg:
RP/0/RP0/CPU0:N112#sh controllers fourHundredGigEctrlr 0/2/0/1 pm current 15-min fec

Ethernet FEC in the current interval [06:00:00 - 06:07:12 Tue Aug 20 2024]

FEC current bucket type : Valid
EC-WORDS : 313148 Threshold : 0 TCA(enable) : NO
UC-WORDS : 0 Threshold : 0 TCA(enable) : NO

MIN AVG MAX Threshold TCA Threshold TCA
(min) (enable) (max) (enable)
PreFEC BER : 8.2E-10 1.7E-09 2.6E-09 0E-15 NO 0E-15 NO
PostFEC BER :0E-15 0E-15 0E-15 0E-15 NO 0E-15 NO

Last clearing of "show controllers ETHERNET" counters 13:52:01
RP/0/RP0/CPU0:N112#
```