System Setup and Software Installation Guide for Cisco NCS 1010, IOS XR Releases

First Published: 2023-12-08

Last Modified: 2025-04-04

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

CONTENTS

CHAPTER 1

Cisco NCS 1010 Optical Line System - An Overview

This chapter provides an overview for NCS 1010 line system.

NCS 1010 Chassis and Line Cards

Cisco NCS 1010 is a next-generation optical line system optimized for ZR/ZR+ WDM router interfaces. Its salient features are:

- Provides point-to-point connectivity between routers with WDM interfaces.
- Multiplexes the signals received from multiple routers over a single fiber.
- With one MPO port, it can be scaled to 8 Degree.
- Caters to C-band WDM transmission to maximize capacity, and can be enhanced to C+L combined band in the future.

Cisco NCS 1010 is a 3RU chassis that has an in-built External Interface Timing Unit (EITU) and the following field-replaceable modules.

- Controller
- Two power supply units
- Two fan trays
- Fan filter
- Line card

See Hardware Installation Guide for Cisco NCS 1010 and Cisco NCS 1000 Passive Modules for more detailed images.

Line Cards

There are five different variants of the line card:

- OLT-C Line Card: C-band Optical Line Terminal without Raman
- OLT-R-C Line Card: C-band Optical Line Terminal with Raman
- ILA-C Line Card: C-band In-Line Amplifier without Raman
- ILA-R-C Line Card: C-band In-Line Amplifier with one side Raman
- ILA-2R-C Line Card: C-band In-Line Amplifier with both sides Raman
- OLT-L Line Card: L-band Optical Line Terminal
- ILA-L Line Card: L-band In-Line Amplifier

OLT-C Line Card

The C-band Optical Line Terminal without Raman (OLT-C) line card includes the following features:

- 25-dBm line preamplifier True Variable Gain (TVG) Erbium-Doped Fiber Amplifier (EDFA) with two switchable gain ranges
- Dedicated amplification of the odd and even add channels through an embedded Fixed Gain (FG) EDFA
- 23-dBm line boost-amplifier TVG EDFA single gain range
- Dedicated EDFA for noise loading
- Embedded Optical Time Domain Reflectometer (OTDR) for line RX and TX monitoring
- 37 ports Optical Channel Monitoring (OCM)
- Dedicated Tunable Laser (TL) enabling Connection Verification (CV) and patch cord discovery features
- Up to 30 EXP ports
- Embedded Optical Service Channel at Fast Ethernet (FE)
- Multiplexing and demultiplexing of odd and even channels
- C+L combiner for multiplexing and demultiplexing L-band channels
- 2x2 switch to reverse transmit direction of Optical Service Channel (OSC)-C
- Fiber reflectors to support fiber end detection by OTDR

OLT-R-C Line Card

The C-band Optical Line Terminal with Raman (OLT-R-C) line card includes the features of the OLT-C line card along with the Raman amplifier.

The following are the features of the Raman amplifier:

- Five different pump wavelengths for supporting C+L Raman amplification
- Embedded Distributed Feedback (DFB) laser at 1568.77 nm (class 1M) to be used for optical safety (link continuity)

- Full monitoring of pumps, DFB laser and signal power
- Raman pump back-reflection detector
- Meets class 1M Laser safety.
- Additional Photodiode (PD) to monitor remnant pump power at the far end

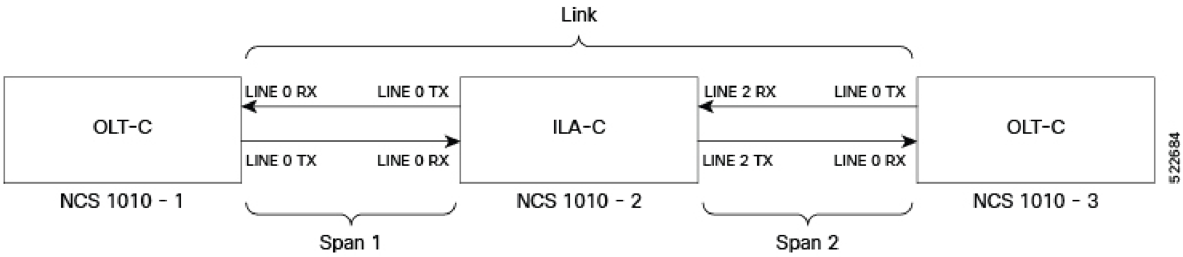
ILA-C Line Card

The C-band In Line Amplifier without Raman (ILA-C) line card includes the following features:

- Two independent TVG EDFA block, covering full operative gain ranging 8–36 dB
- Each EDFA block can provide up to 23 dBm output power
- Dynamic Gain Equalization (DGE) embedded capability to compensate for line tilt and ripple
- Embedded OTDR for line1/2-RX/TX monitoring
- Four-ports OCM for channels monitoring
- Embedded Optical Service Channel at Fast Ethernet (FE)
- C+L combiner for multiplexing/demultiplexing L-band channels
- Dedicated ports for amplifiers output monitoring
- 2x2 switch to reverse transmit direction of OSC-C for both directions
- Fiber reflectors to support fiber end detection by OTDR

The following image displays the port connection between the ILA-C line card and OLT-C line cards.

Figure 1: ILA-C Line Card Port Connection



ILA-R-C Line Card

The C-band In Line Amplifier with Raman (ILA-R-C) line card includes the features of ILA-C and Raman amplifier.

ILA-2R-C Line Card

The C-band In-Line Amplifier with two Raman (ILA-2R-C) line card includes the features of the ILA-C and Raman amplifier on both directions.

OLT-L Line Card

The L-band Optical Line Terminal (OLT-L) line card includes the following features:

- 25-dBm line preamplifier True Variable Gain (TVG) Erbium-Doped Fiber Amplifier (EDFA) with two switchable gain ranges
- Dedicated amplification of the odd and even add channels through an embedded Fixed Gain (FG) EDFA
- 24.5-dBm line boost-amplifier TVG EDFA single gain range
- 15-dBm ADD-side boost-amplifier TVG EDFA with single gain range of 16 dB
- Dedicated EDFA for noise loading
- 37 ports Optical Channel Monitoring (OCM)
- Dedicated Tunable Laser (TL) enabling Connection Verification (CV) and patch cord discovery features
- Up to 30 EXP ports
- Embedded Optical Service Channel at Fast Ethernet (FE) at 184.45 THz (1625.33 nm)
- Multiplexing and demultiplexing of odd and even channels
- 2x2 switch to reverse transmit direction of Optical Service Channel OSC-L

ILA-L Line Card

The L-band In Line Amplifier (ILA-L) line card includes the following features:

- Two independent TVG EDFA block, covering full operative gain ranging 10.8–32.8 dB
- Each EDFA block can provide up to 24.5-dBm total output power
- Dynamic Gain Equalization (DGE) embedded capability to compensate for line tilt and ripple
- Four-ports OCM for channels monitoring
- Embedded Optical Service Channel at Fast Ethernet (FE)
- Dedicated ports for amplifiers output monitoring
- 2x2 switch to reverse transmit direction of OSC-L for both directions

External Interface Timing Unit

The External Interface Timing Unit (EITU) manages the control plane interfaces and includes all user external interfaces (timing and management). It is connected to the controller with a redundant 10G Ethernet bus.

The following is the list of the available user interfaces:

- Coaxial connector for GPS antenna RF input (with +5V antenna power, if necessary)
- Console/Universal Asynchronous Receiver/Transmitter (UART) Interface (1x)
- Two Small Form-Factor Pluggables (SFP) for 1GE optical PTP port (1588 and SyncE)
- Two SFPs for 1GE optical User Data Channels (UDC)

- Three USB 2.0 type A, 1.8A max @5V/12V (with Cisco NCS 1000 Breakout Patch Panel support)
- Coaxial connector for 10MHz sync signal (bidirectional)
- Coaxial connector for 1PPS sync signal (bidirectional)
- RJ45 for 1588 TOD (1x)
- Three 10/100/1000 RJ-45 Ethernet management ports and Interconnection Link (ILINK)

Cisco NCS 1000 Passive Modules

The Cisco NCS 1000 passive modules power the Cisco NCS 1010 chassis to offer an optical line system solution. The passive modules enable the NCS 1010 chassis to implement long-haul and metro topologies. The Cisco NCS 1010 supports the following passive modules:

Cisco NCS 1000 Breakout Patch Panel

Cisco NCS 1000 Breakout Patch Panel is colorless breakout-modular patch panel. It is powered by the NCS 1010 chassis using a single USB 2.0 cable from the NCS 1010 EITU. The breakout panel contains four USB 2.0 connections that power the breakout modules. It allows connections between the OLT-C and OLT-R-C line cards that are installed in the NCS 1010 chassis and the four breakout modules using MPO cables. The breakout panel supports up to 72 colorless Mux/Demux channels and 8-directional interconnections. The breakout panel is 4 RU high and has adjustable fiber guides for fiber routing. The empty slots are covered with dummy covers. The panel is shipped with USB 2.0 connectors that are connected to the corresponding dummy covers. The plastic transparent cover can be installed in front of the panel for fiber protection. The panel is designed to fit a 19-inch rack. The panel can also be installed on ETSI and 23-inch rack using adapter brackets.

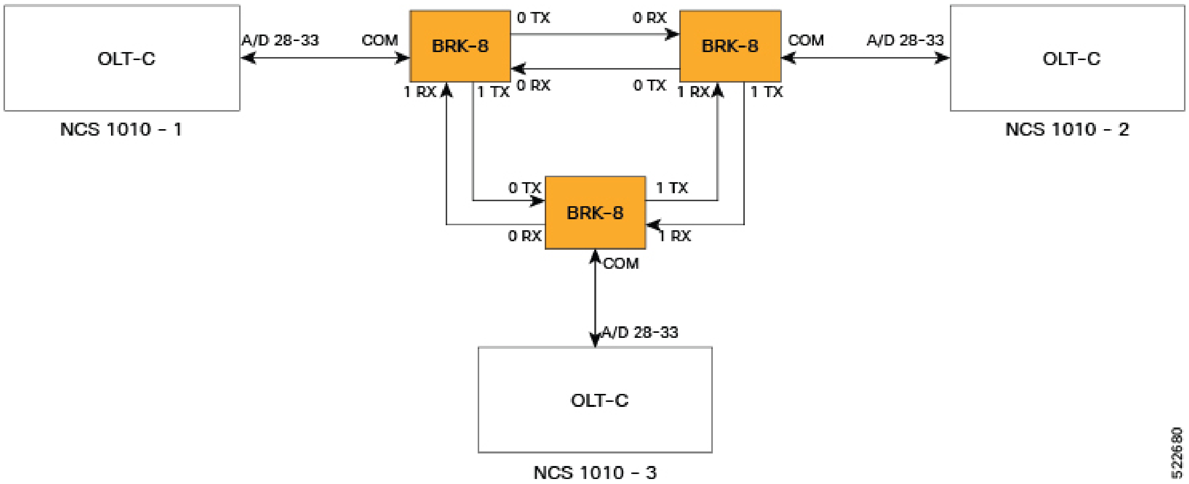
The breakout panel supports the following breakout modules:

NCS1K-BRK-8

The NCS1K-BRK-8 module provides the breakout of 16 fibers from an MPO-24 connector to 8 duplex line card connectors. It essentially performs an optical connection adaptation of MPO-to-LC connectors for the ADD/DROP signals of the MPO ports of OLT line cards. For each port (MPO and LC), power monitors with tone detection capability are available. A filtered optical loopback (191.175 THz) from one MPO input port (fiber-1) to all MPO output ports is available for connection verification.

The following image displays the port connection between BRK-8 and OLT-C cards.

Figure 2: BRK-8 Panel Port Connection with OLT-C Cards

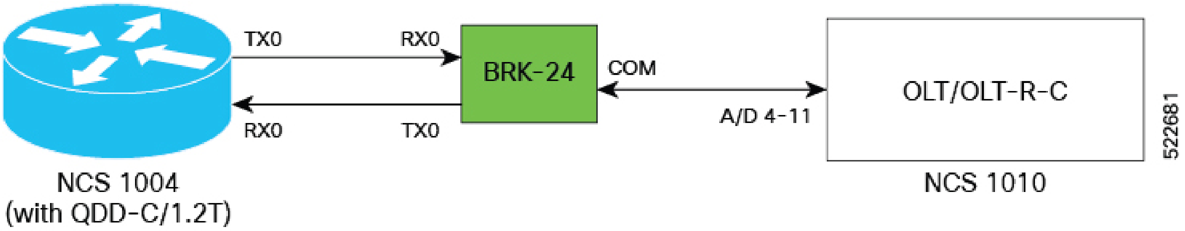


NCS1K-BRK-24

The NCS1K-BRK-24 module provides the breakout of 16 fibers from an MPO-24 connector to 24 duplex LC connectors. The signals on each fiber from the MPO input ports are split over three LC output ports by a 1x3 optical splitter. The signals from the three adjacent input LC ports are combined into a single MPO fiber output port through a 1x3 optical coupler. For each port (MPO and LC), power monitors with tone detection capability are available. A filtered optical loopback (191.175 THz) from one MPO input port (fiber-1) to all MPO output ports is available for connection verification.

The following image displays port connections between BRK-24 panel and NCS 1010 and NCS 1004 chassis.

Figure 3: Port Connections Between BRK-24 Panel and NCS 1010 and NCS 1004 Chassis



Cisco NCS 1000 32-Channel Mux/Demux Patch Panels

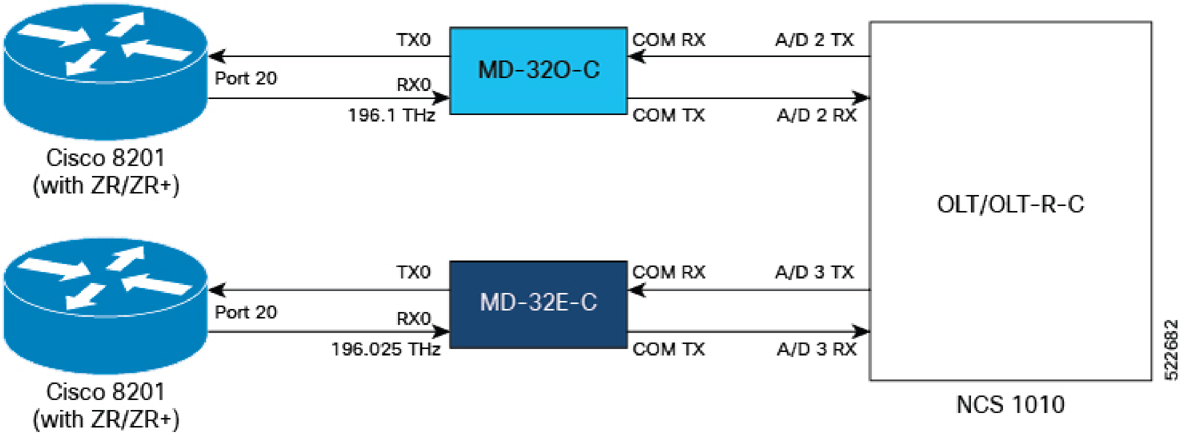
The Cisco NCS 1000 32-Channel Mux/Demux patch panels are a pair of passive Athermal Arrayed Waveguide Grating (AAWG) based modules (PIDs NCS1K-MD-32O-C and NCS1K-MD-32E-C). Each Mux/Demux panel has 32 channels and works as an add/drop unit for the OLT-C and OLT-R-C line cards. Each Mux/Demux panel allows the multiplexing and demultiplexing of 32 channels with 150-GHz spacing. 75-GHz frequency shift exists between the ODD and EVEN panels. When both panels are used on the same OLT (OLT-C and OLT-R-C) line cards, the combined capacity becomes 64 channels with 75-GHz spacing. Each Mux/Demux panel provides a wide optical pass-band support. When used as a standalone, each panel acts as an add/drop unit for 32 channels at 140 GBd.

The NCS1K-MD-32O/E-C panel operates in C-band.

The Cisco NCS 1000 Mux/Demux patch panels are fully passive. The units are powered with a USB 2.0 connection in the NCS 1010 chassis. The panels are capable of monitoring channel power, verifying connection, detecting tone, and reporting the inventory data.

The following image displays the port connection between the Mux/Demux panels and NCS 1010 and routers.

Figure 4: Port Connection between the Mux/Demux Panels and NCS 1010 and Routers



Document Objective

Cisco Network Convergence System (NCS) 1010 platform has the following configuration guides.

- The *Cisco NCS 1010 System Setup and Software Installation Guide* describes how to bring up the NCS 1010 system and perform the required software installation.
- The *Cisco NCS 1010 Datapath Configuration Guide* describes how to configure various datapaths on NCS 1010.
- The *Cisco NCS 1010 Optical Applications Configuration Guide* describes multiple optical applications on NCS 1010, that help to bring up the link and maintain traffic on the link.

Document Organization

This document is organized into the following chapters:

Chapter	Description
Setup Procedures, on page 13	Various methods to boot up the Cisco NCS 1010 system, and preliminary checks to be performed after successfully logging into the console and the suggested corrective actions if any setup issue is detected.
Deploy NCS 1010 Using Classic ZTP, on page 81	Zero Touch Provisioning to boot up the Cisco NCS 1010 system .

Chapter	Description
Disaster Recovery, on page 115	The disaster recovery process and the health check of backup ISO image.
Workflow to Setup and Upgrade NCS 1010, on page 59	Procedures to upgrade the Cisco IOS XR software and FPDs.
Remote Node Management in NCS 1010	Manage an ILA node remotely in NCS 1010.
Configure CDP	Configures Cisco Discovery Protocol (CDP) in NCS 1010.
Daisy Chain, on page 131	Describes how you can connect NCS 1010 devices in a Daisy Chain topology.
Configure Access Control List, on page 137	Procedures to configure access control lists (ACL).

CHAPTER 2

Key Concepts

Use this information to understand the key terms, concepts, types of releases relevant to setting up and upgrading Cisco IOS XR Optical Site Manager System Setup and Software Installation Guide.

This section contains the following topics:

Types of Releases

Cisco IOS XR software model has three types of software releases. The software images are available for download at the Cisco Software Download page.

Feature Release

A Feature Release (FR) contains new features and support for new hardware. Feature releases have the X.X.1 (dot one) designation. For example, releases 24.1.1 and 24.2.1 are feature releases. The list of features added to a feature release is provided in the Release notes along with the installation instructions and dependencies.

Maintenance Release

A maintenance release is the primary mechanism to deliver groups of critical bug fixes to the software feature releases.

SMUs Release

A Software Maintenance Unit (SMU) is a fix that is provided until the End of Maintenance (EoM) of the release. The fix is also committed into the next shipping release. SMUs are posted under `Cisco IOS XR Software Maintenance Upgrade` on the Cisco Software Download page. Each SMU is customized for a specific software release.

For more information on release numbering, types of releases, and their timelines, see Software Lifecycle Support Statement - IOS XR.

Files in Cisco Software Download Page

The following table describes the files available for download from the Cisco Software Download page for Cisco NCS1010.

Table 1: IOS XR Software Installation Files in Cisco Software Download Page

Package File	Example	Description
ncs1010-x64-<*rel. no.*>.iso	ncs1010-x64-24.3.1.iso	Contains required core packages, including operating system, Admin, Base, Forwarding, SNMP Agent, FPD, and Alarm Correlation and Netconf-yang, Telemetry, Extensible Markup Language (XML) Parser, HTTP server packages.
Individually Installable Packages		
xrtelnet-<*rel. no.*>x86_64.rpm xr-telnet-ncs1010-<*rel. no.*>.x86_64.rpm	xr-telnet-24.3.1.v1.0.0-1.x86_64.rpm xr-telnet-ncs1010-24.3.1.v1.0.0-1.x86_64.rpm	Install these packages to support Telnet.
xrcdp-rpms. <*rel. no.*>.rpm xr-cdp-ncs1010- <*rel. no.*>.x86_64.rpm	xr-cdp-24.3.1.v1.0.0-1.x86_64.rpm xr-cdp-ncs1010-24.3.1.v1.0.0-1.x86_64	Install these packages to support CDP.

Command Modes

The NCS 1010 runs on virtualized Cisco IOS XR software. Therefore, the CLI commands must be executed on virtual machines, namely the IOS XR.

The command modes are applicable for the Cisco Series NCS 1010. This table lists the command modes for the IOS XR.

Command Mode	Description
IOS XR EXEC mode (IOS XR execution mode)	Run commands on the XR to display the operational state of the NCS1010 Example: `RP/0/RP0/CPU0:ios#`

Command Mode	Description

IOS XR Config mode
(IOS XR configuration mode)
	Perform security, routing, and other XR feature configurations on the IOS XR.
	Example:
	`RP/0/RP0/CPU0:ios#`**`configure`**
	`RP/0/RP0/CPU0:ios(config)`

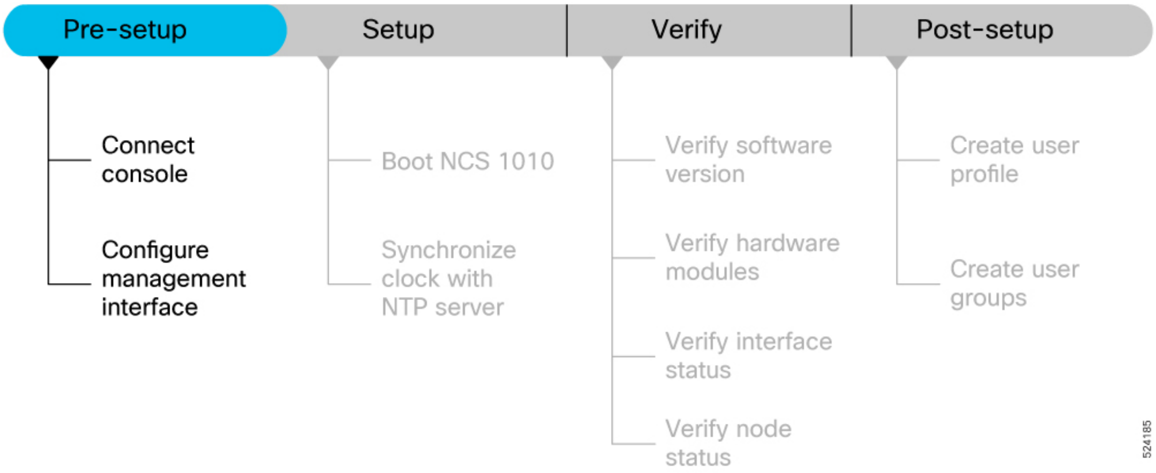
CHAPTER 3

Setup Procedures

Prerequisites to Setup NCS 1010

Complete the following prerequisite tasks to prepare the NCS 1010 for seamless setup.

Figure 5: Pre-setup Workflow for the Cisco NCS 1010



This section contains the following topics:

Connect Console Port to Terminal

The console port on the NCS 1010 is used to log into a NCS 1010 directly without a network connection using a terminal emulation program like HyperTerminal.

Procedure

Step 1 Connect the NCS 1010 to a terminal.

a) Connect the console (or rollover) cable to the console port on the NCS 1010.
b) Use the correct adapter to connect the other end of the cable to your terminal or PC.

Step 2 Configure the console port to match the following default port characteristics.

a) Launch the terminal session.
b) In the **COM1 Properties** window, select **Port Settings** tab, and enter the following settings:

- Speed – 9600
- Data Bits – 8
- Parity – none
- Stop bits – 1
- Flow Control – none

Step 3 Click **OK**.

You should see a blinking cursor in the HyperTerminal window indicating successful connection to the console port.

Configure Management Interface

The management interface can be used for system management and remote communication. To use the management interface for system management, you must configure an IP address and subnet mask. To use the management interface for remote communication, you must configure a static route. Use this procedure when NCS 1010 chassis is not booted using ZTP.

Before you begin

- Consult your network administrator to procure IP addresses and a subnet mask for the management interface.
- Ensure that the management interface is connected to the management network.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:ios#configure
```

Enters IOS XR configuration mode.

Step 2 **interface mgmtEth** *rack/slot/instance/port*

Example:

```
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/0
```

Enters interface configuration mode for the management interface.

Step 3 **ipv4 address** *ipv4-address subnet-mask*

Example:

```
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 192.0.2.254 255.255.255.0
```

Assigns an IP address and a subnet mask to the management interface.

Step 4 **no shutdown**

Example:

```
RP/0/RP0/CPU0:ios(config-if)#no shutdown
```

Places the management interface in an "up" state.

Step 5 **exit**

Example:

```
RP/0/RP0/CPU0:ios(config-if)#exit
```

Exits the management interface configuration mode.

Step 6 **ncs1010 static address-family ipv4 unicast** *0.0.0.0/0 default-gateway*

Example:

```
RP/0/RP0/CPU0:ios(config)#ncs1010 static address-family ipv4 unicast 0.0.0.0/0 198.51.100.4
```

Specifies the IP address of the default gateway to configure a static route. This IP address must be used for communication with devices on other networks.

Step 7 Use the **commit** or **end** command.

commit-Saves the configuration changes and remains within the configuration session.

end-Prompts user to take one of these actions:

- **Yes**-Saves configuration changes and exits the configuration session.
- **No**-Exits the configuration session without committing the configuration changes.
- **Cancel**-Remains in the configuration session without committing the configuration changes.

What to do next

Connect the management interface to the Ethernet network. Establish a SSH or Telnet connection to the management interface using its IP address.

Link Layer Discovery Protocol Support on Management Interface

The Link Layer Discovery Protocol (LLDP) support on management interface feature requires a system to form LLDP neighbor relationship over the system management interface, through which it advertises and learns LLDP neighbor information. This information about neighbors used to learn about the neighbors and in turn the topology of the devices for Operations, Administration, and Maintenance (OAM) purposes.

Advantages of LLDP

- Provides support on non-Cisco devices.
- Enables neighbor discovery between non-Cisco devices.

Limitation

- When you disable LLDP globally, the LLDP gets disabled on all the interfaces.

Note By default, LLDP is enabled for NCS 1010. But when you enable and disable LLDP in the global configuration mode, LLDP gets disabled on all the interfaces.

Workaround: You must enable LLDP globally or reload the NCS1010.

Cisco Discovery Protocol (CDP) vs LLDP

The CDP is a device discovery protocol that runs over Layer 2. Layer 2 is also known as the data link layer that runs on all Cisco devices, such as routers, bridges, access servers, and switches. This protocol allows the network management applications to automatically discover and learn about other Cisco devices that connect to the network.

The LLDP is also a device discovery protocol that runs over Layer 2. This protocol allows the network management applications to automatically discover and learn about other non-Cisco devices that connect to the network.

Interoperability between non-Cisco devices using LLDP

LLDP is also a neighbor discovery protocol that is used by network devices to advertise information about themselves to other devices on the network. This protocol runs over the data link layer, which allows two systems running different network layer protocols to learn about each other.

With LLDP, you can also access the information about a particular physical network connection. If you use a non-Cisco monitoring tool (through SNMP), LLDP helps you identify the Object Identifiers (OIDs) that the system supports. The following OIDs are supported:

- 1.0.8802.1.1.2.1.4.1.1.4
- 1.0.8802.1.1.2.1.4.1.1.5
- 1.0.8802.1.1.2.1.4.1.1.6
- 1.0.8802.1.1.2.1.4.1.1.7
- 1.0.8802.1.1.2.1.4.1.1.8
- 1.0.8802.1.1.2.1.4.1.1.9
- 1.0.8802.1.1.2.1.4.1.1.10
- 1.0.8802.1.1.2.1.4.1.1.11
- 1.0.8802.1.1.2.1.4.1.1.12

Neighbor Discovery

System advertises the LLDP TLV (Type Length Value) details over the management network using which other devices in the management network can learn about this device.

Configuring LLDP

- LLDP full stack functionality is supported on all three management interfaces that are supported in NCS 1010.
- You can selectively enable or disable LLDP on any of the management interfaces on demand.
- You can selectively enable or disable LLDP transmit or receive functionality at the management interface level.
- Information gathered using LLDP can be stored in the device Management Information Database (MIB) and queried with the Simple Network Management protocol (SNMP).
- LLDP operational data is available in both CLI and netconf-yang interface.

Enabling LLDP Globally

When you enable LLDP globally, all interfaces that support LLDP are automatically enabled for both transmit and receive operations.

Note You can override this default operation at the interface to disable receive or transmit operations.

The following table describes the global LLDP attributes that you can configure:

Table 2:

Attribute	Default	Range	Description
Holdtime	120	0–65535	Specifies the holdtime (in sec). Holdtime refers to the time or duration that an LLDP device maintains the neighbor information before discarding.
Reinit	2	2–5	Delay (in sec) for LLDP initialization on any interface
Timer	30	5–65534	Specifies the rate at which LLDP packets are sent (in sec)

The following example shows the commands to configure LLDP globally. The global LLDP configuration enables LLDP on all the three management interfaces.

```
RP/0/RP0/CPU0:ios#configure terminal
RP/0/RP0/CPU0:ios(config)#lldp management enable
RP/0/RP0/CPU0:ios(config)#lldp holdtime 30
```

```
RP/0/RP0/CPU0:ios(config)#lldp reinit 2
RP/0/RP0/CPU0:ios(config)#commit
```

Verification

You can verify the LLDP configuration using the **show running-config lldp** command.

The output of **show running-config lldp** command is as follows:

```
RP/0/RP0/CPU0:ios#show running-config lldp
Tue Dec 10 10:36:11.567 UTC
lldp
timer 30
reinit 2
holdtime 120
management enable
!
```

You can verify the LLDP data using the **show lldp interface** and **show lldp neighbors** commands.

The output of **show lldp interface** command is as follows:

```
RP/0/RP0/CPU0:ios#show lldp interface
Mon Nov 11 14:33:58.982 IST


MgmtEth0/RP0/CPU0/0:
        Tx: enabled
        Rx: enabled
        Tx state: IDLE
        Rx state: WAIT FOR FRAME


MgmtEth0/RP0/CPU0/2:
        Tx: enabled
        Rx: enabled
        Tx state: IDLE
        Rx state: WAIT FOR FRAME


GigabitEthernet0/0/0/0:
        Tx: enabled
        Rx: enabled
        Tx state: IDLE
        Rx state: WAIT FOR FRAME
```

The output of **show lldp neighbors** command is as follows:

```
RP/0/RP0/CPU0ios:M-131#show lldp neighbors
Mon Dec 9 14:57:55.915 IST
Capability codes:
(R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
(W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID Local Intf Hold-time Capability Port ID
P1C_DT_01.cisco.com GigabitEthernet0/0/0/0 120 R GigabitEthernet0/0/0/0
NCS1004-HH-10 MgmtEth0/RP0/CPU0/2 60 R MgmtEth0/RP0/CPU0/2

Total entries displayed: 2
```

where [DISABLED] shows that the LLDP is disabled on the interface MgmtEth0/RP0/CPU0/0.

Enabling LLDP per Management Interface

The following example shows the commands to configure LLDP at the management interface level.

```
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/X
RP/0/RP0/CPU0:ios(config-if)#lldp enable
RP/0/RP0/CPU0:ios(config-if)#commit
```

Disabling LLDP Transmit and Receive Operations

The following example shows the commands to disable the LLDP transmit operations at the specified management interface.

```
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/X
RP/0/RP0/CPU0:ios(config-if)#lldp transmit disable
RP/0/RP0/CPU0:ios(config-if)#commit
```

The following example shows the commands to disable the LLDP receive operations at the specified management interface.

```
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/X
RP/0/RP0/CPU0:ios(config-if)#lldp receive disable
RP/0/RP0/CPU0:ios(config-if)#commit
```

Debugging LLDP Issues

The following commands are used for debugging issues in the LLDP functionality.

- **show lldp traffic**
- **debug lldp all**
- **debug lldp errors**
- **debug lldp events**
- **debug lldp packets**
- **debug lldp tlvs**
- **debug lldp trace**
- **debug lldp verbose**

Configure Telnet

This procedure allows you to establish a telnet session to the management interface using its IP address. Use this procedure when NCS 1010 chassis is not booted using ZTP.

Before you begin

Ensure that two xr-telnet-* rpms are installed. .

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:ios#configure
```

Enters the configuration mode.

Step 2 **telnet {ipv4 | ipv6} server max-servers** *limit*

Example:

```
RP/0/RP0/CPU0:ios(config)#telnet ipv4 server max-servers 10
```

Specifies the number of allowable telnet servers (up to 100). By default, telnet servers are not allowed. You must configure this command to enable the use of telnet servers.

Step 3 Use the **commit** or **end** command.

commit-Saves the configuration changes and remains within the configuration session.

end-Prompts user to take one of these actions:

- **Yes**-Saves configuration changes and exits the configuration session.
- **No**-Exits the configuration session without committing the configuration changes.
- **Cancel**-Remains in the configuration session without committing the configuration changes.

Configure SSH

This procedure allows you to establish an SSH session to the management interface using its IP address. Use this procedure when NCS 1010 chassis is not booted using ZTP.

Before you begin

- Generate the crypto key for SSH using the **crypto key generate dsa** command.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:ios#configure
```

Enters the configuration mode.

Step 2 **ssh server v2**

Example:

```
RP/0/RP0/CPU0:ios(config)# ssh server v2
```

Enables the SSH server to accept only SSHv2 client connections.

Step 3 Use the **commit** or **end** command.

commit-Saves the configuration changes and remains within the configuration session.

end-Prompts the user to take one of these actions:

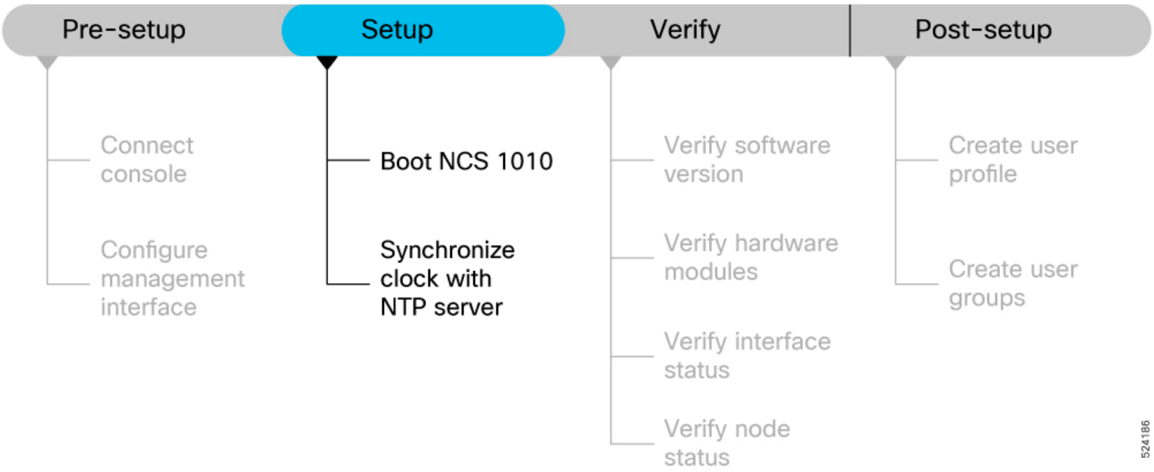
- **Yes**-Saves configuration changes and exits the configuration session.
- **No**-Exits the configuration session without committing the configuration changes.

- **Cancel**-Remains in the configuration session without committing the configuration changes.

Setup NCS 1010

Complete the following tasks to bring up your NCS 1010 for further configurations.

Figure 6: Setup Workflow for the Cisco NCS 1010



Boot NCS 1010

Use the console port to connect to NCS 1010. By default, the console port connects to the XR mode. If necessary, you can establish subsequent connections through the management port, after it is configured.

Procedure

Step 1 Connect a terminal to the console port of the RP.

Step 2 Start the terminal emulation program on your workstation.

The console settings are 9600 bps, 8 data bits, 1 stop bit and no parity.

Step 3 Power on NCS 1010.

To power on the shelves, install the AC or DC power supplies and cables. As NCS 1010 boots up, you can view the boot process details at the console of the terminal emulation program.

Step 4 Press **Enter**.

The boot process is complete when the system prompts you to enter the root-system username. If the prompt does not appear, wait for a while to give NCS 1010 more time to complete the initial boot procedure; then press **Enter**.

Important

If the boot process fails, it may be because the preinstalled image on the NCS 1010 is corrupt. In this case, you can boot NCS 1010 using an external bootable USB drive.

Boot NCS 1010 Using USB Drive

The bootable USB drive is used to reimage NCS 1010 for system upgrade or to boot the NCS 1010 in case of boot failure. A bootable USB drive is created by copying a compressed boot file into a USB drive. The USB drive becomes bootable after the contents of the compressed file are extracted.

You can complete this task using the Windows, Linux, or MAC operating systems available on your local machine. The exact operation to be performed for each generic step that is outlined here depends on the operating system in use.

Use this task to boot the NCS 1010 using the USB drive.

Before you begin

- You need a USB drive with a storage capacity of at least 4 GB.
- The USB drive should have a single partition.
- NCS 1010 software image can be downloaded from Software Download page on Cisco.com.
- Copy the compressed boot file from the software download page at Cisco.com to your local machine. The filename for the compressed boot file is in the format *ncs1010-usb-boot-<release_number>.zip*.

Procedure

Step 1 Connect the USB drive to your local machine and format it with the FAT32 file system.

Step 2 Copy the compressed boot file to the USB drive.

Step 3 Verify that the copy operation is successful. To verify, compare the file size at source and destination. Also, verify the MD5 checksum value.

Step 4 Extract the content of the compressed boot file by unzipping it in the USB drive. This makes the USB drive a bootable drive.

Note
You must extract the contents of the zipped file ("EFI" and "boot" directories) directly in the root folder of the USB drive. If the unzipping application places the extracted files in a new folder, move the "EFI" and "boot" directories to the root folder of the USB drive.

Step 5 Insert the USB drive in one of the USB ports of NCS 1010 line card/controller card.

Step 6 Reboot NCS 1010 using power cycle or console.

Note
Use the **reload bootmedia usb noprompt** command to boot the NCS 1010 from the USB. If you are using the **reload bootmedia usb noprompt** command, then you can skip the remaining steps.

Step 7 Press **Esc** to enter BIOS.

Step 8 Select the **Save & Exit** tab of BIOS.

Step 9 Choose **IOS -XR Install**.

The BIOS UI displays the USB drive vendor in the brackets, in this case, SMART USB 1084.

The system detects USB and boots the image from USB.

```
Booting from USB..
Loading Kernel..
Verifying (loop)/boot/bzImage...
(loop)/boot/bzImage verified using attached signature.
Loading initrd..
Verifying (loop)/boot/initrd.img...
```

Step 10 Remove the USB drive after the `Rebooting the system after installation` message is displayed. The NCS 1010 reboots automatically.

Note

The USB must be removed only after the image is loaded successfully.

Understand NTP

Table 3: Feature History

Feature Name	Release Information	Feature Description
NTP Support	Cisco IOS XR Release 7.11.1	Network Time Protocol (NTP) allows devices to synchronize clocks with the NTP servers, maintaining the most accurate time. NCS 1010 now supports time synchronization. In modern and large networks, time synchronization is critical because every aspect of managing, securing, planning, and debugging a network depends on the time of occurrence of events. Commands added: • **ntp server** • **show ntp associations** • **show ntp status**

NTP uses the User Datagram Protocol (UDP) as its transport protocol. All NTP communication uses Coordinated Universal Time (UTC). An NTP network usually receives its time from an authoritative time source, such as a radio clock or an atomic clock attached to a time server. NTP distributes this time across the network.

NTP uses the concept of a “stratum” to describe how many NTP hops away a machine is from an authoritative time source. A “stratum 1” time server typically has an authoritative time source (such as a radio or atomic clock, or a GPS time source) directly attached, a “stratum 2” time server receives its time through NTP from a “stratum 1” time server, and so on.

The communications between machines running NTP (known as associations) are usually statically configured; each machine is given the IP address of all machines with which it should form associations. Accurate timekeeping is made possible by exchanging NTP messages between each pair of machines with an association.

An NTP broadcast client listens for broadcast messages sent by an NTP broadcast server at a designated IPv4 address. The client synchronizes the local clock using the first received broadcast message.

Synchronize Clock with NTP Server

There is an independent system clock for IOS XR. To ensure that this clock does not deviate from true time, it must be synchronized with the clock of an NTP server.

Before you begin

- Configure Management Interface

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:ios#configure
```

Enters the configuration mode.

Step 2 **ntp**

Example:

```
RP/0/RP0/CPU0:ios(config)#ntp
```

Enters NTP configuration mode.

Step 3 **server** [**ipv4**| **ipv6**] *ntp-server-ip-address* [**version** *version-number*][**key** *key-id*][**minpoll** *interval*][**maxpoll** *interval*][**source** *type interface-path-id*][**prefer**][**burst**][**iburst**]

Example:

IPv4:

```
RP/0/RP0/CPU0:ios(config-ntp)#server 198.51.100.1 version 4 prefer iburst
```

IPv6:

```
RP/0/RP0/CPU0:ios(config-ntp)#server 2001:DB8::1 version 4 prefer iburst
```

Synchronizes the console clock with the specified NTP server.

Note
The NTP server can also be reached through a VRF if the management interface is in a VRF.

Step 4 Use one of the following commands:

- **end**

- **commit**

Example:

```
RP/0/RP0/CPU0:ios(config-ntp)#end
```

or

```
RP/0/RP0/CPU0:ncs1010(config-ntp)#commit
```

Saves configuration changes.

- When you issue the **end** command, the system prompts you to commit changes:

  ```
  Uncommitted changes found, commit them before
    exiting(yes/no/cancel)?
  [cancel]:
  ```

  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns to EXEC mode.
  - Entering **no** exits the configuration session and returns to EXEC mode without committing the configuration changes.
  - Entering **cancel** leaves the system in the current configuration session without exiting or committing the configuration changes.

- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 5 **show running-config ntp**

Example:

```
RP/0/RP0/CPU0:ios#show running-config ntp

Sun Nov  5 15:14:24.969 UTC

ntp

 server 4.33.0.51 burst iburst

!
```

Displays the running configuration.

Verify the Status of the External Reference Clock

This task explains how to verify the status of NTP components.

Note The commands can be entered in any order.

Procedure

Step 1 **show ntp associations [detail] [location** ***node-id*****]**

Example:

```
RP/0/RP0/CPU0:ios#show ntp associations
Sun Nov 5 15:14:44.128 UTC

address ref clock st when poll reach delay offset disp
*~192.0.2.1 198.51.100.1 2 81 128 377 1.84 7.802 2.129
* sys_peer, # selected, + candidate, - outlayer, x falseticker, ~ configured
```

Displays the status of NTP associations.

Example:

```
RP/0/RP0/CPU0:ios#show ntp associations detail
Sun Nov 5 15:14:48.763 UTC

192.0.2.1 configured, our_master, stratum 2
ref ID 198.51.100.1, time E8F22BB9.79D4A841 (14:56:57.475 UTC Sun Nov 5 2023)
our mode client, peer mode server, our poll intvl 128, peer poll intvl 128
root delay 0.6866 msec, root disp 1.04, reach 377, sync dist 6.2590
delay 1.84 msec, offset 7.802 msec, dispersion 2.129
precision 2**23, version 4
org time E8F22F92.B647E8FC (15:13:22.712 UTC Sun Nov 5 2023)
rcv time E8F22F92.B88F303C (15:13:22.720 UTC Sun Nov 5 2023)
xmt time E8F22F92.B88F303C (15:13:22.720 UTC Sun Nov 5 2023)
filtdelay = 1.844 1.772 1.983 1.954 1.945 2.000 1.902 1.778
filtoffset = 7.857 7.802 8.065 8.063 8.332 8.397 8.664 8.684
filterror = 0.000 0.060 1.995 2.055 4.050 4.110 6.060 6.120
```

Example:

```
RP/0/RP0/CPU0:ios#show ntp associations detail location 0/RP0/CPU0
Sun Nov 5 15:38:15.744 UTC

192.0.2.1 configured, our_master, stratum 2
ref ID 198.51.100.1, time E8F233C0.5606A159 (15:31:12.336 UTC Sun Nov 5 2023)
our mode client, peer mode server, our poll intvl 128, peer poll intvl 128
root delay 0.7019 msec, root disp 0.47, reach 377, sync dist 5.6762
delay 2.01 msec, offset 7.226 msec, dispersion 3.856
precision 2**23, version 4
org time E8F23563.DE5D42D5 (15:38:11.868 UTC Sun Nov 5 2023)
rcv time E8F23563.E07C296D (15:38:11.876 UTC Sun Nov 5 2023)
xmt time E8F23563.E07C296D (15:38:11.876 UTC Sun Nov 5 2023)
filtdelay = 2.006 1.865 1.936 1.762 1.932 1.875 1.881 2.011
filtoffset = 7.210 7.305 7.372 7.226 7.298 7.258 7.251 7.224
filterror = 0.000 2.025 2.085 4.035 4.095 6.060 6.120 8.070
```

Step 2 **show ntp status [location** ***node-id*****]**

Example:

```
RP/0/RP0/CPU0:ios#show ntp status
Sun Nov 5 15:14:36.949 UTC

Clock is synchronized, stratum 3, reference is 192.0.2.1
nominal freq is 1000000000.0000 Hz, actual freq is 44881851.3383 Hz, precision is 2**24
reference time is E8F22D7A.AB020D97 (15:04:26.668 UTC Sun Nov 5 2023)
clock offset is 9.690 msec, root delay is 2.553 msec
root dispersion is 24.15 msec, peer dispersion is 2.13 msec
```

```
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.0000212807 s/s
system poll interval is 128, last update was 610 sec ago
authenticate is disabled, panic handling is disabled,
hostname resolution retry interval is 1440 minutes.
```

Verifies that the clock is synchronized with the NTP server.

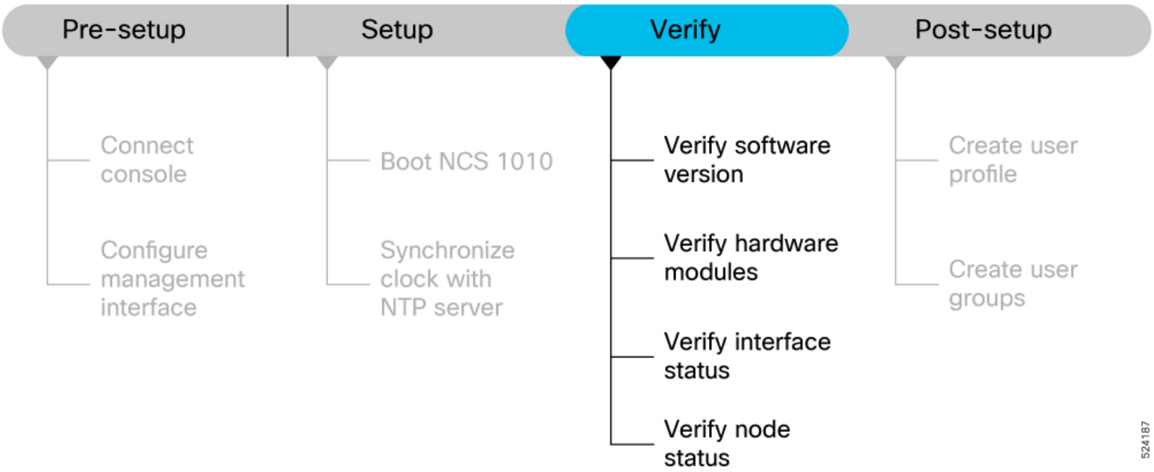
Troubleshoot NTP Issues

For NTP troublehooting information, see here.

Verify the Software and Hardware Status

After logging into the console, perform preliminary checks to verify the default setup.

Figure 7: Verification Workflow for the Cisco NCS 1010 Setup



Ensure that you have completed the procedures in Setup NCS 1010, on page 21 section before proceeding with the following verification tasks:

Note The output of the examples in the procedures is not from the latest software release. The output will change for any explicit references to the current release.

Verify Software Version

View the software version installed on the NCS 1010.

Procedure

Verify the latest version of the Cisco IOS XR software installed on the NCS 1010.

Example:

```
RP/0/RP0/CPU0:ios#show version
Sat Mar 25 11:38:23.614 IST
Cisco IOS XR Software, Version 24.3.1
Copyright (c) 2013-2023 by Cisco Systems, Inc.
Build Information:
Built By : ingunawa
Built On : Tue Mar 07 02:22:55 UTC 2023
Build Host : iox-ucs-063
Workspace : /auto/iox-ucs-063-san2/prod/203.0.113.1I.SIT_IMAGE/ncs1010/ws
Version : 24.3.1
Label : 24.3.1
cisco NCS1010 (C3758 @ 2.20GHz)
cisco NCS1010-SA (C3758 @ 2.20GHz) processor with 32GB of memory
OLT-C-R-SITE-1 uptime is 2 weeks, 12 hours, 59 minutes
NCS 1010 - Chassis
```

Note
You must upgrade the system if a new version of the system is available to avail the latest features on the NCS 1010.

For more information about upgrading the software version, see Upgrade the Software, on page 70.

The **show version** only displays the IOS XR version in the label field if modific ations are made to the running software on the booted ISO image during installation of a newer version.

Verify Hardware Modules

Cisco NCS 1010 have various hardware modules such as processors, line cards, fan trays, and power modules installed on the NCS 1010. Ensure that the firmware on various hardware components of the NCS 1010 is compatible with the installed Cisco IOS XR image. You also must verify that all the installed hardware and firmware modules are operational.

Procedure

Step 1 Verify the status of the hardware modules using the **show platform** command.

Example:

```
RP/0/RP0/CPU0:ios#show platform
Wed Apr 27 08:43:40.130 UTC
Node              Type                     State                    Config state
--------------------------------------------------------------------------------
0/RP0/CPU0        NCS1010-CNTLR-K9(Active) IOS XR RUN               NSHUT,NMON
0/PM0             NCS1010-AC-PSU           OFFLINE                  NSHUT,NMON
0/PM1             NCS1010-AC-PSU           OPERATIONAL              NSHUT,NMON
0/FT0             NCS1010-FAN              OPERATIONAL              NSHUT,NMON
0/FT1             NCS1010-FAN              OPERATIONAL              NSHUT,NMON
0/0/NXR0          NCS1K-OLT-C              OPERATIONAL              NSHUT,NMON
```

```
0/1               NCS1K-BRK-SA             OPERATIONAL              NSHUT,NMON
0/1/0             NCS1K-BRK-8              OPERATIONAL              NSHUT,NMON
0/1/1             NCS1K-BRK-8              OPERATIONAL              NSHUT,NMON
0/1/2             NCS1K-BRK-24             OPERATIONAL              NSHUT,NMON
0/1/3             NCS1K-BRK-24             OPERATIONAL              NSHUT,NMON
0/2               NCS1K-MD-32E-C           OPERATIONAL              NSHUT,NMON
0/3               NCS1K-MD-32O-C           OPERATIONAL              NSHUT,NMON
```

Step 2 View the list of hardware and firmware modules that are detected on the NCS 1010 using the **show hw-module fpd** command.

Example:

```
RP/0/RP0/CPU0:ios# show hw-module fpd

Fri Aug 30 05:59:44.248 IST

Auto-upgrade:Enabled,PM excluded
Attribute codes: B golden, P protect, S secure, A Anti Theft aware
                                                                         FPD Versions
                                                                        ==============
Location   Card type             HWver FPD device       ATR Status   Running Programd  Reload Loc
-------------------------------------------------------------------------------------------------
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  ADMConfig            CURRENT    3.40    3.40       NOT REQ
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  BIOS             S   CURRENT    4.80    4.80         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  BIOS-Golden      BS  CURRENT            4.10         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  CpuFpga          S   CURRENT    1.13    1.13         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  CpuFpgaGolden    BS  CURRENT            1.01         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  SsdMicron5300    S   CURRENT    0.01    0.01         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  TamFw            S   CURRENT    6.13    6.13         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  TamFwGolden      BS  CURRENT            6.11         0/RP0
0/PM0      NCS1010-AC-PSU        1.0   AP-PriMCU            CURRENT    1.03    1.03       NOT REQ
0/PM0      NCS1010-AC-PSU        1.0   AP-SecMCU            CURRENT    2.01    2.01       NOT REQ
0/PM1      NCS1010-AC-PSU        1.0   AP-PriMCU            CURRENT    1.03    1.03       NOT REQ
0/PM1      NCS1010-AC-PSU        1.0   AP-SecMCU            CURRENT    2.01    2.01       NOT REQ
0/0/NXR0   NCS1K-E-OLT-R-C       1.0   OLT              S   CURRENT    3.16    3.16       NOT REQ
0/0/NXR0   NCS1K-E-OLT-R-C       1.0   Raman-1          S   CURRENT    3.16    3.16       NOT REQ
0/Rack     NCS1010-SA            2.1   EITU-ADMConfig       CURRENT    2.10    2.10       NOT REQ
0/Rack     NCS1010-SA            2.1   IoFpga           S   CURRENT    1.19    1.19       NOT REQ
0/Rack     NCS1010-SA            2.1   IoFpgaGolden     BS  CURRENT            1.01       NOT REQ
0/Rack     NCS1010-SA            2.1   SsdMicron5300    S   CURRENT    0.01    0.01        0/Rack
```

From the **show hw-module fpd** output, verify that all hardware modules that are installed on the chassis are listed. An unlisted module indicates that the module is either malfunctioning, or has not been installed properly. You must remove and reinstall the hardware module.

The fields in the **show hw-module fpd** output are:

- **FPD Device**: Name of the hardware component, such as IO FPGA, or BIOS. The Golden FPDs are not field upgradable.
- **Running**: Current version of the firmware running on the FPD.
- **Programd**: Version of the FPD programmed on the module
- **Status**: Upgrade status of the firmware. The different states are:

Table 4: Status and Description of the Firmware Upgrade

Status	Description
CURRENT	The firmware version is the latest version.

READY	The firmware of the FPD is ready for an upgrade.
NOT READY	The firmware of the FPD is not ready for an upgrade.
NEED UPGD	A new firmware version is available in the installed image. We recommend that you to perform an upgrade of the firmware version.
RLOAD REQ	The upgrade is complete, and the ISO image requires a reload.
UPGD DONE	The firmware upgrade is successful.
UPGD FAIL	The firmware upgrade has failed.
BACK IMG	The firmware is corrupt. Reinstall the firmware.
UPGD SKIP	The upgrade is skipped because the installed firmware version is higher than the one available in the image.

Step 3 Upgrade the required firmware as required, using the **upgrade hw-module location all fpd all** command.

Example:

```
RP/0/RP0/CPU0:ios#upgrade hw-module location all fpd all
Alarms are created showing all modules that needs to be upgraded.

Active Alarms
----------------------------------------------------------------------------------------------------
Location     Severity  Group      Set Time                  Description
----------------------------------------------------------------------------------------------------
0/6/CPU0     Major     FPD_Infra  09/16/2019 12:34:59 UTC   One Or More FPDs Need Upgrade Or Not In
 Current State
0/10/CPU0    Major     FPD_Infra  09/16/2019 12:34:59 UTC   One Or More FPDs Need Upgrade Or Not In
 Current State
0/RP0/CPU0   Major     FPD_Infra  09/16/2019 12:34:59 UTC   One Or More FPDs Need Upgrade Or Not In
 Current State
0/RP1/CPU0   Major     FPD_Infra  09/16/2019 12:34:59 UTC   One Or More FPDs Need Upgrade Or Not In
 Current State
0/FC0        Major     FPD_Infra  09/16/2019 12:34:59 UTC   One Or More FPDs Need Upgrade Or Not In
 Current State
0/FC1        Major     FPD_Infra  09/16/2019 12:34:59 UTC   One Or More FPDs Need Upgrade Or Not In
 Current State
```

Note

The BIOS and IOFPGA upgrades require a restart of the NCS 1010 for the new version to take effect.

Step 4 Verify status of the modules after upgrade using the **show hw-module fpd** command.

Example:

```
RP/0/RP0/CPU0:ios#show hw-module fpd
                                                      REQ
Wed Jun 29 08:50:21.057 UTC

Auto-upgrade:Disabled
                                                                         FPD Versions
                                                                        ==============
Location   Card type             HWver FPD device       ATR Status    Running   Programd  Reload Loc

----------------------------------------------------------------------------------------------------
0/RP0/CPU0 NCS1010-CNTLR-K9      1.0   ADMConfig            CURRENT    3.40    3.40       NOT REQ
0/RP0/CPU0 NCS1010-CNTLR-K9      1.0   BIOS              S  CURRENT    4.10    4.10         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.0   BIOS-Golden      BS  CURRENT            4.10         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.0   CpuFpga           S  CURRENT    1.02    1.02         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.0   CpuFpgaGolden    BS  CURRENT            1.01         0/RP0
```

```
0/RP0/CPU0 NCS1010-CNTLR-K9      1.0   SsdIntelS4510     S  CURRENT   11.32   11.32         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.0   TamFw             S  CURRENT    6.13    6.13         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.0   TamFwGolden      BS  CURRENT            6.11         0/RP0
0/PM0      NCS1010-AC-PSU        0.0   AP-PriMCU            CURRENT    1.03    1.03       NOT REQ
0/PM0      NCS1010-AC-PSU        0.0   AP-SecMCU            CURRENT    2.01    2.01       NOT REQ
0/PM1      NCS1010-AC-PSU        0.0   AP-PriMCU            CURRENT    1.03    1.03       NOT REQ
0/PM1      NCS1010-AC-PSU        0.0   AP-SecMCU            CURRENT    2.01    2.01       NOT REQ
0/0/NXR0   NCS1K-ILA-C           1.0   ILA               S  CURRENT    1.00    1.00       NOT REQ
0/Rack     NCS1010-SA            1.0   EITU-ADMConfig       CURRENT    2.10    2.10       NOT REQ
0/Rack     NCS1010-SA            1.0   IoFpga            S  CURRENT    1.04    1.04       NOT REQ
0/Rack     NCS1010-SA            1.0   IoFpgaGolden     BS  CURRENT            1.01       NOT REQ
0/Rack     NCS1010-SA            1.0   SsdIntelS4510     S  CURRENT   11.32   11.32        0/Rack
```

The status of the upgraded nodes shows that a reload is required.

Step 5 Reload the individual nodes that require an upgrade.

Example:

```
RP/0/RP0/CPU0:ios#reload location node-location
```

Step 6 Verify that all nodes that had required an upgrade now shows an updated status of CURRENT with an updated FPD version.

Example:

```
Thu Mar  2 12:35:06.602 IST

Auto-upgrade:Enabled
Attribute codes: B golden, P protect, S secure, A Anti Theft aware
                                                                         FPD Versions
                                                                        ==============
Location   Card type             HWver FPD device       ATR Status   Running Programd  Reload Loc
-------------------------------------------------------------------------------------------------
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  ADMConfig            CURRENT    3.40    3.40       NOT REQ
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  BIOS             S   CURRENT    4.20    4.20         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  BIOS-Golden      BS  CURRENT            4.10         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  CpuFpga          S   CURRENT    1.11    1.11         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  CpuFpgaGolden    BS  CURRENT            1.01         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  SsdIntelS4510    S   CURRENT   11.32   11.32         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  TamFw            S   CURRENT    6.13    6.13         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  TamFwGolden      BS  CURRENT            6.11         0/RP0
0/PM0      NCS1010-AC-PSU        0.0   AP-PriMCU            CURRENT    1.03    1.03       NOT REQ
0/PM0      NCS1010-AC-PSU        0.0   AP-SecMCU            CURRENT    2.01    2.01       NOT REQ
0/PM1      NCS1010-AC-PSU        0.0   AP-PriMCU            CURRENT    1.03    1.03       NOT REQ
0/PM1      NCS1010-AC-PSU        0.0   AP-SecMCU            CURRENT    2.01    2.01       NOT REQ
0/0/NXR0   NCS1K-OLT-L           1.0   OLT              S   CURRENT    1.02    1.02       NOT REQ
0/Rack     NCS1010-SA            2.1   EITU-ADMConfig       CURRENT    2.10    2.10       NOT REQ
0/Rack     NCS1010-SA            2.1   IoFpga           S   CURRENT    1.12    1.12       NOT REQ
0/Rack     NCS1010-SA            2.1   IoFpgaGolden     BS  CURRENT            1.01       NOT REQ
0/Rack     NCS1010-SA            2.1   SsdIntelS4510    S   CURRENT   11.32   11.32        0/Rack
```

Verify Interface Status

All available interfaces must be discovered by the system after booting the Cisco NCS 1010. Interfaces not discovered might indicate a malfunction in the unit.

Procedure

Use the **show ipv4 interfaces brief** or **show ipv6 interfaces brief** command to view the interfaces discovered by the system.

Example:

```
RP/0/RP0/CPU0:ios#show ipv4 interfaces brief
Wed May 25 11:50:28.438 UTC

               Intf     Intf        LineP              Encap          MTU        BW
               Name    State       State               Type        (byte)    (Kbps)
----------------------------------------------------------------------------------------------------

                Lo0       up          up             Loopback       1500          0
                Lo3       up          up             Loopback       1500          0
                Nu0       up          up              Null           1500          0
          Gi0/0/0/0       up          up              ARPA           1514    1000000
         Mg0/RP0/CPU0/0    up          up              ARPA          1514    1000000
        Mg0/RP0/CPU0/1    admin-down  admin-down        ARPA  1514    1000000
        Mg0/RP0/CPU0/2    admin-down  admin-down        ARPA  1514    1000000
        PT0/RP0/CPU0/0    admin-down  admin-down        ARPA  1514    1000000
        PT0/RP0/CPU0/1    admin-down  admin-down        ARPA  1514    1000000
```

Example:

```
RP/0/RP0/CPU0:ios#show ipv4 interfaces brief
Tue Jul 12 07:32:42.390 UTC

Interface                      IP-Address      Status          Protocol Vrf-Name
Loopback0                      198.51.100.1       Up             Up       default
Loopback3                      203.0.113.1        Up             Up       default
GigabitEthernet0/0/0/0         192.0.2.1          Up             Up       default
MgmtEth0/RP0/CPU0/0            192.0.2.255        Up             Up       default
PTP0/RP0/CPU0/0                unassigned      Shutdown        Down     default
MgmtEth0/RP0/CPU0/1            unassigned      Down            Down     default
PTP0/RP0/CPU0/1                unassigned      Shutdown        Down     default
MgmtEth0/RP0/CPU0/2            unassigned      Down            Down     default
```

When a NCS 1010 is turned ON for the first time, all interfaces are in the **unassigned** state.

Ensure that the total number of interfaces that are displayed in the result matches with the actual number of interfaces present on the NCS 1010, and that the interfaces are created according to the type of line cards displayed in **show platform** command.

Verify Node Status

A node can be a specified location, or the complete hardware module in the system. You must verify that the software state of all route processors, line cards, and the hardware state of fabric cards, fan trays, and power modules are listed, and their state is OPERATIONAL. This indicates that the IOS XR console is operational on the cards.

Procedure

Verify the operational status of the node using the **show platform** command.

Example:

```
RP/0/RP0/CPU0:ios#show platform
Wed Apr 27 08:43:40.130 UTC
Node              Type                     State                    Config state
--------------------------------------------------------------------------------
0/RP0/CPU0        NCS1010-CNTLR-K9(Active) IOS XR RUN               NSHUT,NMON
0/PM0             NCS1010-AC-PSU           OFFLINE                  NSHUT,NMON
0/PM1             NCS1010-AC-PSU           OPERATIONAL              NSHUT,NMON
0/FT0             NCS1010-FAN              OPERATIONAL              NSHUT,NMON
0/FT1             NCS1010-FAN              OPERATIONAL              NSHUT,NMON
0/0/NXR0          NCS1K-OLT-C              OPERATIONAL              NSHUT,NMON
0/1               NCS1K-BRK-SA             OPERATIONAL              NSHUT,NMON
0/1/0             NCS1K-BRK-8              OPERATIONAL              NSHUT,NMON
0/1/1             NCS1K-BRK-8              OPERATIONAL              NSHUT,NMON
0/1/2             NCS1K-BRK-24             OPERATIONAL              NSHUT,NMON
0/1/3             NCS1K-BRK-24             OPERATIONAL              NSHUT,NMON
0/2               NCS1K-MD-32E-C           OPERATIONAL              NSHUT,NMON
0/3               NCS1K-MD-32O-C           OPERATIONAL              NSHUT,NMON
```

Example:

```
RP/0/RP0/CPU0:ios#show platform
Thu Mar  2 12:35:01.883 IST
Node              Type                     State                    Config state
--------------------------------------------------------------------------------
0/RP0/CPU0        NCS1010-CNTLR-K9(Active) IOS XR RUN               NSHUT,NMON
0/PM0             NCS1010-AC-PSU           OPERATIONAL              NSHUT,NMON
0/PM1             NCS1010-AC-PSU           OFFLINE                  NSHUT,NMON
0/FT0             NCS1010-FAN              OPERATIONAL              NSHUT,NMON
0/FT1             NCS1010-FAN              OPERATIONAL              NSHUT,NMON
0/0/NXR0          NCS1K-OLT-L              OPERATIONAL              NSHUT,NMON
0/3               NCS1K-BRK-24             OPERATIONAL              NSHUT,NMON
```

What to do next

This completes verification of the basic NCS 1010 setup. You can now complete the post-setup tasks where you manage user profiles and groups.

Verify Inventory

The **show inventory** command displays details of the hardware inventory of NCS 1010.

To verify the inventory information for all the physical entities, perform the following procedure.

Procedure

show inventory

Displays the details of the physical entities of NCS 1010 along with the details of SFPs.

Example:

```
RP/0/RP0/CPU0:ios#show inventory
Wed Apr 27 08:43:44.222 UTC

NAME: "Rack 0", DESCR: "NCS1010 - Shelf Assembly"
PID: NCS1010-SA        , VID: V00, SN: FCB2504B0X4

NAME: "0/RP0/CPU0", DESCR: "Network Convergence System 1010 Controller"
PID: NCS1010-CNTLR-K9  , VID: V00, SN: FCB2506B0NX

NAME: "0/1", DESCR: "NCS 1000 shelf for 4 passive modules"
PID: NCS1K-BRK-SA      , VID: V00 , SN: FCB2534B0GR

NAME: "0/1/0", DESCR: "NCS 1000 MTP/MPO to 8 port passive breakout module"
PID: NCS1K-BRK-8       , VID: V00 , SN: MPM25401005

NAME: "0/1/1", DESCR: "NCS 1000 MTP/MPO to 8 port passive breakout module"
PID: NCS1K-BRK-8       , VID: V00 , SN: MPM25401003

NAME: "0/1/2", DESCR: "NCS 1000 MTP/MPO to 24 colorless chs passive breakout module"
PID: NCS1K-BRK-24      , VID: V00 , SN: MPM25141004

NAME: "0/1/3", DESCR: "NCS 1000 MTP/MPO to 24 colorless chs passive breakout module"
PID: NCS1K-BRK-24      , VID: V00 , SN: MPM25371005

NAME: "0/2", DESCR: "NCS 1000 32 chs Even Mux/Demux Patch Panel - 150GHz - C-band"
PID: NCS1K-MD-32E-C    , VID: V00 , SN: ACW2529YE13

NAME: "0/3", DESCR: "NCS 1000 32 chs Odd  Mux/Demux Patch Panel - 150GHz - C-band"
PID: NCS1K-MD-32O-C    , VID: V00 , SN: ACW2529YA13

NAME: "0/FT0", DESCR: "NCS1010 - Shelf Fan"
PID: NCS1010-FAN       , VID: V00, SN: FCB2504B0W3

NAME: "0/FT1", DESCR: "NCS1010 - Shelf Fan"
PID: NCS1010-FAN       , VID: V00, SN: FCB2504B0U8

NAME: "0/PM0", DESCR: "NCS 1010 - AC Power Supply Unit"
PID: NCS1010-AC-PSU    , VID: V00, SN: APS244700D0

NAME: "0/PM1", DESCR: "NCS 1010 - AC Power Supply Unit"
PID: NCS1010-AC-PSU    , VID: V00, SN: APS244700BY
```

Verify Management Interface Status

To verify the management interface status, perform the following procedure.

Procedure

Step 1 **show interfaces MgmtEth 0/RP0/CPU0/0**

Displays the management interface configuration.

Example:

```
RP/0/RP0/CPU0:ios#show interfaces MgmtEth 0/RP0/CPU0/0
Wed May 25 11:49:18.118 UTC
```

```
MgmtEth0/RP0/CPU0/0 is up, line protocol is up
  Interface state transitions: 1
  Hardware is Management Ethernet, address is 38fd.f866.0964 (bia 38fd.f866.0964)
  Internet address is 192.0.2.254/16
  MTU 1514 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 1000Mb/s, CX, link type is autonegotiation
  loopback not set,
  Last link flapped 15:05:21
  ARP type ARPA, ARP timeout 04:00:00
  Last input never, output 00:00:00
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     53138 packets input, 6636701 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 12145 broadcast packets, 40082 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     217288 packets output, 60964220 bytes, 0 total output drops
     Output 1 broadcast packets, 15 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions
```

Step 2 **show interfaces summary** and **show interfaces brief**

Verifies the management interface status.

Example:

```
RP/0/RP0/CPU0:ios#show interfaces summary
Mon Nov  4 18:10:14.996 IST
Interface Type          Total    UP       Down     Admin Down
--------------          -----    --       ----     ----------
ALL TYPES               9        7        0        2
--------------
IFT_GETHERNET           1        1        0        0
IFT_LOOPBACK            1        1        0        0
IFT_ETHERNET            4        4        0        0
IFT_NULL                1        1        0        0
IFT_PTP_ETHERNET        2        0        0        2
```

Example:

```
RP/0/RP0/CPU0:ios#show interfaces brief
Mon Nov  4 18:11:37.222 IST

               Intf       Intf        LineP              Encap  MTU        BW
               Name       State       State               Type (byte)    (Kbps)
--------------------------------------------------------------------------------
                Lo0          up          up          Loopback  1500           0
                Nu0          up          up              Null  1500           0
          Gi0/0/0/0          up          up              ARPA  1514      100000
     Mg0/RP0/CPU0/0          up          up              ARPA  1514     1000000
     Mg0/RP0/CPU0/1          up          up              ARPA  1514     1000000
     Mg0/RP0/CPU0/2          up          up              ARPA  1514     1000000
     PT0/RP0/CPU0/0  admin-down  admin-down              ARPA  1514     1000000
     PT0/RP0/CPU0/1  admin-down  admin-down              ARPA  1514     1000000
    Mg0/RP0/RCOM0/0          up          up              ARPA  1514     1000000
```

Example:

```
RP/0/RP0/CPU0:ios#show ipv4 interfaces brief
Mon Nov  4 18:12:32.082 IST
```

```
Interface                      IP-Address      Status          Protocol Vrf-Name
Loopback0                      192.0.2.1        Up              Up       default
GigabitEthernet0/0/0/0         192.0.2.1        Up              Up       default
MgmtEth0/RP0/CPU0/0            192.0.2.254      Up              Up       default
PTP0/RP0/CPU0/0                unassigned      Shutdown        Down      default
MgmtEth0/RP0/CPU0/1            203.0.113.1      Up              Up       default
PTP0/RP0/CPU0/1                unassigned      Shutdown        Down      default
MgmtEth0/RP0/CPU0/2            192.0.2.255      Up              Up       default
MgmtEth0/RP0/RCOM0/0           unassigned       Up              Up       default
```

Verify Alarms

You can view the alarm information using the **show alarms** command.

Procedure

show alarms [**brief** [**card | rack | system**] [**location** *location*] [**active | history**] | **detail** [**card | rack | system**] [**location** *location*] [**active | clients | history | stats**]]

Displays alarms in brief or detail.

Example:

```
RP/0/RP0/CPU0:ios#show alarms brief system active


Thu Apr 28 06:16:50.524 UTC

------------------------------------------------------------------------------------
Active Alarms
------------------------------------------------------------------------------------
Location        Severity     Group            Set Time                Description
------------------------------------------------------------------------------------
0/RP0/CPU0      Major        Ethernet         04/28/2022 06:03:39 UTC    RP-SW: SPI flash config is incorrect
0/PM0           Major        Environ          04/28/2022 06:03:50 UTC    Power Module Error (PM_VIN_VOLT_OOR)
0/PM0           Major        Environ          04/28/2022 06:03:50 UTC    Power Module Output Disabled
(PM_OUTPUT_DISABLED)
0               Major        Environ          04/28/2022 06:03:50 UTC    Power Group redundancy lost
0/PM0           Major        FPD_Infra        04/28/2022 06:04:08 UTC    One Or More FPDs Need Upgrade Or Not In
Current State
0/PM1           Major        FPD_Infra        04/28/2022 06:04:09 UTC    One Or More FPDs Need Upgrade Or Not In
Current State
0/0             Major        Controller       04/28/2022 06:05:12 UTC    Osc0/0/0/0 - Provisioning Failed
0/0             Major        Controller       04/28/2022 06:05:12 UTC    Osc0/0/0/2 - Provisioning Failed
0/0             Major        Controller       04/28/2022 06:05:12 UTC    Ots0/0/0/0 - Provisioning Failed
0/0             Major        Controller       04/28/2022 06:05:12 UTC    Ots0/0/0/2 - Provisioning Failed
```

Note

In the maintenance mode, all the alarms are moved from active to suppressed and the **show alarms** command does not display the alarms details.

Verify Environmental Parameters

The **show environment** command displays the environmental parameters of NCS 1010.

To verify that the environmental parameters are as expected, perform the following procedure.

Procedure

show environment [**alarm-contact** | **all** | **altitude** | **current** | **fan** | **humidity** | **power** | **voltages** [**location** | *location*] | **temperature** [**location** | *location*]]

Displays the environmental parameters of NCS 1010.

Example:

The following example shows a sample output of the **show environment** command with the **fan** keyword.

```
RP/0/RP0/CPU0:ios#show environment fan
Thu May 26 04:15:37.765 UTC
===========================================================================
                                              Fan speed (rpm)
Location      FRU Type                        FAN_0    FAN_1   FAN_2
---------------------------------------------------------------------------

0/PM0        NCS1010-AC-PSU                    5368
0/PM1        NCS1010-AC-PSU                    5336
0/FT0        NCS1010-FAN                      10020   10020   10020
0/FT1        NCS1010-FAN                      10020   10020    9960

==============================================================================
```

The following example shows a sample output of the **show environment** command with the **temperatures** keyword for *0/RP0 location*.

```
RP/0/RP0/CPU0:ios#show environment temperature location 0/RP0

Thu May 26 04:16:39.832 UTC
==============================================================================================================
Location  TEMPERATURE                          Value     Crit     Major     Minor     Minor     Major     Crit
          Sensor                             (deg C)     (Lo)      (Lo)      (Lo)      (Hi)      (Hi)     (Hi)
--------------------------------------------------------------------------------------------------------------
0/RP0/CPU0
          RP_TEMP_PCB                             30      -10        -5         0        70        75       80
          RP_TEMP_HOT_SPOT                        33      -10        -5         0        70        75       80
          RP_TEMP_LTM4638                         49      -10        -5         0        80        85       90
          RP_TEMP_LTM4644_0                       36      -10        -5         0        80        85       90
          RP_TEMP_LTM4644_1                       39      -10        -5         0        80        85       90
          RP_JMAC_1V0_VCCP_TMON                   33      -10        -5         0        80        85       90
          RP_JMAC_1V0_VNN_TMON                    33      -10        -5         0        80        85       90
          RP_JMAC_1V0_VCC_RAM_TMON                32      -10        -5         0        80        85       90
          RP_JMAC_1V2_DDR_VDDQ_TMON               33      -10        -5         0        80        85       90
===============================================================================================================
```

The following example shows a sample output of the **show environment** command with the **temperatures** keyword for *0/0/NXR0 location*.

```
RP/0/RP0/CPU0:ios#show environment temperature location 0/0/NXR0

Thu May 26 04:16:39.832 UTC
==============================================================================================================
Location  TEMPERATURE                          Value     Crit     Major     Minor     Minor     Major     Crit
          Sensor                             (deg C)     (Lo)      (Lo)      (Lo)      (Hi)      (Hi)     (Hi)
--------------------------------------------------------------------------------------------------------------
0/0/NXR0
          OLTC_LT_P0_iEDFA0                       24       18        19        20        30        31       32
          OLTC_LT_P0_iEDFA1                       25       18        19        20        30        31       32
          OLTC_LT_P0_iEDFA2                       24       18        19        20        30        31       32
          OLTC_LT_P2_iEDFA0                       25       18        19        20        30        31       32
          OLTC_LT_P3_iEDFA0                      25       18       19       20       30       31       32
          OLTC_LT_P0_eEDFA0                      24       18       19       20       30       31       32
          OLTC_CT_1                              32      -10       -7       -5       75       77       80
          OLTC_LT_P0_eEDFA1                      24       18       19       20       30       31       32
          OLTC_CT_2                              27      -10       -7       -5       70       73       75
          OLTC_CT_3                              30      -10       -7       -5       70       73       75
          OLTC_CT_4                              30      -10       -7       -5       70       73       75
          OLTC_FT_P0_iEDFA0                      60       55       57       58       62       64       65
          OLTC_FT_P2_iEDFA0                      60       55       57       58       62       64       65
          OLTC_FT_P3_iEDFA0                      60       55       57       58       62       64       65
          OLTC_FT_P0_eEDFA0                      60       55       57       58       62       64       65
==============================================================================================================
```

The following example shows a sample output of the **show environment** command with the **power** keyword.

```
RP/0/RP0/CPU0:ios#show environment power
Thu May 26 04:17:55.592 UTC
================================================================================
CHASSIS LEVEL POWER INFO: 0
================================================================================
   Total output power capacity (Group 0 + Group 1) :    1050W +     1050W
   Total output power required                     :     700W
   Total power input                               :     228W
   Total power output                              :     140W

Power Group 0:
================================================================================
   Power       Supply         ------Input----   ------Output---     Status
   Module      Type            Volts     Amps    Volts     Amps
================================================================================
   0/PM0       NCS1010-AC-PSU  228.5     0.5     12.1      5.6      OK

Total of Group 0:              114W/0.5A         67W/5.6A

Power Group 1:
================================================================================
   Power       Supply         ------Input----   ------Output---     Status
   Module      Type            Volts     Amps    Volts     Amps
================================================================================
   0/PM1       NCS1010-AC-PSU  228.5     0.5     12.1      6.1      OK

Total of Group 1:              114W/0.5A         73W/6.1A

================================================================================
   Location     Card Type               Power       Power        Status
                                        Allocated   Used
                                        Watts       Watts
================================================================================
   0/RP0/CPU0   NCS1010-CNTLR-K9        90          14           ON
   0/FT0        NCS1010-FAN             110         17           ON
   0/FT1        NCS1010-FAN             110         15           ON
   0/0/NXR0     NCS1K-OLT-C             350         61           ON
   0/Rack       NCS1010-SA              40          19           ON


=================================================================================
```

The following example shows a sample output of the **show environment** command with the **voltages** keyword.

```
RP/0/RP0/CPU0:ios#show environment voltage location 0/RP0
Thu May 26 04:19:16.636 UTC
========================================================================================
Location  VOLTAGE                             Value      Crit    Minor    Minor      Crit
          Sensor                              (mV)       (Lo)    (Lo)     (Hi)       (Hi)
----------------------------------------------------------------------------------------
0/RP0/CPU0
          RP_ADM1266_12V0                     12094    10800    11280    12720    13200
          RP_ADM1266_1V8_CPU                   1806     1670     1750     1850     1930
          RP_ADM1266_1V24_VCCREF               1238     1150     1200     1280     1330
          RP_ADM1266_1V05_CPU                  1047      980     1020     1080     1120
          RP_ADM1266_1V2_DDR_VDDQ              1204     1120     1160     1240     1280
          RP_ADM1266_1V0_VCC_RAM                988      650      700     1250     1300
          RP_ADM1266_1V0_VNN                    869      550      600     1250     1300
          RP_ADM1266_1V0_VCCP                  1018      450      500     1250     1300
          RP_ADM1266_0V6_DDR_VTT                599      560      580      620      640
          RP_ADM1266_3V3_STAND_BY              3301     3070     3200     3400     3530
          RP_ADM1266_5V0                       5004     4650     4850     5150     5350
          RP_ADM1266_3V3                       3325     3070     3200     3400     3530
          RP_ADM1266_2V5_PLL                   2489     2330     2430     2580     2680
          RP_ADM1266_2V5_FPGA                  2502     2330     2430     2580     2680
          RP_ADM1266_1V2_FPGA                  1202     1120     1160     1240     1280
          RP_ADM1266_3V3_CPU                   3332     3070     3200     3400     3530
          RP_ADM1266_2V5_CPU                   2498     2330     2430     2580     2680

=========================================================================================
```

The following example shows a sample output of the **show environment** command with the **current** keyword.

```
RP/0/RP0/CPU0:P2C_DT_02#show environment current
Tue Jul  5 08:36:22.132 UTC
==========================================================================================
Location  CURRENT                              Value
          Sensor                               (mA)
------------------------------------------------------------------------------------------
0/RP0/CPU0
          RP_CURRMON_LTM4638                    395
          RP_CURRMON_LTM4644_0                  179
          RP_CURRMON_LTM4644_1                  307
          RP_JMAC_1V0_VCCP_IMON                 187
          RP_JMAC_1V0_VNN_IMON                   62
          RP_JMAC_1V0_VCC_RAM_IMON                0
          RP_JMAC_1V2_DDR_VDDQ_IMON             187
0/Rack
          SA_ADM1275_12V_MOD0_IMON             4154
          SA_ADM1275_12V_MOD1_IMON               43
          SA_ADM1275_12V_MOD2_IMON               18
          SA_ADM1275_12V_FAN0_IMON             1356
          SA_ADM1275_12V_FAN1_IMON             1517
          SA_INA230_5V0_IMON                    129
          SA_INA230_3V3_IMON                   2998
          SA_INA230_1V0_XGE_CORE_IMON          2464
          SA_INA230_1V0_FPGA_CORE_IMON          787
          SA_ADM1275_12V_SA_IMON               1668
          SA_ADM1275_12V_CPU_IMON              1147
```

The following example shows a sample output of the **show environment** command with the **altitude** keyword.

```
RP/0/RP0/CPU0:P2C_DT_02#show environment altitude
Tue Jul  5 08:36:51.710 UTC
================================================================================
Location     Altitude Value (Meters)    Source
--------------------------------------------------------------------------------
0            760                    sensor
```

The following example shows a sample output of the **show environment** command with the **all** keyword.

```
RP/0/RP0/CPU0:P2C_DT_02#show environment all
```

```
Tue Jul  5 08:37:28.412 UTC
==============================================================================================================
Location  TEMPERATURE                          Value     Crit    Major    Minor    Minor    Major    Crit
          Sensor                             (deg C)     (Lo)     (Lo)     (Lo)     (Hi)     (Hi)     (Hi)
--------------------------------------------------------------------------------------------------------------
```

```
0/RP0/CPU0
          RP_TEMP_PCB                            29      -10       -5        0       70       75       80
          RP_TEMP_HOT_SPOT                       32      -10       -5        0       70       75       80
          RP_TEMP_LTM4638                        45      -10       -5        0       80       85       90
          RP_TEMP_LTM4644_0                      35      -10       -5        0       80       85       90
          RP_TEMP_LTM4644_1                      38      -10       -5        0       80       85       90
          RP_JMAC_1V0_VCCP_TMON                  30      -10       -5        0       80       85       90
          RP_JMAC_1V0_VNN_TMON                   29      -10       -5        0       80       85       90
          RP_JMAC_1V0_VCC_RAM_TMON               30      -10       -5        0       80       85       90
          RP_JMAC_1V2_DDR_VDDQ_TMON              31      -10       -5        0       80       85       90
0/PM0
          Ambient Temp                           29      -10       -5        0       55       60       65
          Secondary HotSpot Temp                 50      -10       -5        0       85       90       95
          Primary HotSpot Temp                   41      -10       -5        0       65       70       75
0/0/NXR0
          ILAC_LT_P0_eEDFA0                      25       18       19       20       30       31       32
          ILAC_LT_P0_eEDFA1                      25       18       19       20       30       31       32
          ILAC_LT_P0_eEDFA2                      25       18       19       20       30       31       32
          ILAC_LT_P2_eEDFA0                      25       18       19       20       30       31       32
          ILAC_LT_P2_eEDFA1                      25       18       19       20       30       31       32
          ILAC_LT_P2_eEDFA2                      25       18       19       20       30       31       32
          ILAC_CT_1                              29      -10       -7       -5       75       77       80
          ILAC_CT_2                              26      -10       -7       -5       70       73       75
          ILAC_CT_3                              28      -10       -7       -5       70       73       75
          ILAC_CT_4                              28      -10       -7       -5       70       73       75
          ILAC_FT_P0_eEDFA0                      59       55       57       58       62       64       65
          ILAC_FT_P0_eEDFA1                      59       55       57       58       62       64       65
0/Rack
          SA_TEMP_AIR_INLET0                     25      -10       -5        0       45       55       60
          SA_TEMP_AIR_INLET1                     25      -10       -5        0       45       55       60
          SA_TEMP_AIR_EXAUST0                    27      -10       -5        0       75       85       90
          SA_TEMP_AIR_EXAUST1                    26      -10       -5        0       75       85       90
          SA_TEMP_PCB_HOT_SPOT0                  28      -10       -5        0       80       85       90
          SA_TEMP_PCB_HOT_SPOT1                  32      -10       -5        0       80       85       90
          SA_TEMP_PCB_HOT_SPOT2                  28      -10       -5        0       80       85       90
          SA_TEMP_PCB_HOT_SPOT3                  30      -10       -5        0       80       85       90

====================================================================================================
Location  VOLTAGE                             Value      Crit    Minor    Minor     Crit
          Sensor                              (mV)       (Lo)    (Lo)     (Hi)      (Hi)
----------------------------------------------------------------------------------------------------
0/RP0/CPU0
          RP_ADM1266_12V0                     12094    10800    11280    12720    13200
          RP_ADM1266_1V8_CPU                   1801     1670     1750     1850     1930
          RP_ADM1266_1V24_VCCREF               1238     1150     1200     1280     1330
          RP_ADM1266_1V05_CPU                  1054      980     1020     1080     1120
          RP_ADM1266_1V2_DDR_VDDQ              1207     1120     1160     1240     1280
          RP_ADM1266_1V0_VCC_RAM                988      650      700     1250     1300
          RP_ADM1266_1V0_VNN                    858      550      600     1250     1300
          RP_ADM1266_1V0_VCCP                  1008      450      500     1250     1300
          RP_ADM1266_0V6_DDR_VTT                603      560      580      620      640
          RP_ADM1266_3V3_STAND_BY              3310     3070     3200     3400     3530
          RP_ADM1266_5V0                       4996     4650     4850     5150     5350
          RP_ADM1266_3V3                       3328     3070     3200     3400     3530
          RP_ADM1266_2V5_PLL                   2489     2330     2430     2580     2680
          RP_ADM1266_2V5_FPGA                  2500     2330     2430     2580     2680
          RP_ADM1266_1V2_FPGA                  1197     1120     1160     1240     1280
          RP_ADM1266_3V3_CPU                   3332     3070     3200     3400     3530
          RP_ADM1266_2V5_CPU                   2502     2330     2430     2580     2680
0/Rack
          SA_ADM1266_12V_BUS_EITU             12057    10800    11280    12720    13200
          SA_ADM1266_5V0                       5022     4650     4800     5200     5350
          SA_ADM1266_1V8_ZARLINK_DPLL          1806     1670     1730     1870     1930
          SA_ADM1266_1V0_PHY                   1009      930      960     1040     1070
          SA_ADM1266_1V0_ALDRIN_CORE            982      910      930     1070     1090
          SA_ADM1266_1V0_ALDRIN_SERDES         1007      930      960     1040     1070
          SA_ADM1266_1V0_FPGA                  1008      930      960     1040     1070
          SA_ADM1266_1V2_FPGA                  1205     1120     1150     1250     1280
          SA_ADM1266_1V8                       1804     1670     1730     1870     1930
          SA_ADM1266_2V5                       2505     2330     2400     2600     2680
          SA_ADM1266_3V3                       3323     3070     3170     3430     3530
          SA_ADM1275_12V_SA_BP                12058    10800    11280    12720    13200
          SA_ADM1275_12V_CPU_BP               12032    10800    11280    12720    13200
          SA_ADM1275_12V_MOD0_BP              12063    10800    11280    12720    13200
          SA_ADM1275_12V_MOD1_BP              12048    10800    11280    12720    13200
          SA_ADM1275_12V_MOD2_BP              12027    10800    11280    12720    13200
          SA_ADM1275_12V_FAN0_BP              12032    10800    11280    12720    13200
          SA_ADM1275_12V_FAN1_BP              12042    10800    11280    12720    13200
====================================================================================================
Location  CURRENT                              Value
          Sensor                               (mA)
----------------------------------------------------------------------------------------------------
0/RP0/CPU0
          RP_CURRMON_LTM4638                    395
          RP_CURRMON_LTM4644_0                  179
          RP_CURRMON_LTM4644_1                  307
          RP_JMAC_1V0_VCCP_IMON                 125
          RP_JMAC_1V0_VNN_IMON                   62
          RP_JMAC_1V0_VCC_RAM_IMON                0
          RP_JMAC_1V2_DDR_VDDQ_IMON             156
0/Rack
          SA_ADM1275_12V_MOD0_IMON             3412
          SA_ADM1275_12V_MOD1_IMON               30
          SA_ADM1275_12V_MOD2_IMON               43
          SA_ADM1275_12V_FAN0_IMON             1418
          SA_ADM1275_12V_FAN1_IMON             1394
          SA_INA230_5V0_IMON                    129
          SA_INA230_3V3_IMON                   3020
          SA_INA230_1V0_XGE_CORE_IMON          2464
          SA_INA230_1V0_FPGA_CORE_IMON          787
          SA_ADM1275_12V_SA_IMON               1640
          SA_ADM1275_12V_CPU_IMON              1157
===========================================================================
                                              Fan speed (rpm)
Location      FRU Type                        FAN_0    FAN_1   FAN_2
---------------------------------------------------------------------------

0/PM0        NCS1010-AC-PSU                     5424
0/FT0        NCS1010-FAN                        9960    9960    9960
0/FT1        NCS1010-FAN                       10020   10020   10020
================================================================================
Location     Altitude Value (Meters)    Source
--------------------------------------------------------------------------------
0            760                    sensor

================================================================================
CHASSIS LEVEL POWER INFO: 0
================================================================================
   Total output power capacity (Group 0 + Group 1) :    1050W +        0W
   Total output power required                     :     700W
   Total power input                               :     159W
   Total power output                              :     129W

Power Group 0:
================================================================================
   Power       Supply         ------Input----   ------Output---     Status
   Module      Type            Volts     Amps    Volts     Amps
================================================================================
   0/PM1       NCS1010-AC-PSU  0.0       0.0     0.0       0.0      OFFLINE

Total of Group 0:                0W/0.0A         0W/0.0A
```

```
Power Group 1:
================================================================================
   Power       Supply         ------Input----   ------Output---     Status
   Module      Type            Volts     Amps    Volts     Amps
================================================================================
   0/PM0       NCS1010-AC-PSU  228.5     0.7     12.1      10.7     OK

Total of Group 1:              159W/0.7A         129W/10.7A

================================================================================
   Location     Card Type               Power       Power        Status
                                        Allocated   Used
                                        Watts       Watts
================================================================================
   0/RP0/CPU0   NCS1010-CNTLR-K9        90          14           ON
   0/FT0        NCS1010-FAN             110         17           ON
   0/FT1        NCS1010-FAN             110         16           ON
   0/0/NXR0     NCS1K-ILA-C             350         54           ON
   0/Rack       NCS1010-SA              40          19           ON
```

Environment parameter anomalies are logged in the syslog. As a result, if an environment parameter that is displayed in the **show environment** command output is not as expected, check the syslog using the **show logging** and **show alarms brief system active** command. The syslog provides details on any logged problems.

Verify Context

The **show context** command displays core dump context information of NCS 1010. Core dump is a result of abnormal exit of any process running in the system.

Procedure

show context

Displays the core dump context information of NCS 1010.

Example:

```
RP/0/RP0/CPU0:ios# show context
Mon Sep 27 17:21:59.219 UTC

node: node0_RP0_CPU0
----------------------------------------------------------------
No context
```

The command output is empty during system upgrade.

Verify Core Files

Use the **run** command to go to the hard disk location and check for the core dumps of NCS 1010.

Procedure

run

Example:

```
RP/0/RP0/CPU0:ios# run
Mon Sep 27 17:29:11.163 UTC
[xr-vm_node0_RP0_CPU0:~]$cd /misc/disk1/
[xr-vm_node0_RP0_CPU0:/misc/disk1]$ls -lrt *.tgz
```

Verify Memory Information

You can view the memory information using the show watchdog memory-state command.

Procedure

show watchdog memory-state location all

Displays memory snapshot in brief.

Example:

```
RP/0/RP0/CPU0:ios#show watchdog memory-state location all
Thu Jun 16 08:36:44.436 UTC
---- node0_RP0_CPU0 ----
Memory information:
    Physical Memory     : 31935.167 MB
    Free Memory         : 29236.0   MB
    Memory State        :   Normal
```

Complete Post-setup Tasks

You must create user profiles and user groups to manage your system, install software packages, and configure your network.

Every user is authenticated using a username and a password. The authentication, authorization, and accounting (AAA) commands help with these services:

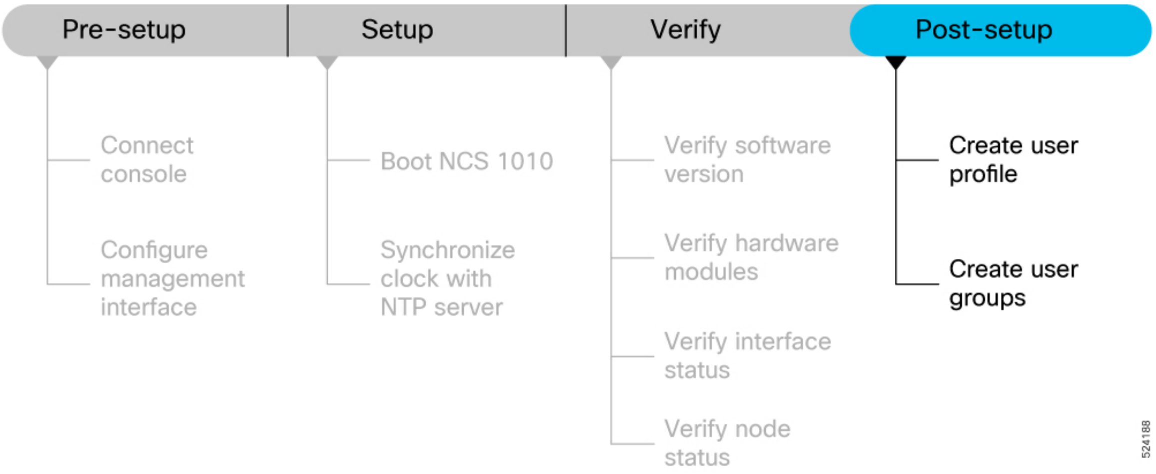
- Create users, groups, command rules, or data rules
- Change the disaster-recovery password

IOS-XR and Linux have separate AAA services and IOS XR AAA is the primary AAA system. A user who is created through IOS-XR can log in directly to the EXEC prompt when connected to the NCS 1010, while a user created through Linux can connect to the NCS 1010, but can log in to the bash prompt. The user must log in to IOS XR explicitly, to access the IOS-XR EXEC prompt.

You must configure the IOS-XR AAA authorization to restrict users from uncontrolled access. If AAA is not configured, the command and data rules associated to the groups that are assigned to the user are ignored. A user can have full read/write access to IOS XR configuration through Network Configuration Protocol (NETCONF), google-defined Remote Procedure Calls (gRPC), or any YANG-based agents. To avoid granting uncontrolled access, enable AAA before setting up any configuration. To gain an understanding about AAA, and to explore the AAA services, see Configure AAA, on page 47.

The following image provides you an overview of the various tasks that are involved in the Cisco NCS 1010 Series NCS 1010 post-setup procedure.

Figure 8: Post-setup Workflow for the Cisco NCS 1010



Ensure that you have completed the Setup NCS 1010, on page 21 and Verify the Software and Hardware Status, on page 27 tasks before you perform the following tasks:

Create User Profile

You can create new users and include the user in a user group with certain privileges. The NCS 1010 supports a maximum of 1024 user profiles.

Perform the following steps to create a user profile:

Procedure

Step 1 Create a user, provide a password and assign the user to a group. For example, **user1** is the user, password is **pw123**, and the group is **root-lr**.

Example:

```
RP/0/RP0/CPU0:ios#config

/* Create a new user */
ios(config)#username user1

/* Set a password for the new user */
ios(config-un)#password pw123
```

```
/* Assign the user to group root-lr */
RP/0/RP0/CPU0:ios(config-un)#group root-lr
```

All users have read privileges. The **root-lr** users inherit write privileges where users can create configurations, create new users, and so on.

Enable display of login banner: The US Department of Defense (DOD)-approved login banner provides information such as number of successful and unsuccessful login attempts, time stamp, login method, and so on. The banner is displayed before granting access to devices. The banner also ensures privacy and security that is consistent with applicable federal laws. In addition, the system keeps track of logins, right from the system boot, or as soon as the user profile is created.

You can enable or diable the login login banner by using the **login-history enable** and **login-history disable** commands.

Note
Login notifications get reset during a NCS 1010 reload.

Step 2 Run the **show running-config username user1** command to verify the state of login banner.

Example:

```
RP/0/RP0/CPU0:ios(config-un)#show running-config username NAME1
 Fri Jan 29 13:55:28.261 UTC
 username NAME1
 group UG1
 secret * **********
 password * ******
 login-history enable
```

Step 3 Commit the configuration.

Example:

```
RP/0/RP0/CPU0:ios(config-un)#commit
```

The user profile is created and allowed access to the NCS 1010 based on the configured privileges.

Create User Groups

You can create a new user group to associate command rules and data rules with it. The command rules and data rules are enforced on all users that are part of the user group. The NCS 1010 supports a maximum of 32 user groups.

Before you begin

Ensure that you have created a user profile. See Create User Profile, on page 44.

Procedure

Step 1 Create a new user group.

Example:

```
RP/0/RP0/CPU0:ios#config

/* Create a new user group, group1 */
```

```
ios#(config)#group group1

/* Specify the name of the user, user1 to assign to this user group */
ios#(config-GRP)#username user1
```

Step 2 Commit the configuration.

Example:

```
RP/0/RP0/CPU0:ios(config-GRP)#commit
```

What to do next

This completes the NCS 1010 setup and verification process. You can now proceed with upgrading the software, installing RPMs, SMUs and bug fixes based on your requirement.

CHAPTER 4

Configure AAA

This chapter describes the implementation of the administrative model of *task-based authorization* used to control user access in the software system. The major tasks required to implement task-based authorization involve configuring TACACS+ and RADIUS servers and groups.

> **Note** From Release 24.4.1, the AAA local database supports configuring up to 3000 usernames. Although you can configure more than 3000 users, it may impact the system's scale and performance, which are not assured beyond this limit.

Deprecation of Type 7 password and Type 5 secret

Password configuration options before Release 24.4.1

Until Release 24.4.1, there were two options for configuring a password:

- Password: Uses Type 7 encryption to store the password.
- Secret: Supports Type 5, 8, 9, or 10 hashing algorithms to store the password securely.

Deprecation notice

Starting from the Release 24.4.1, the use of Type 7 password and Type 5 secret are deprecated due to security concerns. The deprecation process commences from the Release 24.4.1. We expect the full deprecation in a future release. We recommend using the default option, which is Type 10 secret.

- aaa password-policy, on page 50
- secret, on page 50
- masked-secret, on page 51

password

The **password** options available in CLI from the Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config-un)#password ?
LINE The type 7 password followed by '7 ' OR SHA512-based password (deprecated, use 'secret')
```

Changes:

- All the options that were present until the Release 24.4.1are removed except LINE (to accept cleartext).
- **During upgrade**: Any configuration using the Type 7 password configuration is automatically converted to Type 10 secret.

  Post-upgrade: You can still use the Type 7 password configurations option after new commits, but the password will be stored as Type 10 secret.
- New **syslog** has been added to indicate the deprecation process:

  ```
  %SECURITY-PSLIB-4-DEPRECATED_PASSWORD_TYPE : The password configuration is deprecated.

   Converting it to a Type 10 secret for user <user name>.
  ```
- **show running configuration** command output before upgrade:

  ```
  username example
  password 7 106D000A0618
  !
  ```

  show running configuration command output post-upgrade:

  ```
  username example
  Cisco Confidential
  secret 10
  $6$P53pb/FFxNIT4b/.$yVakako4fp9PZiIYYh1xS0.W6b/yPrSyC8j4gLs6xli57iClOryPXyN9y8yojRD2nhAWb9pjr/WAIhbXqq8st.
  !
  ```

masked-password

The **masked-password** options available in CLI from the Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config-un)#masked-password ?
0 Specifies a cleartext password will follow
clear Config deprecated. Will be removed in 7.7.1. Specify '0' instead.
<cr> The cleartext user password
```

Changes:

- The options 7 and encrypted that were present until the Release 24.4.1 are removed.
- **During upgrade**: Any configuration using the Type 7 password configuration is automatically converted to Type 10 secret.
- **Post-upgrade**: Masked-password is an alternate method of configuring the password. You can still use the masked-password keyword with a clear string after new commits, but the password will be stored as Type 10 secret.

- New **syslog** has been added to indicate the deprecation process:

  ```
  %SECURITY-PSLIB-4-DEPRECATED_PASSWORD_TYPE : The password configuration is deprecated.

   Converting it to a Type 10 secret for user <user name>.
  ```

- **show running configuration** command output before upgrade:

  ```
  username example
  password 7 106D000A0618
  !
  ```

  show running configuration command output post-upgrade:

  ```
  username example
  Cisco Confidential
  secret 10
  $6$P53pb/FFxNIT4b/.$yVakako4fp9PZiIYYh1xS0.W6b/yPrSyC8j4gLs6xli57iClOryPXyN9y8yojRD2nhAWb9pjr/WAIhbXqq8st.
  !
  ```

password-policy

The **password-policy** options available in CLI from the Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config-un)#password-policy ?
WORD Specify the password policy name

RP/0/RP0/CPU0:ios(config-un)#password-policy abcd password ?
0 Specifies an UNENCRYPTED password will follow
7 Specifies that an encrypted password will follow
LINE The UNENCRYPTED (cleartext) user password
clear Config deprecated. Will be removed in 7.7.1. Specify '0' instead.
encrypted Config deprecated. Will be removed in 7.7.1. Specify '7' instead.
```

Changes:

- All the options that were present until 24.4.1are removed except LINE (to accept cleartext).
- **During upgrade**: Any configuration using the Type 7 password configuration is automatically converted to Type 10 secret.

  Post-upgrade: You can still use the password-policy configurations option after new commits, but the it will be stored as Type 10 secret.

- New **syslog** has been added to indicate the deprecation process:

  ```
  %SECURITY-PSLIB-4-DEPRECATED_PASSWORD_TYPE : The password configuration is deprecated.

   Converting it to a Type 10 secret for user <username>.
  ```

- **show running configuration** command output before upgrade:

  ```
  username example
  password-policy abcd password 7 106D000A0618
  !
  ```

  show running configuration command output post-upgrade:

  ```
  username example
  secret 10
  $6$P53pb/FFxNIT4b/.$yVakako4fp9PZiIYYh1xS0.W6b/yPrSyC8j4gLs6xli57iClOryPXyN9y8yojRD2nhAWb9pjr/WAIhbXqq8st.
  !
  !
  ```

aaa password-policy

The **aaa password-policy** options available in CLI from the Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config)#aaa password-policy abcd
RP/0/RP0/CPU0:ios(config-pp)#?
min-char-change Number of characters change required between old and new passwords
(deprecated, will be removed in 25.3.1)
restrict-password-advanced Advanced restrictions on new password (deprecated, will be removed
 in 25.3.1)
restrict-password-reverse Restricts the password to be same as reversed old password
(deprecated, will be removed in 25.3.1)
```

Changes:

- The options min-char-change, restrict-password-advanced, and restrict-password-reverse that were present until the Release 24.4.1 are deprecated.
- **During upgrade**: These deprecated configurations do not go through any change during upgrade.

  Post-upgrade: These deprecated keywords do not take effect when configured post-upgrade.
- New **syslog** have been added to indicate the deprecation process:

  -
    ```
    %SECURITY-LOCALD-4-DEPRECATED_PASSWORD_POLICY_OPTION : The password policy option
    'min-char-change' is deprecated.
    Password/Secret will not be checked against this option now.
    ```

  -
    ```
    %SECURITY-LOCALD-4-DEPRECATED_PASSWORD_POLICY_OPTION : The password policy option
    'restrict-password-reverse' is deprecated.
     Password/Secret will not be checked against this option now.
    ```

  -
    ```
    %SECURITY-LOCALD-4-DEPRECATED_PASSWORD_POLICY_OPTION : The password policy option
    'restrict-password-advanced' is deprecated.
    Password/Secret will not be checked against this option now.
    ```
- **show running configuration** command output before upgrade:

  ```
  aaa password-policy abcd
  lower-case 3
  min-char-change 1
  restrict-password-reverse
  restrict-password-advanced
  !
  ```

  show running configuration command output post-upgrade:

  ```
  aaa password-policy abcd
  lower-case 3
  min-char-change 1
  restrict-password-reverse
  restrict-password-advanced
  !
  ```

secret

The **secret** options available in CLI from the Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config-un)#secret ?
0 Specifies a cleartext password will follow
10 Specifies that SHA512-based password will follow
8 Specifies that SHA256-based password will follow
9 Specifies that Scrypt-based password will follow
LINE The cleartext user password
```

```
RP/0/RP0/CPU0:ios(config-un)#secret 0 enc-type ?
<8-10> Specifies which algorithm to use. Only 8,9,10 supported [Note: Option ‘5’ is not
available to use from 24.4]
```

Changes:

- The options 5 and encrypted are removed.
- **During upgrade**: Configurations using Type 5 secret will remain unchanged.

  Post-upgrade: Though the keyword 5 has been deprecated, you can still apply the existing configurations using Type 5 secret.
- New **syslog** has been added to indicate the deprecation process:

  ```
  %SECURITY-LOCALD-2-DEPRECATED_SECRET_TYPE : Type 5 secret is deprecated.
  Please use the 'secret' keyword with option type 10 for user.
  ```
- **show running configuration** command output before upgrade:

  ```
  username example
  secret 5 $1$kACo$2RtpcwyiRuRB/DhWzabfU1
  !
  !
  ```

  show running configuration command output post-upgrade:

  ```
  username example
  secret 5 $1$kACo$2RtpcwyiRuRB/DhWzabfU1
  !
  !
  ```

masked-secret

The **masked-secret** options available in CLI from the Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config-un)#masked-secret ?
0 Specifies a cleartext password will follow
Cisco Confidential
10 Specifies that SHA512-based password will follow
8 Specifies that SHA256-based password will follow
9 Specifies that Scrypt-based password will follow
clear Config deprecated. Will be removed in 7.7.1. Specify '0' instead.
<cr> The cleartext user password
```

Changes:

- The options 5 and encrypted are removed.
- **During upgrade**: Configurations using masked-secret with Type 5 will remain unchanged.

  Post-upgrade: Though the keyword 5 has been deprecated, you can still apply the existing configurations using Type 5 masked secret.
- New **syslog** has been added to indicate the deprecation process:

  ```
  %SECURITY-LOCALD-2-DEPRECATED_SECRET_TYPE : Type 5 secret is deprecated.
  Please use the 'secret' keyword with option type 10 for user.
  ```
- **show running configuration** command output before upgrade:

  ```
  username example
  secret 5 $1$kACo$2RtpcwyiRuRB/DhWzabfU1
  !
  !
  ```

show running configuration command output post-upgrade:

```
username example
secret 5 $1$kACo$2RtpcwyiRuRB/DhWzabfU1
!
!
```

Special use cases

Use case 1: Configurations using both Type 7 password and secret with 8, 9, or 10 hashing, for the same user

- **During upgrade**:
  - For the first 3000 username configurations, the password configuration will be rejected, and the secret configuration will remain unchanged.
  - For the rest of the username configurations, the original secret configuration will be rejected, and the password will be converted to Type 10 secret.
- **Post-upgrade**:
  - For a new username configured, or the username that is already present before the upgrade, the password configuration will be rejected.
  - New **syslog** has been added to indicate the deprecation process:

    ```
    %SECURITY-PSLIB-4-SECRET_CONFIG_PRESENT : The password configuration is deprecated.

    Once secret is configured, cannot use password config for user <user name> at index
     <x> now.
    ```

    where 'x' is a number representing the index.

Use case 2: Configurations using both Type 7 password and Type 5 secret, for the same user

- **During upgrade**:
  - For any username configuration, the original Type 5 secret configuration will be rejected, and the password will be converted to Type 10 secret.
- **Post-upgrade**:
  - For a new username configured, or the username that is already present before the upgrade, the password configuration will be converted to Type 10 secret.
  - New **syslog** has been added to indicate the deprecation process:

    ```
    %SECURITY-PSLIB-4-DEPRECATED_PASSWORD_TYPE : The password configuration is
    deprecated.
     Converting it to a Type 10 secret for user <username>.
    ```

About TACACS+

The Terminal Access Controller Access Control System Plus (TACACS+) application is designed to enhance the security of the NCS 1010 device by centralizing user validation. It uses AAA commands and can be enabled and configured on NCS 1010 for improved security. TACACS+ provides detailed accounting information and flexible administrative control over user access.

When TACACS+ server is configured and protocol is enabled on the node, the user credentials are authenticated through TACACS+ server. When the user attempts to log into the node, the username and password is forwarded to the configured TACACS+ servers and get authentication status. If the authentication fails through TACACS+ server, the credentials are sent to the node and are authenticated against the node. If the authentication fails against the node, the user is not allowed to log into the node.

Configure TACACS+ Server

Enabling the AAA accounting feature on a switch allows it to track the network services that users are accessing and the amount of network resources they are using. The switch then sends this user activity data to the TACACS+ security server in the form of accounting records. Each record contains attribute-value pairs and is saved on the security server for analysis. This data can be used for network management, client billing, or auditing purposes.

To configure TACACS+ server, perform these steps:

Procedure

Step 1 Enter into the IOS XR configuration mode.

Example:

```
RP/0/RP0/CPU0:ios#configure
```

Step 2 Enable the TACACS+ accounting to send a start-record accounting notice at the beginning of a privileged EXEC process and a stop-record at the end.

Example:

```
RP/0/RP0/CPU0:ios(config)#aaa accounting exec default start-stop group TACACS_ALL
```

Step 3 Create a default command accounting method list for accounting services provided by a TACACS+ security server. This list is configured for privilege level commands and set with a stop-only restriction.

Example:

```
RP/0/RP0/CPU0:ios(config)#aaa accounting exec default start-stop group TACACS_ALL
```

Configure TACACS+ Server Groups

Configuring NCS 1010 to use AAA server groups provides a way to group existing server hosts. This allows you to select a subset of the configured server hosts and use them for a particular service. A server group is used in conjunction with a global server-host list. The server group lists the IP addresses of the selected server hosts.

You can enter one or more **server** commands. The **server** command specifies the hostname or IP address of an external TACACS+ server. Once configured, this server group can be referenced from the AAA method lists (used while configuring authentication, authorization, or accounting).

To configure TACACS+ server groups, perform these steps:

Before you begin

For successful configuration, the external server should be accessible at the time of configuration. When configuring the same IP address for global configuration, server-private parameters are required.

Procedure

Step 1 Enter into the IOS XR configuration mode.

Example:

```
RP/0/RP0/CPU0:ios# configure
```

Step 2 Create an AAA server-group and enter into the server group sub-configuration mode.

Example:

```
RP/0/RP0/CPU0:ios(config)# aaa group server tacacs+ tacgroup1
```

Step 3 Configure the IP address of the private TACACS+ server for the group server.

Example:

```
RP/0/RP0/CPU0:ios(config-sg-tacacs+)# server-private 10.1.1.1 port 49 key a_secret
```

Note

- You can configure a maximum of 10 TACACS+ private servers in a server group.
- If private server parameters are not specified, global configurations are used. If global configurations are not specified, default values are used.

Step 4 Configure the authentication and encryption key used between NCS 1010 and the TACACS+ daemon running on the TACACS+ server. If no key string is specified, the global value is used.

Example:

```
 RP/0/RP0/CPU0:ios(config-sg-tacacs+)# key 7 08984B1A4D0C19157A5F57
```

Step 5 Configure the timeout value that sets the length of time the authentication, authorization, and accounting (AAA) server waits to receive a response from the TACACS+ server.

Example:

```
RP/0/RP0/CPU0:ios(config-sg-tacacs-private)# timeout 4
```

Step 6 Repeat steps 3 to 5 for every private server to be added to the server group.

Step 7 Configure certificate-based authentication for users configured in the TACACS+ server or server groups.

Example:

```
RP/0/RP0/CPU0:ios(config-sg-tacacs-private)#aaa authorization exec default group TACACS_ALL local
```

Step 8 Set the default method list for authentication, and also enables authentication for console in global configuration mode.

Example:

```
RP/0/RP0/CPU0:ios(config-sg-tacacs-private)#aaa authentication login default group TACACS_ALL local
```

Step 9 Commit the changes and exit all the configuration modes.

commit

end

Step 10 Verify the TACACS+ server group configuration details.

Example:

```
RP/0/RP0/CPU0:ios# show tacacs server-groups
```

About RADIUS

Remote Authentication Dial-In User Service (RADIUS) is a distributed client/server system that provides security against unauthorized access in distributed client/server networks. In Cisco's implementation, RADIUS clients operate on Cisco NCS 1010 and send requests for authentication and accounting to a central RADIUS server that contains all user authentication and network service access information.

Cisco's AAA security paradigm supports RADIUS, which can be used alongside other security protocols like TACACS+, Kerberos, and local username lookup.

Configure RADIUS Server Groups

You can enter one or more **server** commands. The **server** command specifies the hostname or IP address of an external RADIUS server along with port numbers. When configured, this server group can be referenced from the AAA method lists (used while configuring authentication, authorization, or accounting).

You can configure a maximum of 30 servers and private servers each per RADIUS server group. To configure RADIUS server groups, perform these tasks:

Before you begin

Ensure that the external server is accessible at the time of configuration.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:ios# configure
```

Enters mode.

Step 2 **aaa group server radius** *group-name*

Example:

```
RP/0/RP0/CPU0:ios(config)# aaa group server radius radgroup1
```

Groups different server hosts into distinct lists and enters the server group configuration mode.

Step 3 **radius-server** {*ip-address*}

Example:

```
RP/0/RP0/CPU0:ios(config)# radius-server host 192.168.20.0
```

Specifies the hostname or IP address of the RADIUS server host.

Step 4 **auth-port** *port-number*

Example:

```
RP/0/RP0/CPU0:ios(config)#auth-port 1812
```

Specifies the User Datagram Protocol (UDP) destination port for authentication requests; the host is not used for authentication if set to 0. If unspecified, the port number defaults to 1645.

Step 5 **acct-port** *port-number*

Example:

```
RP/0/RP0/CPU0:ios(config)# acct-port 1813
```

Specifies the UDP destination port for accounting requests; the host is not used for accounting if set to 0. If unspecified, the port number defaults to 1646.

Step 6 **key** *string*

Example:

```
RP/0/RP0/CPU0:ios(config-radius-host)#key 7 08984B1A4D0C19157A5F57
```

Specifies the authentication and encryption key used between NCS 1010 and the RADIUS server. This key overrides the global setting of the **radius-server key** command. If no key string is specified, the global value is used.

The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the **radius-server host** command syntax. This is because the leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key.

Step 7 Repeat steps 4 to 6 for every external radius server to be added to the server group.

—

Step 8 **aaa authentication**{**login**}{**default**} **group** *group-name* **local**

Example:

```
RP/0/RP0/CPU0:ios(config-radius-host)#aaa authentication login default group radius local
```

Specifies the default method list for authentication, and also enables authentication for console in global configuration mode.

Step 9 Use the **commit** or **end** command.

Step 10 **show radius server-groups** [*group-name* [**detail**]]

Example:

```
RP/0/RP0/CPU0:ios# show radius server-groups
```

(Optional) Displays information about each RADIUS server group that is configured in the system.

CHAPTER 5

Workflow to Setup and Upgrade NCS 1010

NCS 1010 comes preinstalled with IOS XR software. You can upgrade NCS 1010 by installing a new version of the software. We recommend that you keep the software up-to-date to ensure that NCS 1010 works with the latest features and bug fixes.

During an upgrade:

- The newer software replaces the currently active software on NCS 1010.
- Packages (RPMs) that have the same name and version in the current and target release versions are not removed or reinstalled.

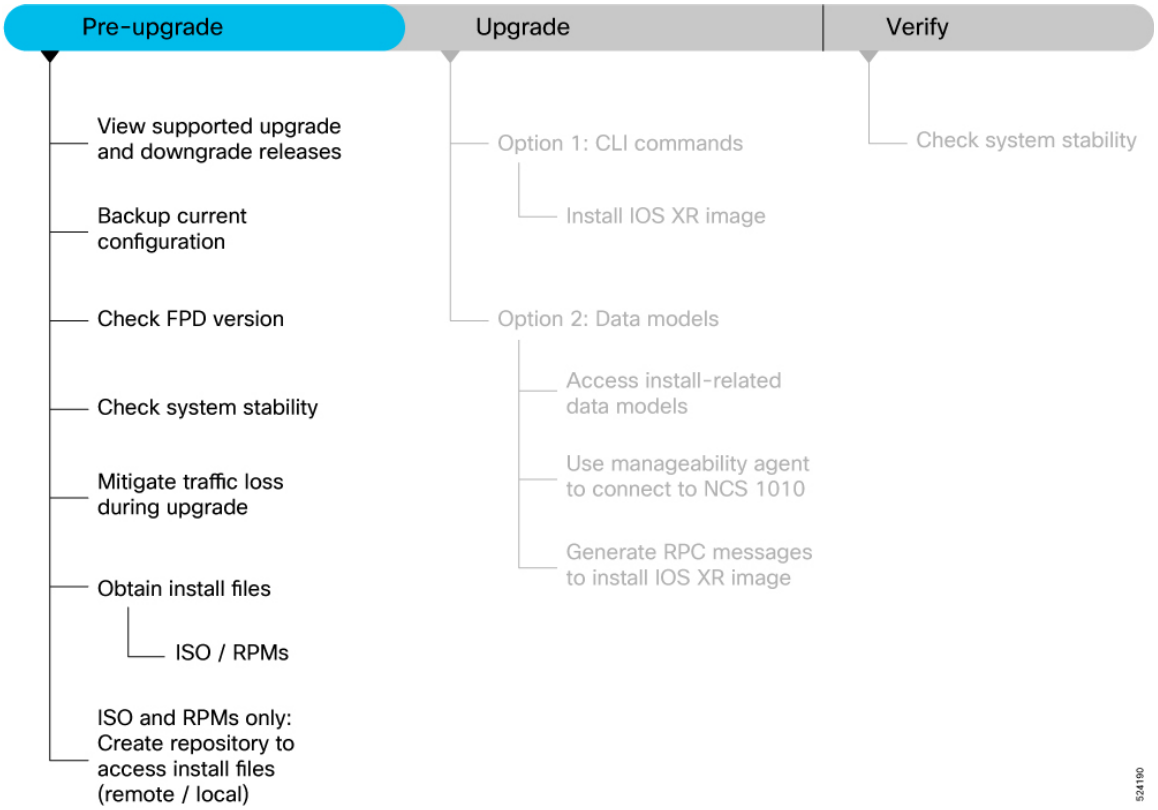
This section contains the following topics:

Plan the Software Upgrade

Before you upgrade the software version, prepare NCS 1010 to ensure that the upgrade process is seamless.

Pre-upgrade workflow image for NCS 1010

Figure 9: NCS 1010 Pre-upgrade workflow


Pre-upgrade
Upgrade
Verify
View supported upgrade and downgrade releases
Backup current configuration
Check FPD version
Check system stability
Mitigate traffic loss during upgrade
Obtain install files
ISO / RPMs
ISO and RPMs only: Create repository to access install files (remote / local)
Option 1: CLI commands
Install IOS XR image
Option 2: Data models
Access install-related data models
Use manageability agent to connect to NCS 1010
Generate RPC messages to install IOS XR image
Check system stability
524190

View Supported Upgrade and Downgrade Releases

This section provides the supported upgrade and downgrade paths for NCS 1010. Consider the following guidelines when performing an upgrade or downgrade:

- When downgrading the software image from release 24.4.x to an earlier version, we recommend to manually downgrade the line card firmware as well to prevent any impact on various functionalities.
- It is recommended to perform the upgrade with FPD auto-upgrade enabled to ensure the FPD versions are up to date and to prevent potential upgrade-related issues.

The following table lists the upgrade and downgrade paths supported for Cisco NCS 1010.

Upgrade Path			Downgrade Path		
Source Release	Destination Release	Bridge SMU	Source Release	Destination Release	Bridge SMU
• R7.10.1 • R7.11.2 • R24.2.11 • R24.3.1 • R24.4.15	R25.1.1	No	R25.1.1	• R24.4.15 • R24.3.1 • R24.2.11 • R7.11.2 • R7.10.1	No

Backup Current Configuration

The ability to recover from a disaster is an essential part of any system maintenance plan. We recommend you backup the configurations in a secure remote location and verify that the transfer is a success, both before and after upgrade.

Procedure

Step 1 Create a backup of the running configuration to one of the following locations based on your requirement:

- Copy the configuration to the `harddisk:` location on NCS 1010.

```
RP/0/RP0/CPU0:ios#copy running-config harddisk:/running_config-<mmddyyyy>
Destination filename [running_config-<mmddyyyy>]?
Building configuration...
[OK]
Verifying checksum...  OK (0xDCF1)
```

- Copy the configuration to a remote server. Ensure NCS 1010 has root access to the server.

```
RP/0/RP0/CPU0:ios#scp harddisk:/ running_config-<mmddyyyy>
user:password@<ip-address>:<location>
```

Step 2 Verify that the configuration is backed up.

FPD

Table 5: Feature History

Feature Name	Release Information	Feature Description
FPD Upgrade for Passive Modules	Cisco IOS XR Release 7.10.1	You can now perform FPD upgrade of the breakout modules and multiplexer/demultiplexer modules. It is essential to upgrade the passive modules to ensure the proper functioning of the modules. You can upgrade the FPD on all passive modules simultaneously or selectively upgrade the required modules.

A Field Programmable Device (FPD) refers to any programmable hardware device on a chassis, which includes a Field Programmable Gate Array (FPGA). NCS 1010 uses several FPDs that are necessary for chassis, route processor, line cards, and power modules to function properly. Before upgrading the software, check whether the latest FPDs are available on NCS 1010.

Note FPD auto-upgrade is enabled by default on NCS 1010.

From Release 7.10.1, you can perform FPD upgrade for the breakout and multiplexer/demultiplexer modules. For the breakout modules, you can perform the FPD upgrade in both direct and indirect connections. You can upgrade all the passive modules at once or selectively upgrade the necessary modules as needed.

Note If the FPD in a given SSD is not supported by the current IOS XR software release, the status is displayed as *NOT READY*. The status will change once FPD support for these SSDs is enabled in future releases.

Table 6: NCS 1010 FPDs

Location	FPDs
RP	• ADMConfig • CpuFpga • CpuFpgaGolden • BIOS • BIOS-Golden • SsdIntelS4510 • SsdMicron5300 • SsdSmartModular • TamFw • TamFwGolden
PM0 and PM1	• AP-PrimMCU • AP-SecMCU
LC	• ILA • OLT • Raman-1 • Raman-2
Rack	• IoFpga • IoFpgaGolden • EITU-ADMConfig • SsdIntelS4510 • SsdMicron5300 • SsdSmartModular
Breakout module	• BRK-8 • BRK-24
Multiplexer and demultiplexer modules	• MD-32-ACC • MD-32-NEO

Golden FPDs serve as backup FPDs for the primary FPDs. For example, **BIOS-Golden** is the backup Golden FPD for the **BIOS** primary FPD. If a primary FPD is corrupted, NCS 1010 boots with the corresponding Golden FPD. The Golden FPDs cannot be upgraded.

Check FPD Version

There are multiple types of FPDs for each type of module. The **show hw-module fpd** command provides information about each FPD.

```
RP/0/RP0/CPU0:ios#show hw-module fpd
Fri Feb 17 11:43:28.878 UTC

Auto-upgrade:Enabled
Attribute codes: B golden, P protect, S secure, A Anti Theft aware
                                                                         FPD Versions
                                                                        ==============
Location   Card type              HWver FPD device       ATR Status   Running Programd
Reload Loc
-------------------------------------------------------------------------------------------
0/RP0/CPU0 NCS1010-CTLR-B-K9     1.0   ADMConfig            CURRENT    2.30    2.30
NOT REQ
0/RP0/CPU0 NCS1010-CTLR-B-K9     1.0   BIOS             S   CURRENT    4.40    4.40
  0/RP0
0/RP0/CPU0 NCS1010-CTLR-B-K9     1.0   BIOS-Golden      BS  CURRENT            4.40
  0/RP0
0/RP0/CPU0 NCS1010-CTLR-B-K9     1.0   CpuFpga          S   CURRENT    1.11    1.11
  0/RP0
0/RP0/CPU0 NCS1010-CTLR-B-K9     1.0   CpuFpgaGolden    BS  CURRENT            1.01
  0/RP0
0/RP0/CPU0 NCS1010-CTLR-B-K9     1.0   SsdIntelS4510    S   CURRENT   11.32   11.32
  0/RP0
0/RP0/CPU0 NCS1010-CTLR-B-K9     1.0   TamFw            S   CURRENT    6.13    6.13
  0/RP0
0/RP0/CPU0 NCS1010-CTLR-B-K9     1.0   TamFwGolden      BS  CURRENT            6.11
  0/RP0
0/PM0      NCS1010-AC-PSU        0.0   AP-PriMCU            CURRENT    1.03    1.03
NOT REQ
0/PM0      NCS1010-AC-PSU        0.0   AP-SecMCU            CURRENT    2.01    2.01
NOT REQ
0/PM1      NCS1010-AC-PSU        0.0   AP-PriMCU            CURRENT    1.03    1.03
NOT REQ
0/PM1      NCS1010-AC-PSU        0.0   AP-SecMCU            NEED UPGD  1.06    1.06
NOT REQ
0/0/NXR0   NCS1K-E-OLT-R-C       1.0   OLT              S   CURRENT    1.16    1.16
NOT REQ
0/0/NXR0   NCS1K-E-OLT-R-C       1.0   Raman-1          S   CURRENT    1.04    1.04
NOT REQ
0/Rack     NCS1010-SA            0.1   EITU-ADMConfig       CURRENT    1.04    1.04
NOT REQ
0/Rack     NCS1010-SA            0.1   IoFpga           S   CURRENT            1.12
NOT REQ
0/Rack     NCS1010-SA            0.1   IoFpgaGolden     BS  NEED UPGD  1.12    0.08
NOT REQ
0/Rack     NCS1010-SA            0.1   SsdIntelS4510    S   CURRENT   11.32   11.32
 0/Rack
0/1        NCS1K-MD-32E-C        0.1   MD-32-NEO        S   CURRENT    2.02    2.02
NOT REQ
0/2        NCS1K-MD-32O-C        10.2  MD-32-ACC        S   CURRENT    2.18    2.18
NOT REQ
0/3/0      NCS1K-BRK-8           1.0   BRK-8            S   CURRENT    2.08    2.08
NOT REQ
0/3/3      NCS1K-BRK-24          1.0   BRK-24           S   CURRENT    2.08    2.08
NOT REQ
```

If the status of any FPD is **NEED UPGD**, then the upgrade is required for that FPD. In this example, `IoFpgaGolden` FPD devices need an upgrade. You must ensure that FPDs are upgraded *before* upgrading NCS 1010.

The following table lists the NCS 1010 FPDs that are distributed across route processor (RP), power modules (PM), line cards (LC), and Rack.

The following table describes the significant fields in the output of the **show hw-module fpd** command.

Table 7: Description of Fields in show hw-module fpd Command

Field	Description
Location	Location of the FPD.
Card type	PID of the modules such as chassis, card, CPU, and PSU.
HWver	Hardware version where the FPD resides.
FPD device	Name of the FPD.
ATR	Attribute codes. The possible values are: • B - Golden Image • S - Secure Image • P - Protect Image The attribute code of the primary FPDs is S and the Golden FPDs is BS.
Status	Status of the FPD. See Table 8: Description of FPD Status Values in show hw-module fpd Command, on page 65.
Running	FPD image version that has been activated and currently running in the FPD device.
Programd	FPD image version that has been programmed into the FPD device, but might not be activated.
Reload Loc	Indicates whether reload of the location is required or not.

The following table describes the possible values of the Status field in the output of the **show hw-module fpd** command.

Table 8: Description of FPD Status Values in show hw-module fpd Command

FPD Status	Description
NOT READY	The driver that owns the FPD device has not initialized the FPD client to handle this device.
CURRENT	FPD version is up to date and upgrade is not required.

FPD Status	Description
NEED UPGD	Upgrade is required for this FPD. Check the output of the **show fpd package** command to determine the recommended FPD version.
UPGD PREP	FPD is preparing for upgrade.
IN QUEUE	Upgrade of this FPD is in queue.
UPGD SKIP	FPD upgrade is not required. For example, • FPD version is up to date and compatible. • FPD image is protected.
UPGRADING	FPD upgrade started and the driver did not report the upgrade progress information yet.
%UPGD	Percentage of FPD upgrade completion.
RLOAD REQ	FPD upgrade is successfully completed and the FPD must be reloaded for the new version to take effect.
UPGD FAIL	FPD upgrade has failed. Check the syslog for failure reason. It could be a timeout or a failure that is reported by the driver.
UPGD DONE	FPD upgrade is successfully completed.

The **show fpd package** command is used to determine the FPDs that are supported with the current software release and the minimum hardware requirements for each FPD.

The following table describes the fields in the output of the **show fpd package** command.

Table 9: Description of Fields in show fpd package Command

Field	Description
Card Type	PID of the modules such as chassis, card, CPU, and PSU.
FPD Description	Description of the FPD.
Req Reload	Determines whether reload is required to activate the FPD image.
SW Ver	Recommended FPD software version for the associated module running the current Cisco IOS XR Software.
Min Req SW Ver	Minimum required FPD software version to operate the module.

Field	Description
Min Req Board Ver	Minimum required hardware version for the associated FPD. A minimum hardware requirement of version 0.0 indicates that all the hardware can support this FPD version.

Upgrade FPDs Automatically

The automatic FPD upgrade upgrades the FPD version of all the modules to the latest version. When automatic FPD upgrade is enabled, all the FPDs (except the Golden FPDs) that are in NEED UPGD status are upgraded to CURRENT status during the software upgrade.

In NCS 1010, automatic FPD upgrade is enabled by default.

Procedure

Use the following commands to disable automatic FPD upgrade.

Example:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#fpd auto-upgrade disable
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:ios(config)#end
```

Upgrade FPDs Manually

Use the following procedure to upgrade the FPDs manually.

Note The Golden FPDs cannot be upgraded using the CLI.

Procedure

Step 1 Use the **show hw-module fpd** command to display information about the current FPD version.

You can use this command to determine if you must upgrade the FPD.

Step 2 Use the **show alarms brief system active** command to display the active alarms.

You must upgrade the FPD when the **One Or More FPDs Need Upgrade Or Not In Current State** alarm is present.

Step 3 Use the **upgrade hw-module location** *[location-id]* **fpd** *[fpd name]* command to upgrade a specific FPD.

After upgrading the FPD, the user must wait for upgrade completion. The progress of the FPD upgrade can be monitored using the **show hw-module fpd** command.

Example:

```
RP/0/RP0/CPU0:ios#upgrade hw-module location 0/Rack fpd IoFpga
```

Note
The FPDs of power modules belong to 0/PM0 and 0/PM1 locations. The FPDs belonging to both the PM locations cannot be simultaneously upgraded.

Step 4 Use the **reload location** *location-id* to reload the FPDs belonging to a specific location with the new version.

The **Reload Loc** field in the output of **show hw-module fpd** command indicates whether the reload is required or not.

Example:

```
RP/0/RP0/CPU0:ios#reload location 0/RP0/CPU0
```

Step 5 (Optional) Use the **upgrade hw-module location all fpd all** command to upgrade all the FPDs at once.

Step 6 (Optional) Use the **upgrade hw-module [location** [*location-id* | **all**]] **fpd** [*fpd name*] | **all**] command to upgrade a specific FPD, all the FPDs, or the FPDs belonging to a specific location.

Example:

```
RP/0/RP0/CPU0:ios#upgrade hw-module location all fpd all
```

Note
The FPDs of power modules and SSDs cannot be forcefully upgraded.

Upgrading FPDs Using Yang Data Models

YANG is a data modeling language that helps to create configurations, retrieve operational data and execute actions. NCS 1010 acts on the data definition when these operations are requested using NETCONF RPCs. The data model handles the following types of requirements on NCS 1010 for FPD:

Operational Data	Native Data Model
Auto Upgrade: Enabling or disabling of automatic upgrade of FPD	Cisco-IOS-XR-fpd-infra-cfg.yang
Auto Reload: Enabling or disabling of automatic reload of FPD	Cisco-IOS-XR-fpd-infra-cfg.yang

Check System Stability

System stability checks are essential to measure the efficiency and ability of an upgrade to function over an extended period.

At the EXEC prompt, execute the following commands to assess basic system stability checks before and after the software upgrade.

Command	Reason	Workaround
`show platofrm`	Verify that all nodes are in `IOS XR RUN/OPERATIONAL` state	NA

Command	Reason	Workaround

`show ipv4 interface brief`
Or
`show ipv6 interface brief`
Or
`show interfaces summary`
	Verify that all necessary interfaces are `UP`	NA

`show install active summary`	Verify that the proper set of packages are active	NA
`show install committed summary`	Verify that the proper set of committed packages are same as active	Execute 'install commit' command
`clear configuration inconsistency`	Verify/fix configuration file system	NA
`show hw-module fpd`	Ensure all the FPD versions status are CURRENT	Execute `upgrade hw-module fpd` command
`show media`	Display the current state of the disk storage media	To free up space, remove older .iso image files and bug fix .tar files.
`show media | i rootfs`	Display the current state of the root filesystem (rootfs). By default, the following files are stored in **rootfs**: • Older config commits • Older .iso image and .tar files for SMUs • All the extracted .tar files	The installation is blocked if it utilizes more than 92% of the disk space on the **rootfs**. To avoid this, we recommend maintaining: • Twice the free space of the .iso image file size when installing the software • At least two and a half times the size of the .tar file when installing SMUs To free up space in **rootfs**: • use the **clear install rollback id** *id* to remove older rollback points • consider storing all user data in the **harddisk:/** location
`show inventory`	Show chassis inventory information	NA
`show logging`	Capture show logging to check for any errors	NA

Obtain Install Files

You can obtain the install files based on one of the following options that is best suited to your network:

- **Golden ISO:** You can build a customized golden ISO (GISO) image with the base ISO and the required RPMs to automatically upgrade the software.
- **Base ISO and Optional RPMs:** You can upgrade the software through the standard method where you install the ISO followed by the required RPMs.

Standard ISO and RPMs

Download Install Files from Cisco Software Center

Obtain the install files (base ISO and RPMs) for the target release.

Procedure

Step 1 Access the Cisco Software Download page.

For optimum website experience, we recommend any of the following browsers: Google Chrome, Mozilla Firefox or Internet Explorer.

Step 2 Click **Browse All** and navigate to NCS 1010 using **Optical Networking** > **Optical Data Center Interconnects** > **Network Convergence System 1000 Series** > **Network Convergence System 1010**.

Step 3 Select the Software Type: IOS XR Software or IOS XR Software Maintenance Upgrades (SMU).

Step 4 From the left pane, select the release.

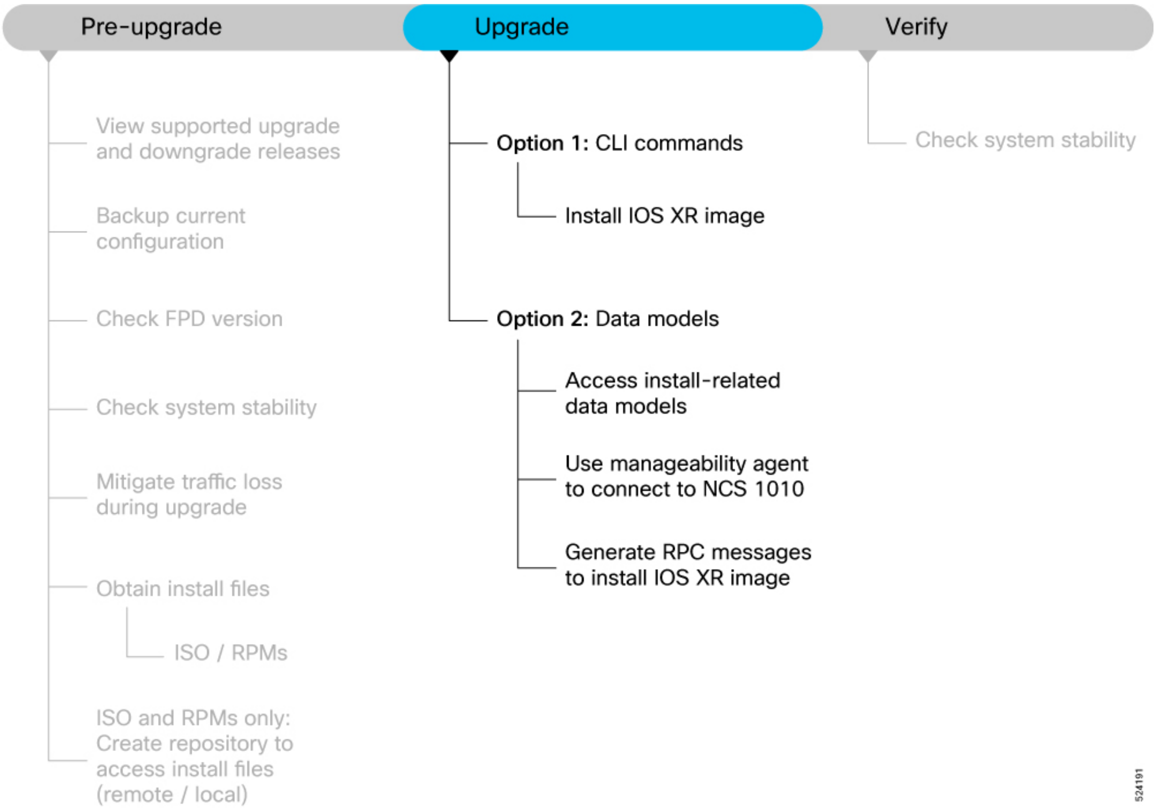
For the selected release, the Software Download page displays the downloadable files. For more information, see Install ISO and RPMs, on page 72.

Step 5 Use your Cisco login credentials to download the files.

Upgrade the Software

This section provides information about the processes involved in upgrading the IOS XR software on NCS 1010

Figure 10: NCS 1010 Upgrade workflow


Pre-upgrade
Upgrade
Verify
View supported upgrade and downgrade releases
Backup current configuration
Check FPD version
Check system stability
Mitigate traffic loss during upgrade
Obtain install files
ISO / RPMs
ISO and RPMs only: Create repository to access install files (remote / local)
Option 1: CLI commands
Install IOS XR image
Option 2: Data models
Access install-related data models
Use manageability agent to connect to NCS 1010
Generate RPC messages to install IOS XR image
Check system stability
524191

The Cisco IOS XR software can be upgraded using one of these methods:

- Upgrade NCS 1010 Using CLI Commands
- Upgrade NCS 1010 Using YANG Data Models

View Supported Upgrade and Downgrade Releases

This section provides the supported upgrade and downgrade paths for NCS 1010. Consider the following guidelines when performing an upgrade or downgrade:

- When downgrading the software image from release 24.4.x to an earlier version, we recommend to manually downgrade the line card firmware as well to prevent any impact on various functionalities.
- It is recommended to perform the upgrade with FPD auto-upgrade enabled to ensure the FPD versions are up to date and to prevent potential upgrade-related issues.

The following table lists the upgrade and downgrade paths supported for Cisco NCS 1010.

Upgrade Path			Downgrade Path		
Source Release	Destination Release	Bridge SMU	Source Release	Destination Release	Bridge SMU
• R7.10.1 • R7.11.2 • R24.2.11 • R24.3.1 • R24.4.15	R25.1.1	No	R25.1.1	• R24.4.15 • R24.3.1 • R24.2.11 • R7.11.2 • R7.10.1	No

Upgrade NCS 1010 Using CLI Commands

There are two options to upgrade your Cisco IOS XR software using the Command Line Interface (CLI):

- Base ISO and optional RPMs
- Golden ISO (GISO)

Install ISO and RPMs

Use this procedure to install the base ISO and optional RPMs.

Procedure

Step 1 Copy the ISO image to be installed either on the NCS 1010 hard disk or on a network server to which NCS 1010 has access.

Example:

```
RP/0/RP0/CPU0:ios#scp root@<ip-address>:/<dir>/1010-x64-release.iso harddisk:
```

Step 2 To verify data integrity, verify the md5 checksum of the copied file with the original MD5 values on CCO.

Example:

```
RP/0/RP0/CPU0:ios#show md5 file /harddisk:/1010-x64-release.iso
```

Step 3 Install the base image to upgrade the system.

- **Option 1:** Install ISO without control over reload timing.

  ```
  RP/0/RP0/CPU0:ios#install replace /harddisk:/1010-x64-release.iso
  ```

  The image is installed, the changes are applied through a reload or a restart of the system, and commits the changes. However, you do not have control over the timing of the reload or restart —these occur as soon as the package operation completes and the system is ready.

  If you want to control when your system reloads (management of a network outage), we recommend that you schedule an upgrade window and perform an **install replace**, letting the system reload without intervention.

- **Option 2:** Install ISO with control over reload timing.

  a. Install the image.

  ```
  RP/0/RP0/CPU0:ios#install package replace /harddisk:/1010-x64-release.iso
  ```

  b. Apply the changes.

  ```
  RP/0/RP0/CPU0:ios#install apply [reload | restart]
  ```

  You can use either the `reload` or `restart` options based on the file that is installed. To determine whether a `reload` or `restart` is required, check the output of **show install request** command. The output indicates the required actions.

Step 4 After the base image is upgraded, install the additional packages. For more information, see Install Additional RPMs and Bug Fixes, on page 97.

If a system fails to boot successfully, or reboots unexpectedly when the package is undergoing a version change, the system is automatically recovered to its old software state.

Note
If you perform a manual or automatic system reload without completing the transaction with the **install commit** command, the action will revert the system to the point before the install transaction commenced, including any configuration changes. Only the log is preserved for debugging.

Install Golden ISO

Use this procedure to install the Golden ISO (GISO) that contains the base ISO and a customized list of optional RPMs..

Golden ISO (GISO) upgrades NCS 1010 to a version that has a predefined list of bug fixes (sometimes also called software maintenance updates) with a single operation.

To update the system to the same release version with a different set of bug fixes:

- Create a GISO with the base version and all the bug fixes you require
- Use the **install replace** or **install package replace** commands to install the GISO.

The GISO can include bridging bug fixes for multiple source releases, and installs only the specific bridging bug fixes required for the target release.

The bridging bug fix RPMs can be used in the following scenarios:

- To resolve a bug that might stop upgrade.
- To meet the prerequisite requirements of a new release version that were not met by the earlier version.

Note The **install replace** command is supported only with GISO, but not with .rpm packages directly.

Procedure

Step 1 Copy the GISO image file to the /harddisk: of NCS 1010.

Example:

```
RP/0/RP0/CPU0:ios#scp root@<ip-address>:/auto/tftp-test/1010-x64-release.iso harddisk:
```

Step 2 Install the GISO.

- **Option 1:** Install GISO without control over reload timing.

  a. Install GISO to upgrade to a new release, add or remove bugfixes or optional packages.

  ```
  RP/0/RP0/CPU0:ios#install replace source-location/giso-name.iso
  ```

  The *source-location* can be in the following location.

  b. Local path to the GISO—files located in or under `/var/xr/disk1/`, `/harddisk:/` or `/misc/disk1/`

  This command runs the replace operation and applies the new version via NCS 1010 restart or reload, whichever is least impactful, given the change. For example, if you have a GISO that is the same as your base image except one bugfix, and that bugfix can be applied by process restart, the command will install the bugfix and apply by restart, no NCS 1010 reload occurs. However, you do not have control over the timing of the reload or restart—these operations occur as soon as the packaging is complete and the system is ready. If you want to control the timing of system reloads, we recommend that you schedule an upgrade window and run the **install replace** command, allowing the system to reload without manual intervention or network impact.

  c. [Optional] Specify **reload** keyword to force reload for all operations. This may be useful if you want a reliable flow.

  d. [Optional] Specify **commit** keyword for the install, apply and commit operations to be performed without user intervention.

- **Option 2:** Install GISO with control over reload timing.

  a. Install GISO to upgrade to a new release, add or remove bugfixes or optional packages. The functionality is similar to **install replace** command, except that the staging of packaging changes is performed using this command.

  ```
  RP/0/RP0/CPU0:ios#install package replace source-location/giso-name.iso
  ```

  The **install package replace** command does not apply the changes.

  b. Apply the changes.

  ```
  RP/0/RP0/CPU0:ios#install apply [reload | restart]
  ```

  You can use either the `reload` or `restart` options based on the change that is installed. You can only apply the changes by restarting the software if the difference between the GISO being installed and the running image is minimal such as bugfixes or package updates.

  To determine whether a `reload` or `restart` is required, check the output of **show install request** command. The output indicates the required actions.

Note

A GISO label is a string that identifies a GISO. Any install operation, such as adding or removing a package or modifying the software image (replace or package replace) will change the custom label to a system-generated default label. For example:

```
RP/0/RP0/CPU0:ios#show install active summary

...
```

In this example, the software image is modified to remove the CDP package.

```
RP/0/RP0/CPU0:ios#install package remove xr-cdp

Install remove operation 39.1.1 has started
Install operation will continue in the background
...
Packaging operation 39.1.1: 'install package remove xr-cdp' completed without error
```

Apply the changes.

```
RP/0/RP0/CPU0:ios#install apply
Thu Feb 02 11:13:09.015
Once the packaging dependencies have been determined, the install operation may have to reload the
system.
If you want more control of the operation, then explicitly use 'install apply restart' or 'install
apply reload' as
reported by 'show install request'.
Continue? [yes/no]:[yes] yes
RP/0/RP0/CPU0:Feb 02 11:13:12.771 : instorch[404]: %INSTALL-6-ACTION_BEGIN : Apply by restart 39.1
started
Install apply operation 39.1 has started
Install operation will continue in the background
```

View the software version.

```
RP/0/RP0/CPU0:ios#show version
```

The `GISO1` custom label is replaced with the label `24.3.1` generated by the system.

Upgrade Using Data Models

Data models are a programmatic way of configuring and collecting operational data of a network device. They replace the process of manual configuration and can be used to automate configuration tasks across heterogeneous devices in a network.

Access Install-related Data Models

You can use YANG data models to install and upgrade NCS 1010. The data models are packaged with the release image in the `/pkg/yang` directory.

Procedure

Step 1 Navigate to the directory in the release image where the YANG data models are available.

Example:

```
RP/0/RP0/CPU0:ios#run
[node_RP0_CPU0:~]$cd /pkg/yang
```

Step 2 View the list of install-related data models on NCS 1010.

Example:

```
node0_RP0_CPU0:/pkg/yang]$ls -ltr *install*
-rw-r--r--.   1   root   root   8646   Jul   2   01:59   Cisco-IOS-XR-install-act.yang
-rw-r--r--.   1   root   root   7267   Jul   2   01:59   Cisco-IOS-XR-install-search-act.yang
-rw-r--r--.   1   root   root   10664  Jul   2   01:59   Cisco-IOS-XR-install-augmented-act.yang
-rw-r--r--.   1   root   root   2511   Jul   2   02:00   Cisco-IOS-XR-um-install-cfg.yang
-rw-r--r--.   1   root   root   2270   Jul   2   02:04   Cisco-IOS-XR-install-cfg.yang
-rw-r--r--.   1   root   root   6222   Jul   2   02:04   Cisco-IOS-XR-install-oper.yang
-rw-r--r--.   1   root   root   14009  Jul   2   02:04
Cisco-IOS-XR-install-augmented-oper.yang
```

The following table describes the function of the install-related data models:

Date Model	Description
Cisco-IOS-XR-um-install-cfg	Unified data model that contains a collection of YANG definitions for Cisco IOS XR install package configuration, and augments the modules with configuration data.
Cisco-IOS-XR-install-oper	Operational data model to view details that are related to basic package information, active and committed packages, and fixes.
Cisco-IOS-XR-install-cfg	Configuration data model to specify the location of the install source.
Cisco-IOS-XR-install-act	Action model to perform basic install operations and software upgrade.
Cisco-IOS-XR-install-search-act	Action model that contains a collection of YANG definitions for install actions related to searching for package information.
Cisco-IOS-XR-install-augmented-oper	Augmented operational model that displays information aboutpackaging, atomic changes, and history of the install operation on the NCS 1010.
Cisco-IOS-XR-install-augmented-act	Action model to perform flexible install operations, includingcontrolling the exact timing of system reloads and rolling back to a previous commit.
Cisco-IOS-XR-shellutil-copy-act	Action model to copy files on the NCS 1010 from a source location.

You can also access the supported data models to install Cisco IOS XR software from the Github repository.

Use Manageability Agent to Connect to NCS 1010

Use a manageability agent like NETCONF or gRPC to connect and communicate with NCS 1010. You can send Remote Procedure Calls (RPC) requests to configure or retrieve operational data from NCS 1010. NCS 1010 processes the request and responds to the request through an RPC response. You use the RPCs to send requests to install the software by populating the relevant parameters of a container and leaf in the data model. For more information about understanding the data model structure and using data models, see the *Cisco NCS 1010 Data Models Configuration Guide* .

Generate RPC Messages to Install IOS XR Image

Before you begin

Not all software versions are supported as the target upgrade software version. You must review the supported upgrade and downgrade paths, hardware or software limitations, and bridging SMUs required for the version. For more information about checking the release support between the current and target versions, see View Supported Upgrade and Downgrade Releases, on page 60.

Procedure

Step 1 Use the `install-replace` RPC on the `Cisco-IOS-XR-install-act.yang` data model to upgrade NCS 1010.

Step 2 Configure the values of the `source-type`, `source`, and `file` parameters.

Step 3 Send `edit-config` NETCONF RPC request using the data model to configure the repository. Edit the values in the `repositories` parameters and send this request to NCS 1010 from the client.

Step 4 Apply the changes to activate the ISO on NCS 1010 using RPCs by using the `install-apply` RPC on the`Cisco-IOS-XR-install-augmented-act.yang` datamodel and send the RPC from the client to NCS 1010.

Example:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <install-apply xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-install-augmented-act">
   <apply-method>least-impactful</apply-method>
  </install-apply>
</rpc>
```

View the RPC response received from NCS 1010.

```
<?xml version="1.0"?>
  <rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <op-id xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-install-augmented-act">2.1</op-id>
  </rpc-reply>
```

In the response, NCS 1010 sends an ID indicating that the changes are applied successfully.

Step 5 Verify that the software upgrade is successful. Use the `get`RPCon `Cisco-IOS-XR-install-oper.yang` data model. Edit the `install` parameter and send an RPC request from the client to NCS 1010.

Example:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <get>
   <filter>
    <install xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-install-oper">
     <request/>
    </install>
   </filter>
  </get>
</rpc>
```

View the RPC response received from NCS 1010.

```
<?xml version="1.0"?>
  <rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <data>
    <install xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-install-oper">
     <request>
      <request>install commit</request>
      <state>success</state>
     <timestamp>2022-06-27 T02:52:07Z</timestamp>
    <operation-id>26</operation-id>
   </request>
  </install>
```

The state of the install operation in the RPC response indicates that the software and the RPMs are upgraded successfully.

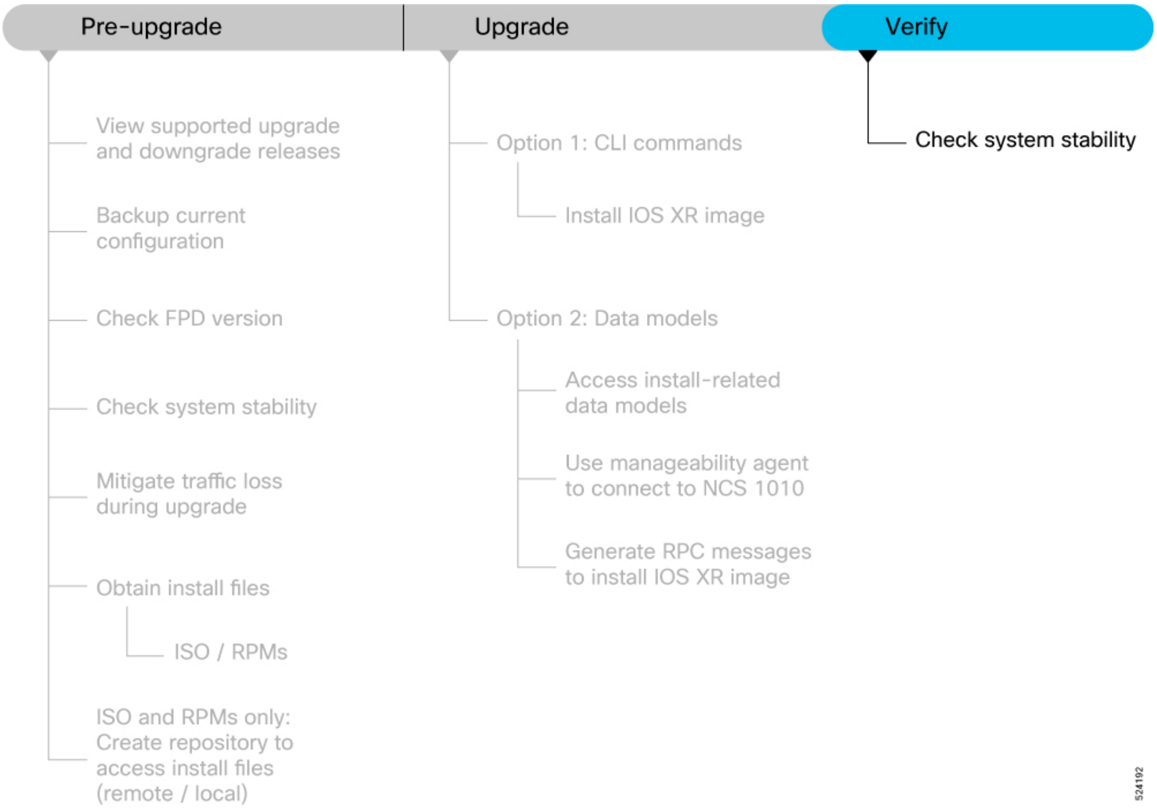
What to do next

Perform prelimiary checks to verify that NCS 1010 is upgraded successfully.

Verify the Software Upgrade

This section provides information about the processes involved in verifying the upgraded software on your NCS 1010.

Figure 11: Workflow to Verify the Software Upgrade


Pre-upgrade
Upgrade
Verify
View supported upgrade and downgrade releases
Backup current configuration
Check FPD version
Check system stability
Mitigate traffic loss during upgrade
Obtain install files
ISO / RPMs
ISO and RPMs only: Create repository to access install files (remote / local)
Option 1: CLI commands
Install IOS XR image
Option 2: Data models
Access install-related data models
Use manageability agent to connect to NCS 1010
Generate RPC messages to install IOS XR image
Check system stability
524192

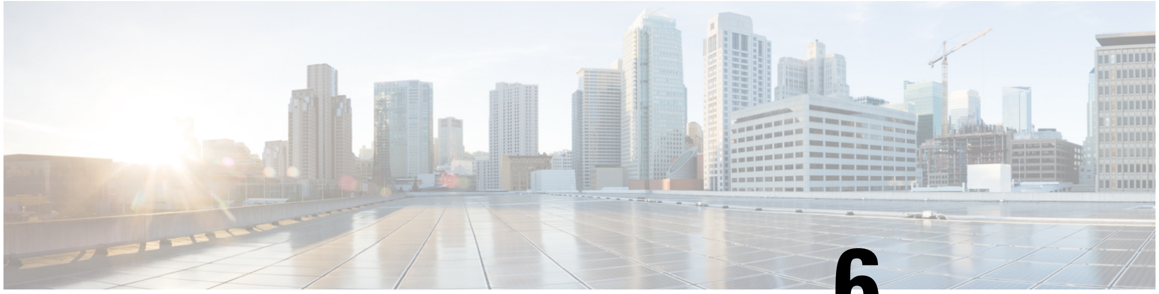
This section contains the following topics:

Check System Stability

System stability checks are essential to measure the efficiency and ability of an upgrade to function over an extended period.

At the EXEC prompt, execute the following commands to assess basic system stability checks before and after the software upgrade.

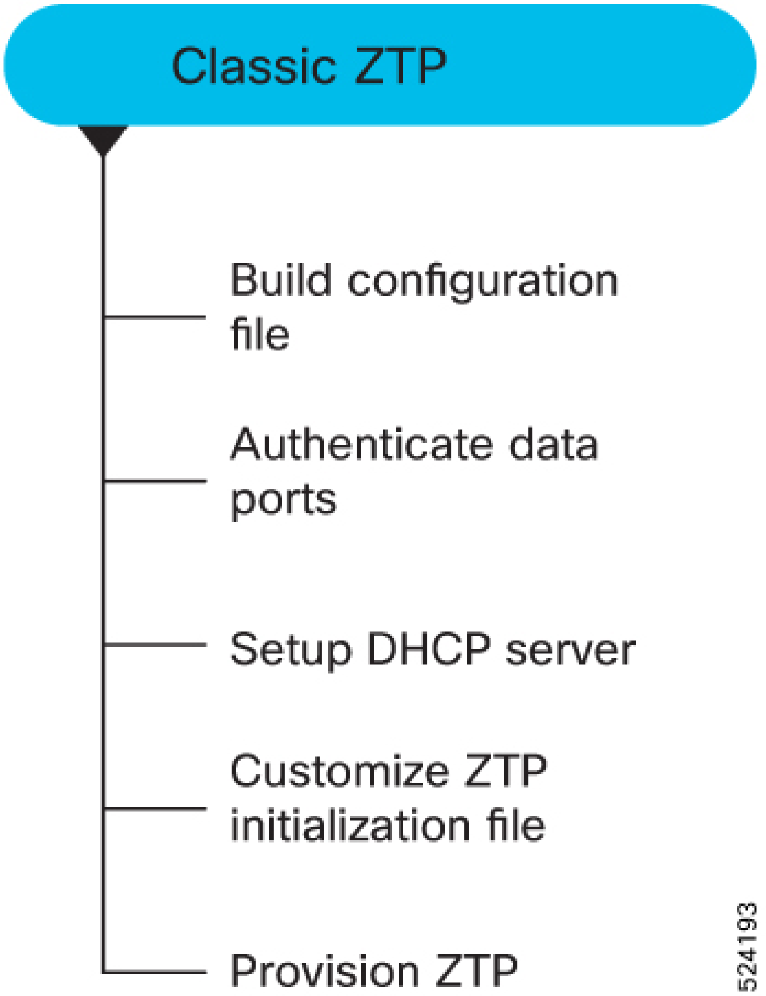
Command	Reason	Workaround
show platform	Verify that all nodes are in `IOS XR RUN/OPERATIONAL` state	NA
show redundancy	Verify that a standby RP is available, and the system is in `NSR-ready` state	NA
show install active summary	Verify that the proper set of packages are active	NA
show install committed summary	Verify that the proper set of committed packages are same as active	Execute 'install commit' command
clear configuration inconsistency	Verify/fix configuration file system	NA
show hw-module fpd	Ensure all the FPD versions status are CURRENT	Execute `upgrade hw-module fpd` command
show media	Display the current state of the disk storage media	To free up space, remove older .iso image files and bug fix .tar files.
show inventory	Show chassis inventory information	NA



CHAPTER 6

Deploy NCS 1010 Using Classic ZTP

Figure 12: Classic ZTP Work Flow



Fresh Boot Using DHCP

The ZTP process initiates when you boot the network-device with an IOS-XR image. The process starts only on the device that doesn't have a prior configuration.

This image depicts the high-level work flow of the ZTP process:

1. ZTP sends DHCP request to fetch the ZTP configuration file or user script. To help the Bootstrap server uniquely identify the device, ZTP sends below DHCP option.

   - DHCP(v4/v6) client-id=Serial Number
   - DHCPv4 option 124: Vendor, Platform, Serial-Number
   - DHCPv6 option 16: Vendor, Platform, Serial-Number

   The following is the default sequential flow of the ZTP process:

   - ZTP sends IPv4 DHCP request first on all the management port. In case there is a failure, then ZTP sends IPv6 DHCP request on all the management port.
   - ZTP sends IPv4 DHCP request first on all the data port. In case there is a failure, then ZTP sends IPv6 DHCP request on all the data port.

     The default sequential flow is defined in configuration file and you can modify the sequence using the configuration file.

2. DHCP server identifies the device and responds with DHCP response using one of the following options:

   DHCP server should be configured to respond with the DHCP options.

   - DHCPv4 using BOOTP filename to supply script/config location.
   - DHCPv4 using Option 67 (bootfile-name) to supply script/config location.
   - DHCPv6 using Option 59 (OPT_BOOTFILE_URL) to supply script/config location

3. The network device downloads the file from the web server using the URL location that is provided in the DHCP response.

4. The device receives a configuration file or script file from the HTTP server.

Note

- If the downloaded file content starts with !! IOS XR it is considered as a configuration file.
- If the downloaded file content starts with #! /bin/bash, #! /bin/sh or #!/usr/bin/python it is considered as a script file.

5. The device applies the configuration file or executes the script or binary in the default bash shell.
6. The Network device is now up and running.

Build your Configuration File

Based on the business need, you can use a configuration or script file to initiate the ZTP process.

The configuration file content starts with !! IOS XR.

The following is the sample configuration file. You can automate all the configurations. For more information on creating ZTP configuration file, refer ZTP Configuration Files Creation.

```
Tue May 4 18:08:59.544 UTC
Building configuration...
!! IOS XR Configuration 192.0.2.254I
!! Last configuration change at Tue May 4 17:12:47 2021 by cisco
!
line console
exec-timeout 0 0
!
line default
exec-timeout 0 0
session-timeout 0
!
vty-pool default 0 20
alias alarms show alarms brief system active
interface MgmtEth0/RP0/CPU0/0
ipv4 address dhcp
no shut
!
interface MgmtEth0/RP0/CPU0/1
description noshut-interface-ztp
ipv4 address 192.0.2.255 255.255.255.0
no shut
!
interface MgmtEth0/RP0/CPU0/2
description noshut-interface-ztp
no shut
!
interface PTP0/RP0/CPU0/0
description noshut-interface-ztp
no shut
Cisco NCS 1010 System Setup and Software Installation Guide, IOS XR Release 7.7.x
19
Bring-up Cisco NCS 1010
Build your Configuration File
!
telnet vrf default ipv4 server max-servers 100a
ssh server v2
ssh server netconf vrf default
```

```
netconf-yang agent
ssh
!
netconf agent tty
grpc
ncs1010 static
address-family ipv4 unicast
0.0.0.0/0 192.0.2.255
end
```

Configure ZTP BootScript

ZTP downloads and executes the script files. These script files include a programmatic approach to complete a task. For example, scripts created using IOS XR commands to perform patch upgrades. The first line of the file must contain #! /bin/bash or #! /bin/sh for ZTP to process the file as script. You can either use the ZTP bash script or the ZTP configuration file.

You can either use the ZTP bash script or the ZTP configuration file.

If you want to hardcode a script to be executed every boot, configure the following.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ztp bootscript /disk0:/myscript
RP/0/RP0/CPU0:ios(config)#commit
```

The above configuration waits for the first data-plane interface to be configured and then wait an extra minute for the management interface to be configured with an IP address, to ensure that we have connectivity in the third-party namespace for applications to use. If the delay is not desired, use:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ztp bootscript preip /disk0:/myscript
RP/0/RP0/CPU0:ios(config)#commit
```

Note When the above command is first configured, you will be prompted if you wish to invoke it now. The prompt helps with testing.

This is the example content of **/disk0:/myscript**:

```
host ncs1010_P1B_DT_08_ETH0 {
#hardware ethernet 68:9e:0b:b8:6f:5c ;
option dhcp-client-identifier "FCB2437B05N" ;
if exists user-class and option user-class = "iPXE" {
filename "http://192.0.2.1/P1B_DT_08/ncs1010-x64.iso";
} else {
filename "http://192.0.2.1/P1B_DT_08/startup.cfg";
}
fixed-address 203.0.113.254;
}
```

The following is the sample content of the ZTP bash script.

```
#! /bin/bash
#
# NCS1010 Demo Sample
# ZTP installation of config and day-0 SMU's
#
```

```
source ztp_helper

wget http://downloads.sourceforge.net/project/yourcode/application.tgz
#install the downloaded application.tgz

#Run XR CLI’s from the script
`xrcmd “show version”`
```

The following is the sample content of the ZTP configuration file.

```
Tue May 4 18:08:59.544 UTC
Building configuration...
!! IOS XR Configuration 203.0.113.254
!! Last configuration change at Tue May 4 17:12:47 2021 by cisco
!
line console
exec-timeout 0 0
!
line default
exec-timeout 0 0
session-timeout 0
!
vty-pool default 0 20
alias alarms show alarms brief system active
interface MgmtEth0/RP0/CPU0/0
ipv4 address dhcp
no shut
!
interface MgmtEth0/RP0/CPU0/1
description noshut-interface-ztp
ipv4 address 192.0.2.255 255.255.255.0
no shut
!
interface MgmtEth0/RP0/CPU0/2
description noshut-interface-ztp
no shut
!
interface PTP0/RP0/CPU0/0
description noshut-interface-ztp
no shut
!
interface PTP0/RP0/CPU0/1
description noshut-interface-ztp
no shut
end
```

Invoke ZTP Manually through CLI

Manual ZTP can be invoked through CLI commands. This manual way helps you to provision the NCS 1010 in stages. Ideal for testing out ZTP configuration without a reboot. If you want to invoke a ZTP on an interface (data ports or management port), you don't have to bring up and configure the interface first. You can execute the ztp initiate command, even if the interface is down, ZTP script brings it up and invoke dhclient. So ZTP could run over all interfaces no matter it is up or down.

Use the ztp initiate, ztp terminate, and ztp clean commands to force ZTP to run over more interfaces.

- ztp initiate—Invokes a new ZTP DHCP session. Logs can be found in /disk0:/ztp/ztp.log.
- ztp terminate—Terminates any ZTP session in progress.
- ztp clean—Removes only the ZTP state files.

The log file ztp.log is saved in /var/log/ztp.log folder, and a copy of log file is available at /disk0:/ztp/ztp.log location using a soft link. However, executing ztp clean clears files saved on disk and not on /var/logztp.log folder where current ZTP logs are saved. In order to have a log from current ZTP run, you must manually clear the ZTP log file from /var/log/ztp.log folder.

Procedure

Step 1 (optional) **ztp clean**

Example:

```
RP/0/RP0/CPU0:ios#ztp clean
Fri Apr 29 06:49:29.760 UTC
This would remove all ZTP temporary files.
Would you like to proceed? [no]: yes
All ZTP operation files have been removed.
ZTP logs are present in /var/log/ztp*.log for logrotate.
Please remove manually if needed.
If you now wish ZTP to run again from boot, do 'conf t/commit replace' followed by reload.
```

Removes all the ZTP logs and saved settings.

Step 2 **ztp initiate**

Example:

```
RP/0/RP0/CPU0:ios#ztp initiate
Fri Jun 17 11:44:08.791 UTC
Initiating ZTP may change your configuration.
Interfaces might be brought up if they are in shutdown state
Would you like to proceed? [no]: yes
ZTP will now run in the background.
Please use "show logging" or look at /var/log/ztp.log to check progress.
RP/0/RP0/CPU0:ios#
```

Use the **show logging** command or see the /var/log/ztp.log to check progress.

Reboots the Cisco NCS 1010 system.

Step 3 (Optional) **ztp terminate**

Example:

```
RP/0/RP0/CPU0:ios#ztp terminate
Fri Apr 29 06:38:59.238 UTC
This would terminate active ZTP session if any (this may leave your system in a partially configured
 state)
Would you like to proceed? [no]: yes
Terminating ZTP
No ZTP process running
```

Terminates the ZTP process.

Invoke ZTP Through Reload

The ZTP process can be automatically invoked by using the reload command.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:P2B_DT_02#configure
```

Enters the configuration mode.

Step 2 **commit replace**

Warning
This operation erases the complete database of the NCS1010 and impacts the traffic.

Example:

```
Fri Apr 29 06:48:46.236 UTC
RP/0/RP0/CPU0:ios(config)#commit replace
Fri Apr 29 06:48:53.199 UTC

This commit will replace or remove the entire running configuration. This
operation can be service affecting.
Do you wish to proceed? [no]: yes
RP/0/RP0/CPU0:ios(config)#end
```

Removes the entire running configuration.

Step 3 **ztp clean**

Example:

```
RP/0/RP0/CPU0:ios#ztp clean
Fri Apr 29 06:49:29.760 UTC
This would remove all ZTP temporary files.
Would you like to proceed? [no]: yes
All ZTP operation files have been removed.
ZTP logs are present in /var/log/ztp*.log for logrotate.
Please remove manually if needed.
If you now wish ZTP to run again from boot, do 'conf t/commit replace' followed by reload.
```

Removes all the ZTP logs and saved settings.

Step 4 **reload**

Example:

```
RP/0/RP0/CPU0:ios#reload
Fri Apr 29 06:50:12.312 UTC
Proceed with reload? [confirm]

RP/0/RP0/CPU0:ios#
Preparing system for backup. This may take a few minutes especially for large configurations.
 Status report: node0_RP0_CPU0: BACKUP INPROGRESS
 Status report: node0_RP0_CPU0: BACKUP HAS COMPLETED SUCCESSFULLY
[Done]
```

After the node comes up, you can check that the ZTP is initiated and the configuration has been restored successfully.

```
RP/0/RP0/CPU0:Apr 29 06:55:33.242 UTC: pyztp2[377]: %INFRA-ZTP-4-CONFIG_INITIATED : ZTP has initiated
 config load and commit operations
RP/0/RP0/CPU0:Apr 29 06:55:39.263 UTC: ifmgr[381]: %PKT_INFRA-LINK-3-UPDOWN : Interface
GigabitEthernet0/0/0/0, changed state to Down
RP/0/RP0/CPU0:Apr 29 06:55:39.287 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR
:PROV-INPROGRESS :DECLARE :GigabitEthernet0/0/0/0:
RP/0/RP0/CPU0:Apr 29 06:55:39.287 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR
:PROV-INPROGRESS :DECLARE :Osc0/0/0/0:
RP/0/RP0/CPU0:Apr 29 06:55:39.287 UTC: ifmgr[381]: %PKT_INFRA-LINK-3-UPDOWN : Interface
GigabitEthernet0/0/0/0, changed state to Up
RP/0/RP0/CPU0:Apr 29 06:55:39.716 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR
:PROV-INPROGRESS :CLEAR :Osc0/0/0/0:
RP/0/RP0/CPU0:Apr 29 06:55:39.728 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR
:PROV-INPROGRESS :CLEAR :GigabitEthernet0/0/0/0:
RP/0/RP0/CPU0:Apr 29 06:55:47.904 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR
:PROV-INPROGRESS :DECLARE :Ots0/0/0/1:

User Access Verification

Username: cisco
Password:
ios con0/RP0/CPU0 is now available
```

Reboots the Cisco NCS 1010 system.

Authenticate Data Ports

On fresh boot, ZTP process is initiated from management ports and may switch to data ports. To validate the connection with DHCP server, authentication is performed on data ports through DHCP option 43 for IPv4 and option 17 for IPv6. These DHCP options are defined in option space and are included within **dhcpd.conf** and **dhcpd6.conf** configuration files. You must provide following parameters for authentication while defining option space:

- Authentication code—The authentication code is either 0 or 1; where 0 indicates that authentication is not required, and 1 indicates that MD5 checksum is required.

Note If the option 43 for IPv4, and option 17 for IPv6 is disabled, the authentication fails.

- Client identifier—The client identifier must be 'exr-config'.
- MD5 checksum—This is chassis serial number. It can be obtained using **echo -n $SERIALNUMBER | md5sum | awk '{print $1}'** .

Here is the sample **dhcpd.conf** configuration. In the example below, the option space called **VendorInfo** is defined with three parameters for authentication:

```
class "vendor-classes" {
   match option vendor-class-identifier;
}

option space VendorInfo;
option VendorInfo.clientId code 1 = string;
option VendorInfo.authCode code 2 = unsigned integer 8;
option VendorInfo.md5sum code 3 = string
option vendor-specific code 43 = encapsulate VendorInfo;
subnet 10.65.2.0 netmask 255.255.255.0 {
  option subnet-mask 255.255.255.0;
  option ncs 1010 10.65.2.1;
  range 10.65.2.1 10.65.2.200;
}
host cisco-mgmt {
   hardware ethernet 00:50:60:45:67:01;
   fixed-address 10.65.2.39;
   vendor-option-space VendorInfo;
   option VendorInfo.clientId "exr-config";
   option VendorInfo.authCode 1;
   option VendorInfo.md5sum "aedf5c457c36390c664f5942ac1ae3829";
   option bootfile-name "http://10.65.2.1:8800/admin-cmd.sh";
}
```

Here is the sample **dhcpd6.conf** configuration file. In the example below, the option space called **VendorInfo** is defined that has code width 2 and length width 2 (as per dhcp standard for IPv6) with three parameters for authentication:

```
log-facility local7;
option dhcp6.name-servers 2001:1451:c632:1::1;
option dhcp6.domain-search "cisco.com";
dhcpv6-lease-file-name "/var/lib/dhcpd/dhcpd6.leases";
option dhcp6.info-refresh-time 21600;
option dhcp6.bootfile-url code 59 = string;
option dhcp6.user-class code 15 = string;
option space CISCO-EXR-CONFIG code width 2 length width 2;
option CISCO-EXR-CONFIG.client-identifier code 1 = string;
option CISCO-EXR-CONFIG.authCode code 2 = integer 8;
option CISCO-EXR-CONFIG.md5sum code 3 = string;
option vsio.CISCO-EXR-CONFIG code 9 = encapsulate CISCO-EXR-CONFIG;
subnet6 2001:1451:c632:1::/64{
 range6 2001:1451:c632:1::2 2001:1451:c632:1::9;
 option CISCO-EXR-CONFIG.client-identifier "exr-config";
 option CISCO-EXR-CONFIG.authCode 1;
 #valid md5
 option CISCO-EXR-CONFIG.md5sum "90fd845ac82c77f834d57a034658d0f0";
 if option dhcp6.user-class = 00:04:69:50:58:45 {
  option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/cisco-2/image.iso";
 }
 else {
   #option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/cisco-2/cisco-mini-x.iso.sh";
   option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/cisco-2/ztp.cfg";
  }
}
```

Setup DHCP Server

For ZTP to operate a valid IPv4 or IPv6 address is required and the DHCP server must send a pointer to the configuration script.

The DHCP request from the NCS 1010 has the following DHCP options to identify itself:

- **Option 60**: “vendor-class-identifier” : Used to Identify the following four elements:
  - The type of client: For example, PXEClient

  - The architecture of The system (Arch): For example: 00009 Identify an EFI system using a x86-64 CPU
  - The Universal Network Driver Interface (UNDI):

    For example 003010 (first 3 octets identify the major version and last 3 octets identify the minor version)
  - The Product Identifier (PID):

- **Option 61**: “dhcp-client-identifier” : Used to identify the Serial Number of the device.
- **Option 66** : Used to request the TFTP server name.
- **Option 67**: Used request the TFTP filename.
- **Option 97**: “uuid” : Used to identify the Universally Unique Identifier a 128-bit value (not usable at this time)

Example

The following DHCP request sample provides a fixed IP address and a configuration file with the mac address of the management interface.

```
host cisco-rp0 {
   hardware ethernet e4:c7:22:be:10:ba;
   fixed-address 172.30.12.54;
   filename "http://172.30.0.22/configs/cisco-1.config";
}
```

The following DHCP request sample provides a fixed IP address and a configuration file with the mac address of the management interface along with capability to re-image the system using iPXE (exr-config "xr-config" option):

```
host cisco-rp0 {
   hardware ethernet e4:c7:22:be:10:ba;
   fixed-address 172.30.12.54;
   if exists user-class and option user-class = "iPXE" {
      filename = "http://172.30.0.22/boot.ipxe";
   } elsif exists user-class and option user-class = "exr-config"  {
      filename = "http://172.30.0.22/scripts/cisco-rp0_ztp.sh";
   }
}
```

DHCP server identifies the device and responds with either an IOS-XR configuration file or a ZTP script as the filename option.

The DHCP server responds with the following DHCP options:

- DHCPv4 using BOOTP filename to supply script/config location.
- DHCPv4 using Option 67 (bootfile-name) to supply script/config location.
- DHCPv6 using Option 59 (OPT_BOOTFILE_URL) to supply script/config location

The following sample shows the DHCP response with bootfile-name (option 67):

```
option space cisco-vendor-id-vendor-class code width 1 length width 1;
option vendor-class.cisco-vendor-id-vendor-class code 9 = {string};

######### Network 11.11.11.0/24 ################
```

```
shared-network 11-11-11-0 {

####### Pools ##############
  subnet 11.11.11.0 netmask 255.255.255.0 {
    option subnet-mask 255.255.255.0;
 option broadcast-address 192.0.2.255;
 option ncs 1010 198.51.100.254;
 option domain-name-servers 198.51.100.254;
 option domain-name "cisco.local";
 # DDNS statements
   ddns-domainname "cisco.local.";
 # use this domain name to update A RR (forward map)
   ddns-rev-domainname "in-addr.arpa.";
   # use this domain name to update PTR RR (reverse map)

    }

######## Matching Classes ##########

  class "cisco" {
       match if (substring(option dhcp-client-identifier,0,11) = "FGE194714QS");
  }


  pool {
      allow members of "cisco";
      range 203.0.113.1 203.0.113.4;
      next-server 198.51.100.254;

      if exists user-class and option user-class = "iPXE" {
         filename="http://198.51.100.254:9090/cisco-mini-x-6.2.25.10I.iso";
      }


      if exists user-class and option user-class = "exr-config"
       {
          if (substring(option vendor-class.cisco-vendor-id-vendor-class,19,99)="cisco")
          {
            option bootfile-name
"http://198.51.100.254:9090/scripts/exhaustive_ztp_script.py";
          }
        }


      ddns-hostname "cisco-local";
      option ncs 1010 198.51.100.254;
  }
}
```

Customize ZTP Initialization File

You can customize the following ZTP configurable options in the *ztp.ini* file:

- `ZTP`: You can enable or disable ZTP at boot using CLI or by editing the *ztp.ini* file.
- `Retry`: Set the ZTP DHCP retry mechanism: The available values are infinite and once.
- `Fetcher Priority`: Fetcher defines which port ZTP should use to get the provisioning details. By default, each port has a fetcher priority defined in the *ztp.ini* file. You can modify the default priority of the fetcher. Allowed range is from 0 to 9.

Note Lower the number higher the priority. The value 0 has the highest priority and 9 has the lowest priority.

By default, the USB port has the higher priority.

In the following example, the Mgmt4 port has the highest priority:

```
[Fetcher Priority]
Mgmt4: 0
Mgmt6: 1
DPort4: 2
DPort6: 3
```

- `progress_bar`: Enable progress bar on the console. By default, the progress bar is disabled. To enable the progress bar, add the following entry in the `ztp.ini` file.

```
[Options]
progress_bar: True
```

- `config_check`: Saves ZTP configuration hashes in the **`/disk0:/ztp/`** location on the NCS 1010. By default, the config check is disabled. To enable the config check, add the following entry in the`ztp.ini` file.

```
[Startup]
start:         True
retry_forever: True
config_check: True
```

You can view the ZTP hashes by using the **show ztp log** command as seen below:

```
RP/0/RP0/CPU0:ios# show ztp log

============================== /var/log/ztp.log ==============================

2023-03-14 12:51:29,251 53612 [Configuration] INF: Provisioning via config replace
2023-03-14 12:51:43,131 53612 [Configuration] INF: Configuration has been applied
2023-03-14 12:51:43,131 53612 [Env ] DEB: cfg::createRefOnConfigCommit: called
2023-03-14 12:51:44,218 53612 [Env ] DEB: cfg:: Generating hash for File name:
/disk0:/ztp/customer/config.inithash_tmp
2023-03-14 12:51:44,218 53612 [Env ] DEB: cfg::_generateCfgAndSaveHash:: HASH :
c7980cfc23a401bbbf296e3d49c76bf9, type : 1
2023-03-14 12:51:59,715 53612 [Env ] DEB: cfg:: Generating hash for File name:
/disk0:/ztp/customer/config.successhash_tmp
..........
2023-03-14 12:51:59,715 53612 [Env ] DEB: cfg::_generateCfgAndSaveHash:: HASH :
c7980cfc23a401bbbf296e3d49c76bf9, type : 2
2023-03-14 12:52:04,901 53612 [Env ] DEB: cfg::getRefOnSuccess :: called
..........
2023-03-14 12:52:05,403 53612 [Engine ] INF: ZAdmin, current state:active, exit
code:success
2023-03-14 12:52:05,403 53612 [Engine ] INF: ZAdmin, current state:final, exit
code:success: state changed to final
```

By default, the `ztp.ini` file is located in the `/pkg/etc/` location. To modify the ZTP configurable options, make a copy of the file in the `/disk0:/ztp/` directory and then edit the `ztp.ini` file.

To reset to the default options, delete the ztp.ini file in the `/disk0:/ztp/` directory.

Note Do not edit or delete the `ztp.ini` file in the `/pkg/etc/` location to avoid issues during installation.

The following example shows the sample of the `ztp.ini` file:

```
[Startup]
start: True
retry_forever: True

[Fetcher Priority]
USB: 0

Mgmt4:  1
Mgmt6:  2
DPort4: 3
DPort6: 4
```

Enable ZTP Using CLI

If you want to enable ZTP using CLI, use the **ztp enable** command.

Configuration example

```
RP/0/RP0/CPU0:ios#ztp enable
Fri Jul 12 16:09:02.154 UTC
Enable ZTP? [confirm] [y/n] :y
ZTP Enabled.
```

Disable ZTP Using CLI

If you want to disable ZTP using CLI, use the **ztp disable** command.

Configuration example

```
RP/0/RP0/CPU0:ios#ztp disable
Fri Jul 12 16:07:18.491 UTC
Disable ZTP? [confirm] [y/n] :y
ZTP Disabled.
Run ZTP enable to run ZTP again.
```

Provision ZTP

When you boot the device, the ZTP process initiates automatically if the device does not have a prior configuration. During the process, the NCS 1010 receives the details of the configuration file from the DHCP server. The ZTP process initiates when you boot the network-device with an IOS-XR image. The process starts only on the device that doesn't have a prior configuration. Here is the high-level work flow of the ZTP process for the Fresh boot:

1. ZTP sends DHCP request to fetch the ZTP configuration file or user script. To help the Bootstrap server uniquely identify the device, ZTP sends below DHCP option
   - DHCP(v4/v6) client-id=Serial Number
   - DHCPv4 option 124: Vendor, Platform, Serial-Number
   - DHCPv6 option 16: Vendor, Platform, Serial-Number

The following is the default sequential flow of the ZTP process:

- ZTP sends IPv4 DHCP request first on all the management port. In case there is a failure, then ZTP sends IPv6 DHCP request on all the management port.
- ZTP sends IPv4 DHCP request first on all the data port. In case there is a failure, then ZTP sends IPv6 DHCP request on all the data port.

  The default sequential flow is defined in configuration file and you can modify the sequence using the configuration file.

2. DHCP server identifies the device and responds with DHCP response using one of the following options: DHCP server should be configured to respond with the DHCP options.
   - DHCPv4 using BOOTP filename to supply script/config location
   - DHCPv4 using Option 67 (bootfile-name) to supply script/config location
   - DHCPv6 using Option 59 (OPT_BOOTFILE_URL) to supply script/config location
3. The network device downloads the file from the web server using the URI location that is provided in the DHCP response.
4. The device receives a configuration file or script file from the HTTP server.

Note

- If the downloaded file content starts with !! IOS XR it is considered as a configuration file.
- If the downloaded file content starts with #! /bin/bash, #! /bin/sh or #!/usr/bin/python it is considered as a script file.

5. The device applies the configuration file or executes the script or binary in the default bash shell.
6. The Network device is now up and running.

ZTP Logging

ZTP logs its operation on the flash file system in the directory /disk0:/ztp/. ZTP logs all the transaction with the DHCP server and all the state transition.

The following example displays the execution of a simple configuration script downloaded from a management interface or a data interface using the command ztp initiate network interface Ten 0/0/0/0 verbose. This script unshuts all the interfaces of the system and configure a load interval of 30 seconds on all of them.

```
2022-06-17 11:52:34,682  19292 [Xr           ] INF: Downloading the file to /tmp/ztp.script
2022-06-17 11:52:35,329  19292 [Report       ] INF: User script downloaded successfully.
Provisioning in progress.
2022-06-17 11:52:35,330  19292 [Engine      ] DEB: ZAdmin, current state:active. Processing
 work: Config device work for ZAdmin. done = False
2022-06-17 11:52:35,330  19292 [ZAdmin       ] DEB: Proceeding to provision the NCS 1010
2022-06-17 11:52:35,331  19292 [Engine      ] DEB: ZAdmin, current state:active. Processing
 work: ZAdmin: Apply configuration. done = False
2022-06-17 11:52:35,331  19292 [Engine      ] INF: ZAdmin, current state:active: state tag
 changed to provision
RP/0/RP0/CPU0:Jun 17 11:52:35.341 UTC: pyztp2[140]: %INFRA-ZTP-4-CONFIG_INITIATED : ZTP has
```

```
 initiated config load and commit operations
2022-06-17 11:52:35,339  19292 [Env          ] DEB: No MTU configs detected
2022-06-17 11:52:35,340  19292 [Engine      ] DEB: ZAdmin, current state:active. Processing
 work: ZAdmin: Apply configuration. done = False
2022-06-17 11:52:35,354  19292 [Xr           ] DEB: Will apply the following config:
/disk0:/ztp/customer/config.candidate
2022-06-17 11:52:35,354  19292 [Xr           ] INF: Applying user configurations
2022-06-17 11:52:35,355  19292 [Configuration] INF: Provisioning via config replace
2022-06-17 11:52:54,656  19292 [Configuration] INF: Configuration has been applied
2022-06-17 11:52:54,656  19292 [Engine      ] DEB: ZAdmin, current state:active. Processing
 work: Sending standby sync message. done = False
2022-06-17 11:52:54,663  19292 [Engine      ] DEB: ZAdmin, current state:active. Processing
 work: [privileged] getting engine status. done = False
2022-06-17 11:52:54,664  19292 [Engine      ] DEB: ZAdmin, current state:active. Processing
 work: ZAdmin: Execute post-configuration script. done = False
2022-06-17 11:52:55,212  19292 [Env         ] INF: Env::cleanup, success:True, exiting:False
2022-06-17 11:52:55,213  19292 [ZtpHelpers   ] DEB: Executing: source /pkg/bin/ztp_helper.sh
 && echo -ne  | xrcmd "show running-config"
2022-06-17 11:52:55,825  19292 [Env          ] INF: Executing command ip netns exec
vrf-default /sbin/dhclient -4 -cf /etc/dhcp/dhclient.conf.ztp -lf
/var/lib/dhcp/dhclient.leases.ztp -sf /etc/dhcp/dhclient-script.ztp2 -r Mg0_RP0_CPU0_0 to
release IP
2022-06-17 11:52:56,968  19292 [Xr          ] INF: Removing linux route with ip 203.0.113.254
2022-06-17 11:52:57,023  19292 [Engine       ] INF: ZAdmin, current state:active, exit
code:success
2022-06-17 11:52:57,023  19292 [Engine       ] INF: ZAdmin, current state:final, exit
code:success: state changed to final
2022-06-17 11:52:59,737  19292 [Engine       ] DEB: ZAdmin, current state:final, exit
code:success. Processing work: Sending standby sync message. done = False
2022-06-17 11:52:59,738  19292 [Engine       ] WAR: ZAdmin, current state:final, exit
code:success: work is ignored: work=<desc='Sending standby sync message' done=False
priv=False>
2022-06-17 11:52:59,738  19292 [Engine       ] DEB: ZAdmin, current state:final, exit
code:success. Processing work: [privileged] getting engine status. done = False
2022-06-17 11:53:04,744  19292 [main     ] DEB: Moved to final state
2022-06-17 11:53:04,745  19292 [main     ] DEB: ZTP completed successfully
2022-06-17 11:53:04,745  19292 [main     ] INF: Exiting SUCCESSFULLY
2022-06-17 11:53:04,746  19292 [main     ] DEB: Exiting. Will not retry now.
2022-06-17 11:53:04,746  19292 [main    ] DEB: Shutting down adaptor. Cleanup False. Exiting
 False
2022-06-17 11:53:04,748  19292 [Engine       ] DEB: ZAdmin, current state:final, exit
code:success. Processing work: [privileged] prepare engine shutdown. done = False
2022-06-17 11:53:04,849  19292 [Engine       ] DEB: ZAdmin, current state:final, exit
code:success. Processing work: [privileged] shutting down ZAdmin engine. done = False
2022-06-17 11:53:04,849  19292 [Engine       ] INF: ZAdmin, current state:final, exit
code:shutdown
2022-06-17 11:53:04,849  19292 [Engine      ] INF: ZAdmin, exit code:shutdown: state changed
 to None
2022-06-17 11:53:04,849  19292 [Engine       ] DEB: ZAdmin, exit code:shutdown: breaking
engine loop after shutdown
2022-06-17 11:53:04,850  19292 [Engine      ] DEB: ZAdmin, exit code:shutdown: end of event
 loop
2022-06-17 11:53:04,850  19292 [Adaptor      ] DEB: Adaptor : Cleanup for admin context on
 Terminate
2022-06-17 11:53:06,119  19292 [main     ] INF: Exiting SUCCESSFULLY
2022-06-17 11:53:06,119  19292 [main     ] INF: ZTP Exited
RP/0/RP0/CPU0:Jun 17 11:53:06.119 UTC: pyztp2[140]: %INFRA-ZTP-4-EXITED : ZTP exited
```

Generate Tech Support Information for ZTP

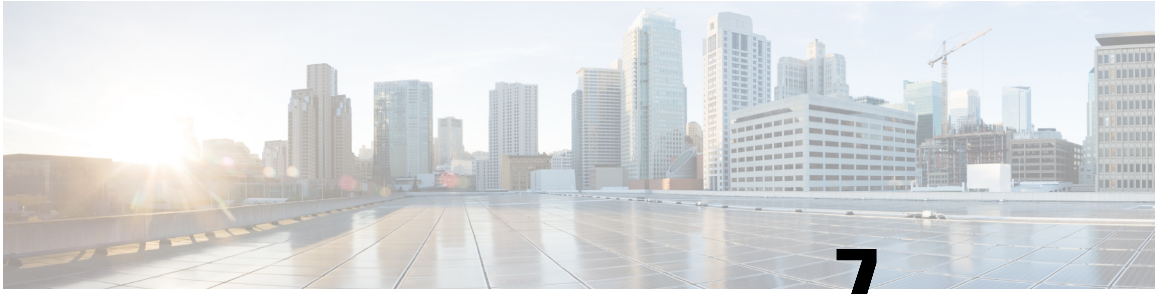
When you have a problem in the ztp process that you cannot resolve, the resource of last resort is your Cisco Systems technical support representative. To analyze a problem, your technical support representative needs certain information about the situation and the symptoms that you are experiencing. To speed up the problem isolation and resolution process, collect the necessary data before you contact your representative.

Use the **show tech-support ztp** command to collect all debugging information of ztp process.

Example:

```
RRP/0/RP0/CPU0:ios#show tech-support ztp
Thu Jul 28 08:33:27.531 UTC
++ Show tech start time: 2022-Jul-28.083327.UTC ++
Thu Jul 28 08:33:28 UTC 2022 Waiting for gathering to complete
..
Thu Jul 28 08:33:34 UTC 2022 Compressing show tech output
Show tech output available at 0/RP0/CPU0 :
/harddisk:/showtech/showtech-R1-ZTP-2022-Jul-28.083327.UTC.tgz
++ Show tech end time: 2022-Jul-28.083334.UTC ++
RP/0/RP0/CPU0:R1#
```

In the above example, the tech support information is saved as .tgz file in the specified location. This information can be shared with the Cisco Technical Support representatives for troubleshooting the ztp process.



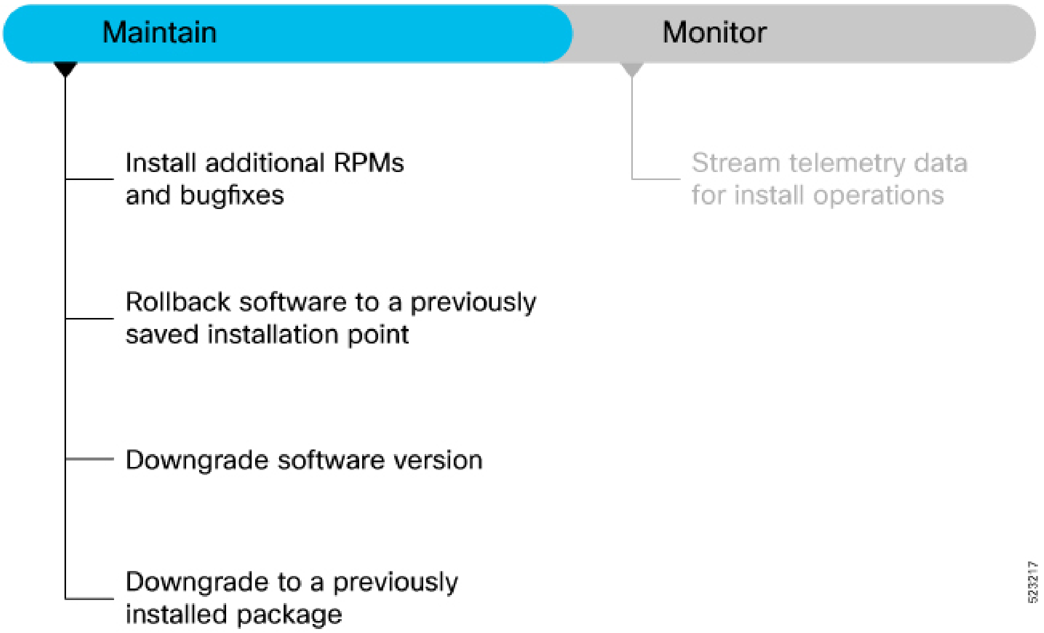
CHAPTER 7

Manage the NCS 1010

Use the procedures in this section to maintain the NCS 1010 at optimum conditions and monitor the install operation by streaming telemetry data.

The following workflow shows the tasks involved in managing the software:

Figure 13: Workflow to Manintain and Monitor the Software Installation


Maintain
Monitor
Install additional RPMs and bugfixes
Rollback software to a previously saved installation point
Downgrade software version
Downgrade to a previously installed package
Stream telemetry data for install operations
523217

This section contains the following topics:

- Install Additional RPMs and Bug Fixes, on page 97
- Downgrade Software Version, on page 100
- Downgrade to a Previously Installed Package, on page 102
- Stream Telemetry Data for Install Operations, on page 104

Install Additional RPMs and Bug Fixes

You can install individual optional packages when new features are added or software problems are fixed.

Before you begin

When you upgrade the Cisco IOS XR software, you can also install or remove optional feature packages (RPMs or bug fixes) *before* applying the changes in the NCS 1010. You can perform this operation while an atomic change is already in progress. However, all packaging operations before this command are discarded.

You can install the packages from a remote repository or copy the files to the NCS 1010. If you are using a remote repository, ensure you have created and configured an external repository to store the packages. See the topic.

Download the specific additional RPMs and latest bug fix RPMs as tarballs to the repository. If the bug fix has dependencies, we recommend that you create a bug fix tarball that contains all dependencies.The *README* file in the tarball provides relevant information about the bug fix and identifies any dependencies – for example, whether other bug fix RPMs may be required for a complete fix.

Option 1: Install RPMs Using Command Line Interface

Optional RPMs and bug fixes are available as TAR files on the Software Download page. Starting with Cisco IOS XR Release 24.3.1, you are no longer required to manually extract the RPMs from the TAR file; you can install the bug fix RPM directly from the TAR file.

Procedure

Step 1 Check the available packages in the repository.

Example:

```
RP/0/RP0/CPU0:ios#show install available

Trying to access repositories...
Package         Architecture        Version            Repository
xr-8000-core     x86_64             7.8.1              remote-repo
xr-core          x86_64             7.8.1              remote-repo
```

Step 2 Install the packages (additional RPMs or bug fixes).

- **Option 1:** Install RPMs without control over reload operation.

  Important
  This option is not applicable when you downgrade or remove RPMs.

  You can either specify a tarfile (with bug fixes or optional packages), or a repository containing the RPMs. Use this command:

  ```
  RP/0/RP0/CPU0:ios#install source full-path-to-rpm [all]
  ```

  Specify the **all** keyword if you want to install optional packages. Exclude the **all** keyword if you want to upgrade the packages that are currently installed on the system.

  The *full-path-to-rpm* can be one of the following locations based on where you have saved the files.

  - Local path—files located in or under `/var/xr/disk1/`, `/harddisk:/` or `/misc/disk1/`
  - Remote repository or tar file—`ftp://<server>[;<named-vrf>]/<remote_path>`, `https://<server>[;<named-vrf>]/<remote_path>` or `http://<server>[;<named-vrf>]/<remote_path>`

    If you want to add new packages from this source, you must use the **all** keyword:

```
RP/0/RP0/CPU0:ios#install source full-path-to-rpm all sync
```

Note

If the remote repository is reachable through a named VRF, you must mention the named VRF in the above commands. For example,

```
RP/0/RP0/CPU0:ios#install source http://203.0.113.1;vrf1/repoinfra/install_rpms.tar
```

where **vrf1** is the named VRF through which the remote repository is accessible.

The operation adds the RPMs and applies the change via `reload` or `restart` operation, whichever is least impactful based on the update.

- **Option 2:** Install RPMs with control over reload operation.

  Important

  This option is applicable when you downgrade, remove or rollback RPMs.

  a. Install RPMs by providing the RPM name, Cisco bug fix ID (example, CSCab12345) or add packages from a specified source. Use the **install package add** command if you want to add new optional packages, else use the **install package upgrade** command.

  ```
  RP/0/RP0/CPU0:ios#install package add <pkg1> <pkg2> <pkgn>
  ```

  Or

  ```
  RP/0/RP0/CPU0:ios#install package upgrade <pkg1> <pkg2> <pkgn>
  ```

  b. Apply the changes.

  ```
  RP/0/RP0/CPU0:ios#install apply [reload | restart]
  ```

  You can use the `reload` or `restart` options based on the change that is installed. To determine whether a `reload` or `restart` is required, check the output of **show install request** or **show install history last transaction verbose** command. The output indicates the required actions.

  ```
  RP/0/RP0/CPU0:ios#show install history last transaction verbose
  2023-01-25 05:45:37 UTC    Transaction 87 started
  2023-01-25 05:45:37 UTC      Atomic change 87.1 started
  2023-01-25 05:45:37 UTC      Packaging operation 87.1.1 started
  2023-01-25 05:45:37 UTC    Transaction 87 complete

  Least impactful apply method: process restart
  ```

Step 3 Check the status of the install operation.

Example:

```
RP/0/RP0/CPU0:ios#show install request
User request: No user requests found
State:        Success
Current activity:    No install operation in progress

The following actions are available:

    install package add
    install package remove
    install package upgrade
    install package downgrade
    install package replace
    install package rollback
    install replace
    install rollback
```

```
    install source
```

Note

Include the keyword `noprompt` in the commands to enable the system to bypass your permission to reload the NCS 1010.

Step 4 Verify the image and packages are activated successfully.

Example:

```
RP/0/RP0/CPU0:ios# show install request
User request: install package add xr-mcast
Operation ID: 87.1.1
State: Success
```

Step 5 Commit the transaction.

Example:

```
RP/0/RP0/CPU0:ios#install commit
```

Option 2: Install RPMs Using YANG Data Model

Use `Cisco-IOS-XR-install-augmented-act.yang` data model to install the RPMs or bug fixes.

Procedure

Use the `install-package-replace` RPC on the data model.

Example:

```
<install-package-replace>
  <source-type>remote</source-type>
  <source>remote-repo</source>
  <file>rpm-file-name</file>
</install-package-replace>
```

If the install operation lists the repository reachable through a VRF, you must add the VRF name for the operation to be successful.

```
<install-package-upgrade xmlns=http://cisco.com/ns/yang/Cisco-IOS-XR-install-augmented-act>
  <source-type>ftp</source-type>
  <source>203.0.113.1;vrf1/repoinfra/install_rpms.tar</source>
</install-package-upgrade>
```

Downgrade Software Version

Downgrade the current software version to a previous software release in case of an upgrade failure or based on requirement.

Note When downgrading the software image from release 24.4.x to an earlier version, we recommend to manually downgrade the line card firmware as well to prevent any impact on various functionalities.

Before you begin

Check the FPD status and ensure that all the FPDs are in `CURRENT` state.

```
RP/0/RP0/CPU0:ios#show hw-module location all fpd
```

If the FPDs are not in `CURRENT` state, upgrade the FPDs.

```
RP/0/RP0/CPU0:ios#upgrade hw-module location all fpd all
```

After all the FPDs are upgraded, reload the NCS 1010.

```
RP/0/RP0/CPU0:ios#reload location all
Proceed with reload? [confirm]
```

After the NCS 1010 reloads, check that all the FPDs are in `CURRENT` state.

Procedure

Step 1 Determine the supported target versions to downgrade from the current version.

Example:

```
RP/0/RP0/CPU0:ios#show install upgrade-matrix
```

View the hardware or software limitations, and bridging SMUs required for the version downgrade. For more information about checking compatibility between the current and target versions, see .

Downgrading Packages:

Customers can also downgrade user-specified packages (for example, xr-telnet). This is separate from downgrading the entire XR version, but an ISO for an earlier version of XR is used instead of a newer ISO.

Step 2 Back up the file system of the current version for recovery purposes.

Example:

Copy the running configuration to the harddisk: directory on the NCS 1010:

```
RP/0/RP0/CPU0:ios#copy running-config harddisk:/running_config-<mmddyyyy>
```

Copy the running configuration to a remote server:

```
RP/0/RP0/CPU0:ios#scp harddisk:/ running_config user@<ip-address>:<location>
```

Step 3 Download the target version from the Software Download Center.

Step 4 You can either install from the remote repository or copy the ISO image file to the /harddisk: of the NCS 1010.

Example:

```
RP/0/RP0/CPU0:ios#scp root@<ip-address>:/<dir>/1010-x64-release.iso harddisk:
```

Step 5 Verify that the MD5 checksum of the copied target file matches with the MD5 value of the source on the Software Download Center.

Example:

```
RP/0/RP0/CPU0:ios#show md5 file /harddisk:/1010-x64-<target-version>.iso
```

Step 6 Install the base image to downgrade the system.

- **Option 1:** Install ISO without control over reload timing.

  ```
  RP/0/RP0/CPU0:ios#install replace /harddisk:/1010-x64-release.iso
  ```

  The image is installed, the changes are applied through a reload or a restart of the system, and commits the changes. However, you do not have control over the timing of the reload or restart —these occur as soon as the package operation completes and the system is ready.

  If you want to control when your system reloads (management of a network outage), we recommend that you schedule a downgrade window and perform an **install replace** operation, letting the system reload without intervention.

- **Option 2:** Install ISO with control over reload timing.

  a. Install the image.

  ```
  RP/0/RP0/CPU0:ios#install package replace /harddisk:/1010-x64-release.iso
  ```

  b. Apply the changes.

  ```
  RP/0/RP0/CPU0:ios#install apply [reload | restart]
  ```

  You can use either the `reload` or `restart` options based on the file that is installed. To determine whether a `reload` or `restart` is required, check the output of **show install request** command. The output indicates the required actions.

Step 7 After the base image is downgraded, install the additional packages. For more information, see Install Additional RPMs and Bug Fixes, on page 97.

During an install operation, if the system reboots unexpectedly or an apply by reload results in the system failing to boot, it automatically recovers to its software state before the current transaction.

Downgrade to a Previously Installed Package

You can downgrade a package to a previously installed version. By default, the subsequent previous version (version previous to the current version) is installed. Also, you can downgrade the software to a specific version of interest. To remove a bug fix RPM from the installed packages, downgrade the package to a version where the fix was not applied.

Note While downgrading, you can choose any previous version, including the base version of the RPM. However, when downgrading a bug fix RPMs, ensure that you also consider all dependencies of the current version.

Bug fix RPM is an upgrade to the existing package. The action of removing a bug fix RPM either removes the entire feature, or fails if the package is mandatory.

You can use the **show install fixes deactivate** command to view information related to removing a bug fix. This command provides information such as the package changes, other bug fixes that get deactivate, instructions for adding packages missing for the bug fix removal to be successful, command for removing the bug fix, and any recommendations, if applicable. See the following example:

Note You can specify any number of DDTS seperated by a space in the **show install fixes deactivate** command. For example, to know the recommendations for removing bug fix for ABC123, DEF456, and GHI789, you can use **show install fixes deactivate ABC123 DEF456 GHI789** command.

The following example shows the package `xr-telnet-24.3.1v1.0.1` is downgraded to `xr-telnet-24.3.1v1.0.0`. The path to source can be a local location or a configured repository.

Before you begin

Ensure you have access to the previously installed package and its source.

Procedure

Step 1 Downgrade the package using one of the following options:

- Downgrade the package where the fix was applied. When multiple older versions of the package are present in the configured repositories, the immediate previous version of the package is installed. Use caution when using this command as the current version of the package is removed completely.

  ```
  RP/0/RP0/CPU0:ios#install package downgrade xr-telnet
  ```

  Apply the changes.

  ```
  RP/0/RP0/CPU0:ios#install apply [reload | restart]
  ```

  Attention
  To identify whether to reload the NCS 1010 or restart the affected processes as part of the apply operation, use either **show install history last transaction verbose** command or **show install request** command.

- Install a specific earlier version of the optional package. The changes are applied automatically.

  Attention
  An automatic change may trigger a reload of the NCS 1010 depending on the package being downgraded.

  ```
  RP/0/RP0/CPU0:ios#install source <path-to-source> xr-telnet-24.3.1v1.0.0
  ```

- Use `install` RPC on the `Cisco-IOS-XR-install-act.yang` data model. Here is an example usage with a local repository:

  ```
  <install>
   <packages>
     <packagename>xr-telnet-24.3.1v1.0.0
      xr-telnet-24.0.11v1.0.0
  xr-telnet-24.4.1v1.0.0
  </packagename>
   </packages>
     <source>file://<path-to-source>/</source>
  </install>
  ```

  The package version `xr-telnet-24.3.1v1.0.1xr-telnet-24.0.11v1.0.1xr-telnet-24.3.1v1.0.1` is downgraded to `xr-telnet-24.3.1v1.0.0xr-telnet-24.0.11v1.0.0xr-telnet-24.4.1v1.0.0`.

Step 2 Commit the operation.

Example:

```
RP/0/RP0/CPU0:ios#install commit
```

Stream Telemetry Data for Install Operations

To stream telemetry data that is related to software installation, you must create subscriptions to the sensor paths in the YANG data models. See *Obtain Data Models for Install Operation* for the list of supported data models. For information about establishing a telemetry session and creating subscriptions, see the .

Stream Telemetry Data About	Description	YANG Path

Summary of active packages	Data is streamed after a successful **apply** operation. An active package is the software currently running on the system.	`Cisco-IOS-XR-install-oper:`
		`install/packages/active/summary`

Summary of committed packages	Data is streamed after a successful **commit** operation. A package that is committed remains active following a system reload.	`Cisco-IOS-XR-install-oper:`
		`install/packages/committed/summary`

Status of the last request operation	Data is streamed when starting a new request and also when entering an `idle` state. If the operation has failed, this includes error messages along with recovery state.	`Cisco-IOS-XR-install-oper:`
		`install/request`

Image version and GISO label	Data is streamed after a successful **apply** operation.	`Cisco-IOS-XR-install-oper:`
		`install/version`

Packaging information	Data is streamed at the start and end of a packaging operation.	`Cisco-IOS-XR-install-augmented-oper:`
		`install/history/latest-packaging-operation`

Atomic information	Data is streamed at the start and end of **apply** operation.	`Cisco-IOS-XR-install-augmented-oper:`
		`install/history/latest-atomic-change`

Transaction information	Data is streamed at the start, in progress, and end of a **commit** operation.
	Note
	After a transactional rollback, some of the data such as summary of active packages, image version can change. However, telemetry events are not sent after the reload operation.
		`Cisco-IOS-XR-install-augmented-oper:`
		`install/history/latest-transaction`

CHAPTER 8

Troubleshoot NCS 1010 Setup and Upgrade

Use the procedures in this section to troubleshoot NCS 1010 bring-up, software upgrade or downgrade by understanding the problem, probable cause, and the solution.

The following image shows the tasks involved in finding solutions to NCS 1010 setup and upgrade issues:

This section contains the following topics:

Recover NCS 1010 From Boot Failure

If the command line interface is not accessible, you can recover the NCS 1010 from a boot failure using one of these recovery methods.

Boot the NCS 1010 Using USB Drive

Problem:

After installing the hardware, you boot the NCS 1010 after connecting to the console port and powering ON the NCS 1010. The NCS 1010 initiates the boot process using the pre-installed operating system (OS) image. But the NCS 1010 fails to boot, times out or stops responding after the boot process initializes.

Cause:

The NCS 1010 does not boot if an install image is not present on the NCS 1010 or the image is corrupt.

Solution:

Boot the NCS 1010 using a bootable USB flash drive.

The bootable USB flash drive is used to reimage the NCS 1010 during system upgrade or boot the NCS 1010 in case of boot failure. During the USB boot process, the NCS 1010 is re-imaged with the version available on the USB flash drive.

To boot the NCS 1010 using a USB flash drive, you need the following devices:

- A local machine (Windows, Linux, or MAC) with USB Type-A.

- USB flash drive with a storage capacity that is between 8GB (min) and 32 GB (max). USB 2.0 and USB 3.0 are supported.

Note USB Type-C is not supported.

Procedure

Step 1 Create a bootable USB flash drive from your local machine (Windows or MAC):

a) Connect the USB flash drive to your local machine and format it with File Allocation Table (FAT) 32 file system using the Windows Operating System or Apple MAC Disk Utility. Formatting the USB drive to FAT creates addressable sectors that ensures that each piece of information in the file can be found by the computer.

After formatting the USB flash drive, right-click on the USB disk and view the properties.

b) On the Software Download page, navigate to the required Cisco IOS XR product and release. The USB boot image is available in the format `<platform>-usb-<version>.zip` compressed file. For example, the USB boot image for Cisco NCS 1010s for release 24.3.1 is `1010-x64-usb-24.3.1.zipASR9K-x64-usb-24.3.1.zip` file.

c) Download the compressed USB boot image from the Software Download page to your host computer.

d) Verify that the copy operation is successful. To verify, compare the file size on the Software Download page and the copied file on your computer. You can also verify the MD5 checksum value. This value ensures that the copied file is valid and untampered.

e) Unzip the file to extract the content of the compressed boot file inside the USB flash drive. This converts the USB flash drive to a bootable drive.

Note

The content of the zipped file (`EFI` and `boot` directories) should be extracted directly into the root of the USB flash drive. If the unzipping application places the extracted files in a new folder, move the `EFI` and `boot` directories to the root folder of the USB flash drive.

f) Remove the USB flash drive from your computer.

The USB flash drive is ready to be used as a bootable disk to install and boot the Cisco IOS XR image.

Step 2 Boot the NCS 1010 using the bootable USB flash drive.

a) Use this procedure only on active RP; the standby RP must either be powered OFF or removed from the chassis. After the active RP is installed with images from USB, insert or power ON the standby RP as appropriate.

b) Connect to the console.

c) Insert the USB flash drive in the USB Port Type-A on the NCS 1010.

Ensure that the NCS 1010 is powered ON. When the USB bootable drive is plugged into an operational NCS 1010, the device is detected as disk2:. Verify using **show media location all** command.

```
RP/0/RP0/CPU0:ios#show media location all
Fri Jan 27 08:29:00.808 UTC

Media Info for Location: node0_RP0_CPU0
Partition          Size           Used          Percent        Avail
--------------------------------------------------------------------
rootfs:            54.4G          16.5G          30%           38G
data:              77.3G          20.5G          27%           56.8G
disk0:             3.9G           12M            1%            3.6G
```

```
/var/lib/docker   6.6G          17M           1%            6.2G
disk2:            15G           6.1G          42%           8.6G
log:              5.3G          572M          12%           4.4G
harddisk:         61G           19G           32%           39G
```

d) View the contents of the USB drive.

Example:

```
RP/0/RP0/CPU0:ios#dir disk2:
```

e) Initiate the reimage from the USB bootable drive.

Example:

```
RP/0/RP0/CPU0:ios#reload bootmedia usb noprompt
```

```
RP/0/RP0/CPU0:ios#hw-module location all bootmedia usb
```

Note

If the NCS 1010 was powered OFF, power ON the NCS 1010. Press the `Esc` key continuously to pause the boot process and get the RP to the BIOS menu. Use the arrow key and navigate to the `USB Flash Memory` option in the **Boot Manager** menu, and press the `Enter` key. The BIOS GRUB automatically detects the image from the USB flash drive, starts the installation, and displays the progress of the installation operation.

The NCS 1010 reboots after the reimage with new version available in the USB drive. After the installation is complete, the NCS 1010 reboots and enters the prompt to configure the root username and password.

Boot the NCS 1010 Using iPXE

Problem:

You connect to the console port and power ON the NCS 1010. The NCS 1010 initiates the boot process using the pre-installed operating system (OS) image. But the NCS 1010 fails to boot, times out or stops responding after the boot process initializes.

Cause:

The NCS 1010 does not boot if an install image is not present on the NCS 1010 or the image is corrupt.

Solution:

Boot the NCS 1010 using the image from an iPXE server.

iPXE is a pre-boot execution environment that is included in the network card of the management interfaces. It works at the system firmware (UEFI) level of the NCS 1010. iPXE enables network boot for a NCS 1010 that is offline. The bootloader downloads and installs the ISO image located on an HTTP, FTP, or TFTP server. iPXE boot re-images the NCS 1010. iPXE acts as a boot loader and provides the flexibility to choose the image that the system will boot based on the Platform Identifier (PID), the serial number, or the management MAC address. iPXE must be defined in the DHCP server configuration file.

Procedure

Step 1 Configure the DHCP server for IPv4, IPv6, or both communication protocols before you use the iPXE boot.

a) Create dhcpd.conf file in /etc/ or /etc/dhcp directory. This configuration file stores the network information such as the path to the script, location of the ISO install file, location of the provisioning configuration file, serial number, MAC address of the NCS 1010. The following example shows a sample dhcpd.conf file.

Example:

```
allow bootp;
allow booting;
ddns-update-style interim;
option domain-name "cisco.com";
option time-offset -8;
ignore client-updates;
default-lease-time 21600;
max-lease-time 43200;
option domain-name-servers <ip-address-server1>, <ip-address-server2>;
log-facility local0;
 :
subnet <subnet> netmask <netmask> {
  option ncs1010 <ip-address>;
  option subnet-mask <subnet-mask>;
  next-server <server-addr>;
}
  :
host <hostname> {
  hardware ethernet e4:c7:22:be:10:ba;
  fixed-address <address>;
  filename "http://<address>/<path>/<image.bin>";
```

b) Test the server once the DHCP server is running. For example, for IPv4 protocol:

- Use the MAC address of the NCS 1010:

  Note
  Using the host statement provides a fixed address that is used for DNS, however, verify that option 77 is set to iPXE in the request. This option is used to provide the boot file to the system when required.

  ```
  host <platform>
  {
  hardware ethernet <ncs1010-mac-address>;
  if exists user-class and option user-class = "iPXE" {
   filename = "http://<httpserver-address>/<path-to-image>/<image>";
  }
  ```

  Ensure that the above configuration is successful.

- Use the serial number of the NCS 1010:

  ```
  host <platform>
  {
  option dhcp-client-identifier "<ncs1010-serial-number>";
    filename "http://<IP-address>/<path-to-image>/<image>";
    fixed-address <IP-address>;
  }
  ```

  The serial number of the NCS 1010 is derived from the BIOS and is used as an identifier.

Step 2 Recover the NCS 1010 using iPXE boot.

a) Connect to the console.
b) Power ON the NCS 1010.
c) Press Esc key continuously to pause the boot process and get the RP to the BIOS menu.
d) Use the arrow key and navigate to the Built-in EFI iPXE option in the **Boot Manager** menu, and press the Enter key.

Example:

```
iPXE> ifstat
net0: 00:a0:c9:00:00:00 using i350-b on PCI01:00.0 (closed)
  [Link:up, TX:0 TXE:0 RX:0 RXE:0]
net1: 00:a0:c9:00:00:01 using i350-b on PCI01:00.1 (closed)
  [Link:up, TX:0 TXE:0 RX:0 RXE:0]
net2: 00:a0:c9:00:00:02 using i350-b on PCI01:00.2 (closed)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Down (http://ipxe.org/38086193)]
net3: 00:a0:c9:00:00:03 using i350-b on PCI01:00.3 (closed)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Down (http://ipxe.org/38086193)]
net4: 00:00:00:00:00:04 using dh8900cc on PCI02:00.1 (closed)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Down (http://ipxe.org/38086193)]
net5: 00:00:00:00:00:05 using dh8900cc on PCI02:00.2 (closed)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Down (http://ipxe.org/38086193)]
net6: 04:62:73:08:57:86 using dh8900cc on PCI02:00.3 (closed)
  [Link:up, TX:0 TXE:0 RX:0 RXE:0]

iPXE> set net6/ip 192.0.2.255
iPXE> set net6/netmask 255.255.255.0
iPXE> set net6/gateway 10.48.42.1
iPXE>
iPXE> ifopen net6

iPXE> ping 10.48.42.1
64 bytes from 10.48.42.1: seq=1
64 bytes from 10.48.42.1: seq=2
Finished: Operation canceled (http://ipxe.org/0b072095)
```

e) Boot the image using one of the following options:

- Option 1: Boot with ISO image. After the reimage is successful, add optional RPMs, bug fixes and update running configuration file.
- Option 2: [Preferred option] Boot with Golden ISO (GISO) image that contains the ISO image, optional RPMs, bug fixes and configuration file. Booting with GISO saves time by eleminating the need to update the files individually.

You must keep the standby RP in the BIOS while installing the image on the active RP.

```
BIOS Ver: 09.19 Date: xx/xx/xxxx 17:02:33

Press <DEL> or <ESC> to enter boot manager.                          iPXE initialising
devices...ok

iPXE 1.0.0+ (5fbe7) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP TFTP VLAN EFI ISO9660 NBI Menu
BootMode : 1
Trying net0...
net0: 00:00:01:1c:00:00 using i350-b on PCI01:00.0 (open)
  [Link:up, TX:0 TXE:0 RX:0 RXE:0]
Configuring (net0 00:00:01:1c:00:00).................. ok
net0: 203.0.113.1/255.255.255.0
net0: fe80::2a0:c9ff:fe00:0/64
net1: fe80::2a0:c9ff:fe00:1/64 (inaccessible)
net2: fe80::2a0:c9ff:fe00:2/64 (inaccessible)
net3: fe80::2a0:c9ff:fe00:3/64 (inaccessible)
net4: fe80::200:ff:fe00:4/64 (inaccessible)
net5: fe80::200:ff:fe00:5/64 (inaccessible)
```

```
net6: fe80::662:73ff:fe08:1dba/64 (inaccessible)
Next server: 203.0.113.17
Filename: http://203.0.113.15/system_image.iso
http://203.0.113.15/<image>... ok
```

The BIOS GRUB automatically detects the image from the iPXE server, starts the installation, and displays the progress of the installation operation. After the installation is complete, the NCS 1010 reboots and enters the prompt to configure the root username and password.

You can also boot the NCS 1010 from the iPXE server by using the **hw-module location all bootmedia network reload** command.

```
RP/0/RP0/CPU0:ios# hw-module location all bootmedia network reload
Wed Dec 23 15:29:57.376 UTC
Reload hardware module ? [no,yes]
```

This command configures the NCS 1010 to perform a network-based boot across all modules in the NCS 1010 before a restart. Upon reload, the NCS 1010 attempts to load the operating system image from the specified iPXE server.

Recover Password

Problem:

Unable to access the NCS 1010 due to incorrect login credentials.

Cause:

A root password is used to login to the NCS 1010. If you forget this root password, you cannot access the NCS 1010.

Solution:

If you lose your admin and root user credentials, the NCS 1010 becomes inaccessible. The system can be recovered using a NCS 1010 reimage using iPXE or USB boot. However, this approach is not scalable.

You can use the **system recovery** feature to recover the lost password.

With this feature, the system is recovered without the need to reimage the NCS 1010. The system is recovered to its initial state with the current running software. The installed software and SMUs are retained after the system is recovered. The process complies with the Cisco Product Security Baseline (PSB) where user data is securely erased before recovering the NCS 1010. The following data that are generated at run-time are erased:

- XR and admin configuration including the password data
- Cryptographic keys on the disk
- Data on encrypted partition
- Generated core files
- SNMP interface index files
- Third-party application (TPA) software and data
- Files created by the user

Use the following procedure to recover the password on NCS 1010.

Note This procedure is applicable only when you have already enabled the password recovery feature on your NCS 1010.

```
RP/0/RP0/CPU0:ios(config)#system recovery
```

Procedure

Step 1 Power ON the NCS 1010, and press the `ESC` on the RP console to enter the BIOS GRUB menu.

This procedure must be executed on each RP individually on a modular system.

Step 2 Boot on the standby RP. Press `ESC` key to enter the GRUB (bootstrap program) menu.

Step 3 On the RP0 card console select the **IOS-XR-recovery** option from the GRUB menu and press **Enter**.

Step 4 Select the **IOS-XR-recovery** option from the GRUB menu and press **Enter** on the card console when the `Initiating IOS-XR System Recovery...` message is displayed on the card console.

Note
Do not wait until the card reaches the `Enter root-system username:` prompt. If you reach this prompt, the card will reload automatically and exit the BIOS GRUB menu. The card will boot up as active post the recovery process.

Step 5 On the RP card, create a new root user and password. Log in to the NCS 1010 using the new root username and password.

The NCS 1010 boots with the default configuration. Proceed with configuring the NCS 1010 or load a configuration from a backup file if you had already taken a backup. It is recommended to backup data and save the configuration on an external server.

Ensure that you see this message in the RP console. If this message is not displayed, then repeat the process from step 1 to step 5 until you see the message:

```
RP/0/RP1/CPU0:June 10 06:13:24.551 CEST: sys_rec[1188]: %SECURITY-SYSTEM_RECOVERY-1-REPORT :
System Recovery at 06:10:19 CEST Fri June 10 2022 was successful

RP/0/RP1/CPU0:June 10 06:15:13.967 CEST: sys_rec[1188]: %SECURITY-SYSTEM_RECOVERY-1-REPORT :
System Recovery
```

The password recovery procedure is complete.

The option to recover the system using console port is disabled on bootup because all the previous configurations are erased. With this configuration disabled, if you select **IOS-XR-recovery** option from GRUB menu to recover the system, the recovery is skipped. Enable the password recovery feature again using the **system recovery** command.

Rectify Insufficient Disk Space When Installing Software

Problem:

The software installation terminates with the error `Error on 0/1/CPU0: Insufficient disk space to install packages.`

Cause:

To install the Cisco IOS XR software, an unused disk space of so-and-so must be available on the NCS 1010. If this space is not available before installing the software, the installation process terminates with the error.

Solution:

Identify the required disk space using the **show install log** or **install add** command.

View the space consumed by the harddisk: location using the **show media location all** command.

```
RP/0/RP0/CPU0:ios#show media location all
Wed Jan 8 08:29:00.808 UTC

Media Info for Location: node0_RP0_CPU0
Partition                            Size     Used  Percent    Avail
--------------------------------------------------------------------
rootfs:                             54.4G    16.5G      30%      38G
data:                               77.3G    20.5G      27%    56.8G
disk0:                               3.9G      12M       1%     3.6G
/var/lib/docker                      6.6G      17M       1%     6.2G
disk2:                                15G     6.1G      42%     8.6G
log:                                 5.3G     572M      12%     4.4G
harddisk:                             61G      19G      32%      39G

Media Info for Location: node0_RP1_CPU0
Partition                            Size     Used  Percent    Avail
--------------------------------------------------------------------
rootfs:                             54.3G    16.5G      30%    37.9G
data:                               77.4G    46.1G      60%    31.4G
disk0:                               3.9G     8.5M       1%     3.6G
/var/lib/docker                      6.6G      19M       1%     6.2G
log:                                 5.3G     492M      10%     4.5G
harddisk:                             61G      44G      78%      14G

Media Info for Location: node0_0_CPU0
Partition                            Size     Used  Percent    Avail
--------------------------------------------------------------------
rootfs:                             54.4G    10.1G      18%    44.4G
data:                               77.3G     1.9G       2%    75.5G
/var/lib/docker                      6.6G      16M       1%     6.2G
disk0:                               3.9G     8.2M       1%     3.6G
harddisk:                             61G     109M       1%      57G
log:                                 5.3G     372M       8%     4.6G

Media Info for Location: node0_6_CPU0
Partition                            Size     Used  Percent    Avail
--------------------------------------------------------------------
rootfs:                             54.4G    10.1G      18%    44.4G
data:                               77.3G     1.9G       2%    75.4G
disk0:                               3.9G     8.3M       1%     3.6G
/var/lib/docker                      6.6G      16M       1%     6.2G
harddisk:                             61G     154M       1%      57G
log:                                 5.3G     374M       8%     4.6G
```

Use the following procedure to free up the disk space to make room for the software installation.

Procedure

Step 1 Remove inactive packages from the system.

Example:

View the inactive packages:

```
RP/0/RP0/CPU0:ios(admin)#show install inactive
6 inactive package(s) found:
    ncs5500-xr-6.6.1
    ncs5500-k9sec-3.1.0.0-r661
    ncs5500-mpls-2.1.0.0-r661
    ncs5500-isis-2.1.0.0-r661
    ncs5500-mcast-2.1.0.0-r661
    ncs5500-mgbl-3.0.0.0-r661
```

Remove the inactive packages:

```
RP/0/RP0/CPU0:ios(admin)#install remove inactive all synchronous
   instdir[198]: %INSTALL-INSTMGR-6-INSTALL_OPERATION_STARTED :
Install operation 8 '(admin) install remove inactive all' started by user 'user_b'
Install operation 8 '(admin) install remove inactive all' started by user 'user_b' at
   09:25:41 UTC Fri June 10
Info:     This operation will remove the following package:
ncs5500-xr-6.6.1
    ncs5500-k9sec-3.1.0.0-r661
    ncs5500-mpls-2.1.0.0-r661
    ncs5500-isis-2.1.0.0-r661
    ncs5500-mcast-2.1.0.0-r661
    ncs5500-mgbl-3.0.0.0-r661
Proceed with removing these packages? [confirm]
The install operation will continue synchronously.
```

Step 2 Remove stale or unnecessary files from the harddisk: location such as cores, debug logs, kdump and showtech data. We recommended that you do not remove files from other partitions because these locations may contain files that are relevant to collecting debug information. Carefully inspect the files to be deleted.

Example:

```
RP/0/RP0/CPU0:ios#rmdir harddisk:
Remove directory filename []?newdir
Delete harddisk:/newdir[confirm]y
```

Use the **delete** command to remove specific directory or files. When a directory contains files such as images, bug fixes or configuration files, you must remove the files before deleting the directory.

```
RP/0/RP0/CPU0:ios#delete harddisk:/file
```

Verify that the unwanted directory is removed from the harddisk.

```
RP/0/RP0/CPU0:ios#dir harddisk:
 Directory of harddisk:
 37146       drwx  4096        Sun Dec 14 15:30:48 2008  malloc_dump
 43030       drwx  4096        Wed Dec 24 11:20:52 2008  tracebacks
 43035       drwx  4096        Thu Jan  8 18:59:18 2009  sau
 51026       drwx  4096        Sat Dec 27 02:52:46 2008  tempA
 51027       drwx  4096        Sat Dec 27 02:04:10 2008  dir.not.del
 -430307552  -rwx  342         Fri Jan 16 10:47:38 2009  running-config
 -430305504  -rwx  39790       Mon Jan 26 23:45:56 2009  cf.dat
39929724928 bytes total (39883235328 bytes free)
```

Recover Frozen Console Prompt

Problem:

The console access is frozen and does not respond. In this state, no output or input characters are displayed on the console.

Cause:

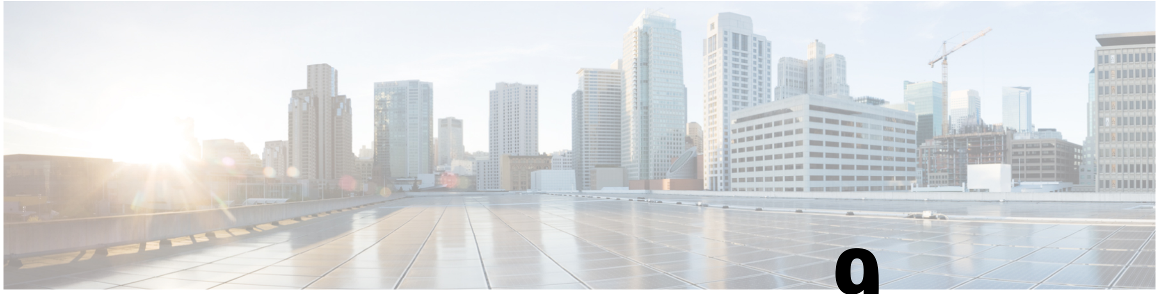
The Priority Flow Control (PFC) functionality is enabed on the console by default. The PFC is also referred to as Class-based Flow Control (CBFC) or Per Priority Pause (PPP) is a mechanism that prevents frame loss due to congestion. Pressing the `Ctrl + S` keys enables the flow control and no output will be seen on the XR console until resumed.

Solution:

Reset the console prompt.

Procedure

Press the `Ctrl + Q` keys to resume the console output.



CHAPTER 9

Disaster Recovery

This chapter describes the disaster recovery process and the health check feature.

Overview

There are two partitions in NCS 1010: RP SSD (CPU partition) and chassis SSD (Disaster Recovery partition). The Disaster Recovery partition contains all the backup configurations such as ISO images, RPMs, and system configuration files. When the node is corrupted, the Disaster Recovery feature allows the CPU to be replaced with the existing configuration. After replacing the CPU, the node reboots and comes up by restoring the software and configuration files from the chassis SSD without traffic loss.

CPU Replacement Considerations

You must consider the following points for CPU replacement.

- When the CPU is removed from the chassis, NCS 1010 chassis runs in headless mode which is non-traffic impacting.
- When the CPU is replaced with another CPU having the same software and RPMs as in the chassis SSD, the configuration is restored from the chassis SSD.
- When the CPU is replaced with another CPU having different software and RPMs as in the chassis SSD, the Disaster recovery process starts. In this case, the node boots with the software from the chassis SSD and the configuration is also restored from the chassis SSD.

Health Check of Backup ISO Image

The Health Check feature ensures error-free booting of NCS 1010 chassis during disaster recovery operations. NCS 1010 has a partition for disaster recovery where the backup ISO image is stored. The backup ISO image is stored in the chassis SSD.

The chassis SSD content is audited against the running software by the install process in the background every 12 hours to detect corruption. If the ISO image is corrupted, the software will recover it by copying from the backup location. If the software fails to synchronize with the chassis SSD, then the **Disaster Recovery ISO Image Corruption** alarm is raised. See the *Troubleshooting Guide for Cisco NCS 1010* to clear the alarm.

Automated File Management System

Table 10: Feature History

Feature Name	Release Information	Feature Description
Automated File Management System	Cisco IOS XR Release 24.4.1	The new Automated File Management System is designed for efficient file handling on each node. This system automatically archives older files and removes them from local nodes to free up valuable SSD space. It manages the following types of files: • System-generated log files • Showtech-related residual files

The automated file management system archives older files to free up valuable SSD space by deleting them from the local nodes.

Types of files

The SSD stores two types of files that are generated by

- **User**: creates and owns files for requirement purposes and deletes the files when are no longer needed.
- **System**: organizes files automatically based on the file content and the application that created the content such as .log and showtech-related residual files.

  Automated file management is applicable for the system-generated files.

How the automated file management system works

These stages describe how the automated file management works for various files.

Log files

The NCS 1014 system uses the log rotation configurations to manage log files as required.

1. The system checks for the *.log* files exceeding 10 MB file size.

Note This threshold is applicable for `/tmp` folder files only. For files in other folders, the system uses a different threshold and follows the same process.

2. After locating the file, the system

   a. archives that file with .gz extension, and

   b. creates a new .log file with the same name.

3. The system then maintains one active log file and two archived files.
4. When new files are archived, the system deletes the oldest archived file to make sure that only the two most recently archived files and the current log file are retained.

showtech-related residual log files

1. The system generates the residual files during the collection of logs when **Showtech logs** is in operation.
2. After collecting the logs, the system automatically removes these residual files to conserve space.

CHAPTER 10

Configuring BGP

BGP Overview

Table 11: Feature History

Feature Name	Release Information	Feature Description
Configuring BGP	Cisco IOS XR Release 7.11.1	BGP routers from a TCP connection between peer routers to exchange network reachability information to open and confirm the connection parameters. BGP on NCS 1010 is enabled only on management port enabling the customer to manage the Dynamic Circuit Network(DCN) connectivity.

BGP uses TCP as its transport protocol. Two BGP routers form a TCP connection between one another (peer routers) and exchange messages to open and confirm the connection parameters.

BGP routers exchange network reachability information. This information is mainly an indication of the full paths (BGP autonomous system numbers) that a route should take to reach the destination network. This information helps construct a graph that shows which autonomous systems are loop free and where routing policies can be applied to enforce restrictions on routing behavior.

Any two routers forming a TCP connection to exchange BGP routing information are called peers or neighbors. BGP peers initially exchange their full BGP routing tables. After this exchange, incremental updates are sent as the routing table changes. BGP keeps a version number of the BGP table, which is the same for all its BGP peers. The version number changes whenever BGP updates the table due to routing information changes. Keepalive packets are sent to ensure that the connection is alive between the BGP peers and notification packets are sent in response to error or special conditions.

Note ASN change for the BGP process is not currently supported via **commit replace**.

Prerequisites for Implementing BGP

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

BGP Router Identifier

For BGP sessions between neighbors to be established, BGP must be assigned a router ID. The router ID is sent to BGP peers in the OPEN message when a BGP session is established.

BGP attempts to obtain a router ID in the following ways (in order of preference):

- By means of the address configured using the **bgp router-id** command in router configuration mode.
- By using the highest IPv4 address on a loopback interface in the system if the router is booted with saved loopback address configuration.
- By using the primary IPv4 address of the first loopback address that gets configured if there are not any in the saved configuration.

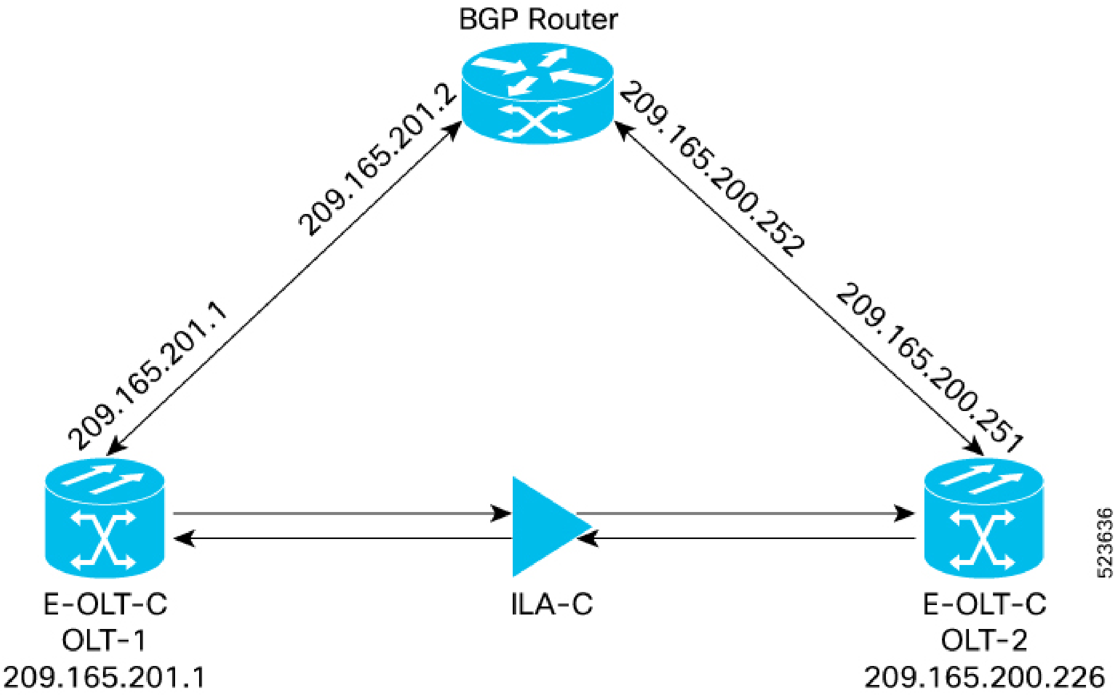
If none of these methods for obtaining a router ID succeeds, BGP does not have a router ID and cannot establish any peering sessions with BGP neighbors. In such an instance, an error message is entered in the system log, and the **show bgp summary** command displays a router ID of 0.0.0.0.

After BGP has obtained a router ID, it continues to use it even if a better router ID becomes available. This usage avoids unnecessary flapping for all BGP sessions. However, if the router ID currently in use becomes invalid (because the interface goes down or its configuration is changed), BGP selects a new router ID (using the rules described) and all established peering sessions are reset.

Note We strongly recommend that the **bgp router-id** command is configured to prevent unnecessary changes to the router ID (and consequent flapping of BGP sessions).

Configuring BGP

Figure 14: BGP Topology



```
config
route-policy pass-all
end-policy
commit

RP/0/RP0/CPU0:OLT-1#conf
Fri Feb 23 09:32:29.216 IST
router bgp 1
bgp router-id 209.165.201.1
 address-family ipv4 unicast
  redistribute connected
  redistribute ospf 1 route-policy pass-all
 !
 address-family vpnv4 unicast
 !
 neighbor 209.165.201.2
  remote-as 100
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!
commit

show running-config router bgp
Thu Jan 11 15:54:43.439 IST
router bgp 1
```

```
 bgp router-id 209.165.201.1
 address-family ipv4 unicast
  redistribute connected
  redistribute ospf 1 route-policy pass-all
 !
 address-family vpnv4 unicast
 !
 neighbor 209.165.201.2
  remote-as 100
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!


RP/0/RP0/CPU0:OLT-2#conf
Fri Feb 23 09:32:29.216 IST
router bgp 1
 bgp router-id 209.165.200.226
 address-family ipv4 unicast
  redistribute connected
  redistribute ospf 1 route-policy pass-all
 !
 address-family vpnv4 unicast
 !
 neighbor 209.165.200.252
  remote-as 100
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!
commit

show running-config router bgp
Thu Jan 11 15:59:12.210 IST
router bgp 1
 bgp router-id 209.165.200.226
 address-family ipv4 unicast
  redistribute connected
  redistribute ospf 1 route-policy pass-all
 !
 address-family vpnv4 unicast
 !
 neighbor 209.165.200.252
  remote-as 100
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!

show route
route  router-id


RP/0/RP0/CPU0:BGP-ROUTER#config
Fri Feb 23 09:32:29.216 IST
router bgp 100
 bgp router-id 209.165.201.19
 address-family ipv4 unicast
  redistribute connected
 !
 neighbor 209.165.201.1
  remote-as 1
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
 neighbor 209.165.200.251
  remote-as 1
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!
commit

show running-config router bgp
Thu Jan 11 15:59:58.059 IST
router bgp 100
 bgp router-id 209.165.201.19
 address-family ipv4 unicast
  redistribute connected
 !
 neighbor 209.165.201.1
  remote-as 1
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
 neighbor 209.165.200.251
  remote-as 1
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!
```

Note Autonomous system numbers 2.0 standard supports 0 to 65535, and autonomous system numbers standard 4.0 supports 65536 onwards.

CHAPTER 11

Configure CDP

Table 12: Feature History

Feature Name	Release Information	Feature Description
CDP Support	Cisco IOS XR Release 7.10.1	Cisco Discovery Protocol (CDP) support is introduced on NCS 1010. CDP is a Layer 2 network discovery protocol for learning about directly connected Cisco devices. This protocol lets you easily view peer Cisco device information such as IP address, version number, platform type, connected ports, and so on for network planning and troubleshooting.

CDP is a Cisco proprietary layer 2 protocol used to obtain information about peer Cisco devices. It exchanges CDP packets with its neighbors to discover the platform type and capabilities of the peer device.

Each device configured for CDP sends periodic messages, known as advertisements, to a multicast address. Each device advertises at least one address at which it can receive messages. The advertisements also contain time-to-live or hold-time information which indicates the length of time a receiving device holds CDP information (180 seconds by default) before discarding it. Each device also listens to the periodic CDP messages sent by others (every 60 seconds by default) to learn about neighboring devices and determine when their interfaces go up or down.

Note CDP feature is available by installing the following RPMs:

- xr-cdp-7.10.1.19Iv1.0.0-1.x86_64.rpm
- xr-cdp-82eb6a4d2fa15d0e-7.10.1.19Iv1.0.0-1.x86_64.rpm
- xr-cdp-ncs1010-7.10.1.19Iv1.0.0-1.x86_64.rpm

Enable CDP Globally

To enable CDP globally, use the following commands:

configure

cdp

commit

Disable CDP Globally

To disable CDP globally, use the following commands:

configure

no cdp

commit

Enable CDP on Interfaces

To enable CDP on the management interface, use the following commands:

configure

interface mgmtEth *rack/slot/instance/port*

cdp

commit

The following example enables CDP on the management interface.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/1
RP/0/RP0/CPU0:ios(config-if)#cdp
RP/0/RP0/CPU0:ios(config-if)#commit
```

To enable CDP on the Gigabit Ethernet (GE) interface, use the following commands:

configure

interface gigabitEthernet *rack/slot/instance/port*

cdp

commit

The following example enables CDP on the Gigabit Ethernet (GE) interface.

```
RP/0/RP0/CPU0:ios#configure
```

```
RP/0/RP0/CPU0:ios(config)#interface gigabitEthernet 0/0/0/0
RP/0/RP0/CPU0:ios(config-if)#cdp
RP/0/RP0/CPU0:ios(config-if)#commit
```

Modify CDP Default Settings

Use this task to modify CDP parameters such as the default version, holdtime, and timer.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:ios#configure
```

Enters global configuration mode.

Step 2 **cdp advertise v1**

Example:

```
RP/0/RP0/CPU0:ios(config)#cdp advertise v1
```

Configures CDP to use only version 1 (CDPv1) in communicating with neighboring devices.

By default, when CDP is enabled, the device sends CDPv2 packets. CDP also sends and receives CDPv1 packets if the device with which CDP is interacting does not process CDPv2 packets.

In this example, the device is configured to send and receive only CDPv1 packets.

To disable CDP v1, use the **no cdp advertise v1** form of this command.

Step 3 **cdp holdtime** *seconds*

Example:

```
RP/0/RP0/CPU0:ios(config)#cdp holdtime 120
```

Specifies the amount of time that the receiving device holds a CDP packet sent from another device before discarding it.

By default, when CDP is enabled, the receiving device holds a CDP packet for 180 seconds before discarding it. The range of **holdtime** parameter is 10 to 255 seconds.

Note
The CDP hold time must be set to a higher number of seconds than the time between CDP transmissions, which is set with the **cdp timer** command.

Step 4 **cdp timer** *seconds*

Example:

```
RP/0/RP0/CPU0:ios(config)#cdp timer 65
```

Specifies the frequency at which CDP update packets are sent.

By default, when CDP is enabled, CDP update packets are sent at a frequency of once every 60 seconds. The range of **timer** parameter is 5 to 254 seconds.

Note
A lower timer setting causes CDP update packets to be sent more frequently.

Step 5 **commit**

Example:

```
RP/0/RP0/CPU0:ios(config)#commit
```

Saves the configuration changes and remains within the configuration session.

Monitor CDP

Use the **show cdp** command to display global CDP information.

```
RP/0/RP0/CPU0:ios#show cdp
Tue Feb 14 16:59:38.255 UTC
Global CDP information:
        Sending CDP packets every 60 seconds
        Sending a holdtime value of 180 seconds
        Sending CDPv2 advertisements is enabled
```

Use the **show cdp neighbors** command to display detailed information about neighboring devices discovered using CDP.

```
RP/0/RP0/CPU0:ios#show cdp neighbors mgmtEth 0/RP0/CPU0/1
Mon Apr 10 12:30:30.902 UTC
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID       Local Intrfce    Holdtme Capability Platform  Port ID
R1              Mg0/RP0/CPU0/1   172     R          NCS1010   Mg0/RP0/CPU0/1
RP/0/RP0/CPU0:R2#show cdp neighbors
Mon Apr 10 12:30:39.251 UTC
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID       Local Intrfce    Holdtme Capability Platform  Port ID
R1              Mg0/RP0/CPU0/1   164     R          NCS1010   Mg0/RP0/CPU0/1


RP/0/RP0/CPU0:ios#show cdp neighbors mgmtEth 0/RP0/CPU0/1 detail
Mon Apr 10 12:31:23.622 UTC

-------------------------
Device ID: R1
SysName : R1
Entry address(es):
  IPv4 address: 192.168.0.2
  IPv6 address: 2000:110::1
Platform: cisco NCS1010,  Capabilities: Router
Interface: MgmtEth0/RP0/CPU0/1
Port ID (outgoing port): MgmtEth0/RP0/CPU0/1
Holdtime : 120 sec
```

```
Version :
7.10.1.19I

advertisement version: 2
Duplex: full
```

Use the **show cdp entry** *entry-name* command to display information about a specific neighboring device or all the neighboring devices discovered using CDP.

```
RP/0/RP0/CPU0:ios#show cdp entry R1
Mon Apr 10 12:22:22.564 UTC

-------------------------
Device ID: R1
SysName : R1
Entry address(es):
  IPv4 address: 192.168.0.2
  IPv6 address: 2000:110::1
Platform: cisco NCS1010,  Capabilities: Router
Interface: MgmtEth0/RP0/CPU0/1
Port ID (outgoing port): MgmtEth0/RP0/CPU0/1
Holdtime : 121 sec

Version :
7.10.1.19I

advertisement version: 2
Duplex: full


RP/0/RP0/CPU0:ios#show cdp entry *
Mon Apr 10 12:24:59.927 UTC

-------------------------
Device ID: R1
SysName : R1
Entry address(es):
  IPv4 address: 192.168.0.2
  IPv6 address: 2000:110::1
Platform: cisco NCS1010,  Capabilities: Router
Interface: MgmtEth0/RP0/CPU0/1
Port ID (outgoing port): MgmtEth0/RP0/CPU0/1
Holdtime : 143 sec

Version :
7.10.1.19I

advertisement version: 2
Duplex: full
```

Use the **show cdp interface** [*interface-name*] command to display information about the interfaces on which CDP is enabled.

```
RP/0/RP0/CPU0:ios#show cdp interface Mg0/RP0/CPU0/1
Mon Apr 10 12:24:27.253 UTC
MgmtEth0/RP0/CPU0/1 is Up
  Encapsulation ether
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
```

Use the **show cdp traffic** command to display information about the traffic gathered between devices using CDP.

```
RP/0/RP0/CPU0:ios#show cdp traffic
Mon Apr 10 12:32:09.247 UTC

CDP counters :
        Packets output: 11, Input: 5
        Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
        No memory: 0, Invalid packet: 0, Truncated: 0
        CDP version 1 advertisements output: 0, Input: 0
        CDP version 2 advertisements output: 11, Input: 5
        Unrecognize Hdr version: 0, File open failed: 0
```

CHAPTER 12

Daisy Chain

This chapter describes the Daisy Chain optical application for Cisco NCS 1010.

Daisy Chain Overview

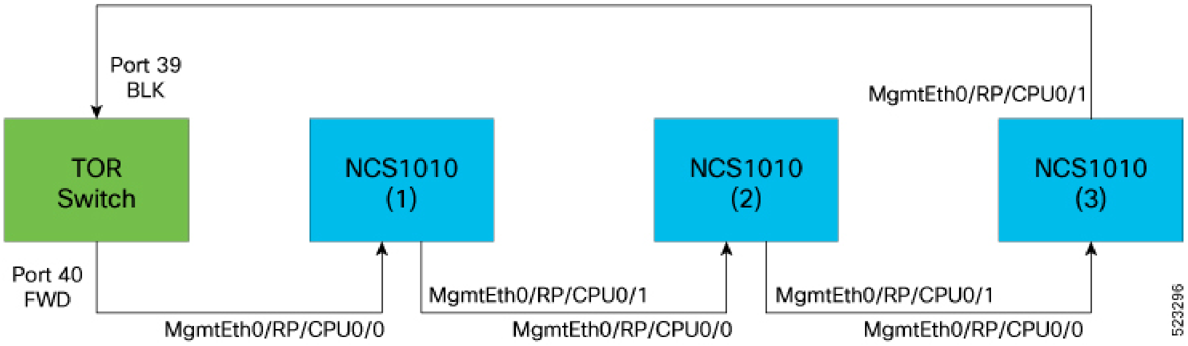
Table 13: Feature History

Feature Name	Release Information	Description
Daisy Chain on NCS 1010 Management Ports	Cisco IOS XR Release 7.10.1	You can now connect NCS 1010 devices in a Daisy Chain topology. Here multiple NCS 1010 devices are connected to form a ring-like topology, and only the first and last nodes are connected to a Top-of-Rack (TOR) switch, thereby reducing the number of connections. The Daisy Chain topology also provides more redundancy as data is transmitted in both directions. The first connection acts as a primary path and carries the traffic whereas the last connection acts as a secondary path. In case the primary path fails, the secondary path serves as its backup for data transmission and allows traffic to continue to transmit in the network.

The daisy chain arrangement allows multiple NCS 1010 nodes to be connected to each other in a ring, where only the first and the last nodes are connected to a TOR switch. The switch allows management of all the NCS 1010 devices in the network and also prevents traffic storm. The data transmitted over the network passes through each node in the ring until it reaches the destination node. This arrangement allows the switch to send data in both directions and prevents one node failure from cutting off certain network parts.

The following diagram shows the Daisy Chain topology where three NCS 1010 nodes are connected to each other over the management ports *0* and *1*.

Figure 15: NCS 1010 in a Daisy Chain Network



Configure Daisy Chain on Management Ports

Before you begin

The following prerequisites must be met before configuring Daisy Chain on NCS1010:

- Enable Storm Control on Switch.
- STP must be running on the TOR switch.
- Daisy chain must be enabled on all the NCS1010 devices in the topology.

Configuring Daisy Chain on managements ports of NCS 1010 devices involves the following tasks:

- Configure IP Address on Management Port
- Configure Daisy Chain

Example

The following example shows how to configure IP address on management port 0 of NCS1010 device:

```
RP/10/RP0:ios(config-if)#int mgmtEth 0/RP0/CPU0/0
RP/10/RP0:ios(config-if)#no shut
RP/10/RP0:ios(config-if)#ipv4 address 192.0.2.1 255.255.255.0
```

Configure Daisy Chain

configure

interface *type* **Rack/Slot/Instance/Port**

no ipv4 address

no ipv6 address

bridge-port routed-interface *type***Rack/Slot/Instance/Port**

Example 1

The following example shows how to configure daisy chain on management port 1 of NCS1010 device:

```
RP/0/RP0:ios(config)# configure
RP/0/RP0:switch(config)# interface mgmtEth0/RP0/CPU0/1
RP/10/RP0:ios(config-if)#no ipv4 address
RP/10/RP0:ios(config-if)#no ipv6 address
RP/10/RP0:ios(config-if)#bridge-port routed-interface mgmtEth 0/RP0/CPU0/0
RP/10/RP0:ios(config-if)#commit
```

Example 2

The following example shows how to configure daisy chain on management port 2 of NCS1010 device:

```
RP/0/RP0:switch(config)# configure
RP/0/RP0:switch(config)# interface mgmtEth0/RP0/CPU0/2
RP/10/RP0:ios(config-if)#no ipv4 address
RP/10/RP0:ios(config-if)#no ipv6 address
RP/10/RP0:ios(config-if)#bridge-port routed-interface mgmtEth 0/RP0/CPU0/0
RP/10/RP0:ios(config-if)#commit
```

Note Daisy chain can be extended to remote node using UDC port and OSC should be active with remote node.

Restriction LLDP and CDP is not supported on the management port if Daisy Chain is configured.

Verify Daisy Chain

To verify daisy chain configuration on management ports of NCS1010 device, use these commands:

show running-config interface*type*

Example

```
RP/0/RP0/CPU0:P2B_DT_02#show running-config interface mgmtEth
Wed Jun  7 12:44:43.673 IST
interface MgmtEth0/RP0/CPU0/0
 ipv4 address 192.0.2.1 255.255.0.0
!
```

```
interface MgmtEth0/RP0/CPU0/1
 bridge-port routed-interface MgmtEth0/RP0/CPU0/0
!
interface MgmtEth0/RP0/CPU0/2
 bridge-port routed-interface MgmtEth0/RP0/CPU0/0
!
```

Enable Storm Control on TOR Switch

When a large number of packets are broadcasted in a short time frame, it results in a traffic storm on a network. In a Daisy Chain network, excessive packet transmission by nodes and subsequent rebroadcasting by other nodes can lead to a traffic storm, overburdening the network.

In the Daisy Chain configuration, data can be transmitted in both directions. One of the Top of the Rack (TOR) switch ports is in the Forward state and carries the traffic whereas the other port is in the Blocked state. Three consecutive hello misses moves the port from Blocked to the Forwarding state.

When the NCS 1010 node reboots, the status of the port is changed from Blocked to Forwarding. Hence, a loop is created momentarily when both the TOR switch ports are in a forwarding state. This loop results in the duplication of packets on the network. To prevent this duplication, storm control must be enabled on the TOR switch.

To enable storm control on a TOR switch, use the following commands:

errdisable recovery interval *value*

errdisable recovery cause storm-control

Example

The following example shows how to enable storm control on a TOR switch:

```
RP/10/RP0:ios(config-if)#errdisable recovery interval 60
RP/10/RP0:ios(config-if)#errdisable recovery cause storm-control
```

Disable DAD on Management Port

By default, IPv6 Duplicate Address Detection (DAD) is enabled on the management ports. Similar to storm control scenario, when IPv6 is configured for a management port, DAD happens due to looping in the network. Since DAD was enabled, management port will be down. In order to avoid management port being down due to momentary looping, DAD must be disabled on the management port on which daisy chain is configured.

To disable DAD on the management port, use the following commands:

configure

interface *type* **Rack/Slot/Instance/Port**

ipv6 nd dad attempts *value*

Example

The following is a sample configuration that disables DAD on management port 1:

```
RP/10/RP0:ios(config-if)#configure
RP/10/RP0:ios(config-if)#interface mgmtEth0/RP0/CPU0/1
RP/10/RP0:ios(config-if)#ipv6 nd dad attempts 1
```

CHAPTER 13

Configure Access Control List

This chapter describes the Access Control List (ACL) and the procedures to configure ACL.

Access Control List

Table 14: Feature History

Feature Name	Release Information	Feature Description
ACL on Management Port	Cisco IOS XR Release 7.11.1	Access Control List feature enables you to permit or deny specific devices to connect to the management port and access NCS 1010 devices. This control enhances network security. Both IPv4 and IPv6 ACLs are supported on the management port. Commands added: • **ipv4-access-list** • **ipv4-access-group** • **show access-lists-ipv4** • **ipv6-access-list** • **ipv6-access-group** • **show access-lists-ipv6**

Access Control Lists

An Access Control List (ACL) is a sequential list consisting of permit and deny statements that apply to IP addresses. ACL performs packet filtering to control the packets that move through the network. These controls allow or restrict the access of devices to the network and limit network traffic.

Access Control Entries

Access Control Entries (ACE) are entries in an ACL that describe the access rights related to a particular security identifier or user. An ACL consists of one or more access control entries (ACE) that collectively define the network traffic profile.

Types of Access Control Lists

ACL types have different set of verification parameters and traffic control methods.

Table 15: Types of ACL

ACL Type	Verifies	Controls traffic by
Standard ACL	only the source IP address of the packets.	comparing the IP address that is configured in the ACL with the source IP address in the packet.

ACL Type	Verifies	Controls traffic by
Extended ACL	• source address • destination address • UDP or TCP port numbers, and • Differentiated Services Code Point (DSCP) of the packets.	comparing the attributes that are defined in the ACL with those in the incoming or outgoing packets.

Benefits of Access Control List

ACL allows you to

- filter incoming or outgoing packets on an interface
- restrict the contents of routing updates
- limit debug output that is based on an address or protocol, and
- control vty access.

Guidelines for Access Control Lists

- Create an ACL before applying it to an interface.
- Write a helpful remark before or after any statement to clarify its purpose.
- Reference an ACL using a command that accepts it after you name the ACL.
- Organize the ACL so that more specific references in a network or subnet appear before more general ones.

Restrictions for Access Control Lists

- Configure an ACL name up to 64 characters.
- Use only letters and numbers for the ACL name.
- Configure ACLs to control ingress and egress traffic on a device, but not traffic originating from the device.

Ingress and Egress Access Control Lists

ACL rules are defined based on the direction of traffic flow relative to the management interface of NCS 1000. ACL is applied either on the ingress or egress interface. Ingress ACL controls traffic that flows from a network to the management interface. Egress ACL controls traffic that comes from the management interface to the network.

The software checks the source address of the packet against the ingress or egress ACL after receiving a packet.

Table 16: Permission and Rejection of Source Address by ACL

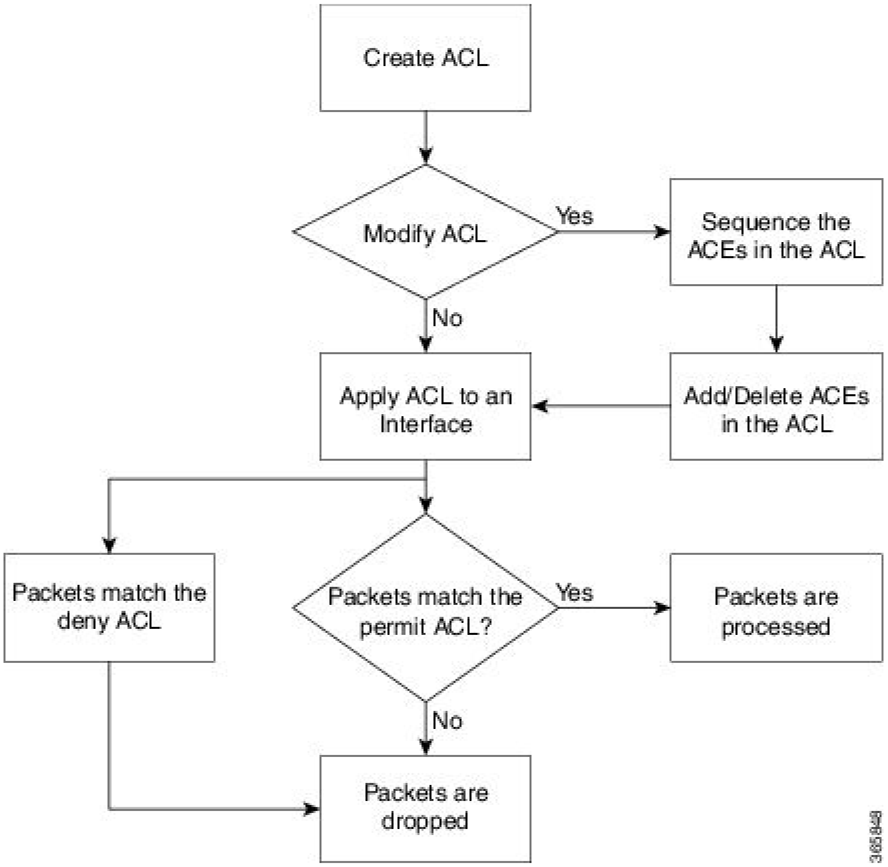
When ACL is ...	And ACL ...	Then ...
Ingress ACL	permits the source address	software continues to process the packet.
	rejects the source address	software discards the packet and returns an Internet Control Message Protocol (ICMP) host unreachable message.
Egress ACL	permits the source address	software sends the packet.
	rejects the source address	software discards the packet and returns an ICMP host unreachable message.

How an Access Control List Works

ACL Workflow

This image illustrates the workflow of an ACL.

Figure 16: ACL Workflow


Create ACL
Modify ACL
Yes
Sequence the ACEs in the ACL
No
Apply ACL to an Interface
Add/Delete ACEs in the ACL
Packets match the deny ACL
Packets match the permit ACL?
Yes
Packets are processed
No
Packets are dropped
365848

Configure IPv4 Standard ACL on Management Ethernet Interface

Follow these steps to configure an IPv4 standard ACL on the management Ethernet interface.

Before you begin

Review the "Guidelines" and "Restrictions" sections of Access Control List, on page 138.

Procedure

Step 1 Run the **ipv4 address** command to configure the management Ethernet interface with an IPv4 address.

Example:

```
RP/0/RP0/CPU0:ios#configure
Thu Oct 19 17:30:23.719 UTC
RP/0/RP0/CPU0:ios(config)#
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 192.0.2.127 255.255.255.0
RP/0/RP0/CPU0:ios(config-if)#no shutdown
RP/0/RP0/CPU0:ios(config-if)#commit
```

```
Thu Oct 19 17:31:25.127 UTC
RP/0/RP0/CPU0:ios(config-if)#exit
```

Step 2 Run the **show ipv4 interface brief** command to verify whether the management Ethernet interface is up.

The entry highlighted in bold shows the status of management Ethernet interface as **Up**.

Example:

```
RP/0/RP0/CPU0:ios(config)#show ipv4 interface brief
Thu Oct 19 17:32:10.998 UTC

Interface                      IP-Address      Status          Protocol Vrf-Name
Loopback0                      192.0.2.121     Up              Up       default
GigabitEthernet0/0/0/0         192.0.2.123     Up              Up       default
GigabitEthernet0/0/0/2         192.0.2.122     Up              Up       default
MgmtEth0/RP0/CPU0/0            192.0.2.127       Up              Up       default
PTP0/RP0/CPU0/0                unassigned      Shutdown        Down     default
MgmtEth0/RP0/CPU0/1            192.0.2.124     Up              Up       default
PTP0/RP0/CPU0/1                unassigned      Shutdown        Down     default
MgmtEth0/RP0/CPU0/2            192.0.2.1       Down            Down     default
```

Step 3 Run the **ipv4 access-list** command to configure an IPv4 ACL.

Example:

```
/* Configure an IPv4 ingress ACL */
RP/0/RP0/CPU0:ios(config)# ipv4 access-list V4-ACL-INGRESS
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 10 permit tcp 192.0.2.2 255.255.255.0 any
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 20 deny udp any any
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 30 permit ipv4 192.0.2.64 255.255.255.0 any
RP/0/RP0/CPU0:ios(config-ipv4-acl)# commit
Thu Oct 19 18:31:25.127 UTC

/* Configure an IPv4 egress ACL */
RP/0/RP0/CPU0:ios(config)# ipv4 access-list V4-ACL-EGRESS
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 10 permit ipv4 203.0.113.1 255.255.255.0 192.0.2.1 0.255.255.255
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 20 deny udp any any
RP/0/RP0/CPU0:ios(config-ipv4-acl)# commit
Thu Oct 19 18:31:25.127 UTC
```

Step 4 Run the **show access-lists ipv4** command to verify the ACL creation.

The entries highlighted in bold show the successful creation of ingress ACL and egresss ACL.

Example:

```
/* Verify the ingress ACL creation */
RP/0/RP0/CPU0:ios(config)# show access-lists ipv4
Thu Oct 19 18:32:25.127 UTC
ipv4 access-list V4-ACL-INGRESS
 10 permit tcp 192.0.2.2 255.255.255.0 any
 20 deny udp any any
 30 permit ipv4 192.0.2.64 255.255.255.0 any

/* Verify the egress ACL creation */
RP/0/RP0/CPU0:ios(config)# show access-lists ipv4
Thu Oct 19 18:32:25.127 UTC
ipv4 access-list V4-ACL-EGRESS
```

```
 10 permit ipv4 203.0.113.1 255.255.255.0 192.0.2.1 255.255.255.0
 20 deny ipv4 any any
```

Step 5 Run the **ipv4 access-group** command to apply the ACL to the management Ethernet interface.

Example:

```
/* Apply the ingress ACL to the mgmtEth interface */
RP/0/RP0/CPU0:ios(config)# interface mgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:ios(config-if)# ipv4 access-group V4-ACL-INGRESS ingress
RP/0/RP0/CPU0:ios(config-if)#commit
Thu Oct 19 18:34:25.127 UTC
RP/0/RP0/CPU0:ios(config-if)#exit


/* Apply the egress ACL to the mgmtEth interface */
RP/0/RP0/CPU0:ios(config)# interface mgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:ios(config-if)# ipv4 access-group V4-ACL-EGRESS egress
RP/0/RP0/CPU0:ios(config-if)# commit
Thu Jul 11 09:19:49.569 UTC
RP/0/RP0/CPU0:ios(config-if)# exit
```

Step 6 Run the **show ipv4 interface** command to verify whether the ACL has been successfully applied to the management Ethernet interface.

The entry highlighted in bold shows the ACL has been successfully applied to the management Ethernet interface.

Example:

```
/* Verify if the ingress ACL has been successfully applied to the mgmtEth interface */
RP/0/RP0/CPU0:ios#show ipv4 interface mgmtEth 0/RP0/CPU0/0
Fri Oct 20 05:07:06.383 UTC
MgmtEth0/RP0/CPU0/0 is Up, ipv4 protocol is Up
  Vrf is default (vrfid 0x60000000)
  Internet address is 192.0.2.127/16
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  common access list is not set, access list is V4-ACL-INGRESS
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000

/* Verify if the egress ACL has been successfully applied to the mgmtEth interface */

RP/0/RP0/CPU0:ios#show ipv4 interface mgmtEth 0/RP0/CPU0/0
Fri Oct 20 05:07:06.383 UTC
MgmtEth0/RP0/CPU0/0 is Up, ipv4 protocol is Up
  Vrf is default (vrfid 0x60000000)
  Internet address is 192.0.2.127/16
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is V4-ACL-EGRESS
  Inbound  common access list is not set, access list is not set
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000
```

You have successfully configured an IPv4 standard ACL on the management Ethernet interface.

Configure IPv6 Standard Access Control List on Management Ethernet Interface

Follow these steps to configure an IPv6 standard ACL on the management Ethernet interface.

Before you begin

Review the "Guidelines" and "Restrictions" sections of Access Control List, on page 138.

Procedure

Step 1 Run the **ipv6 address** command to configure the management Ethernet interface with an IPv6 address.

Example:

```
RP/0/RP0/CPU0:ios#configure
Fri Oct 20 05:25:58.785 UTC
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/1
RP/0/RP0/CPU0:ios(config-if)#ipv6 address 2001::1/64
RP/0/RP0/CPU0:ios(config-if)#no shutdown
RP/0/RP0/CPU0:ios(config-if)#commit
Fri Oct 20 05:26:13.669 UTC
RP/0/RP0/CPU0:ios(config-if)#exit
```

Step 2 Run the **show ipv6 interface** command to verify whether the management Ethernet interface is up.

The entry highlighted in bold shows the status of management Ethernet interface as **Up**.

Example:

```
RP/0/RP0/CPU0:ios(config)#show ipv6 interface mgmtEth 0/RP0/CPU0/1 brief
Fri Oct 20 05:26:52.612 UTC
MgmtEth0/RP0/CPU0/1    [Up/Up]
    fe80::3afd:f8ff:fe66:872
    2001::1
```

Step 3 Run the **ipv6 access-list** command to configure an IPv6 ACL.

Example:

```
/* Configure an IPv6 ingress ACL */
RP/0/RP0/CPU0:ios(config)#ipv6 access-list V6-INGRESS-ACL
RP/0/RP0/CPU0:ios(config-ipv6-acl)#10 permit ipv6 any any
RP/0/RP0/CPU0:ios(config-ipv6-acl)#20 deny udp any any
RP/0/RP0/CPU0:ios(config-ipv6-acl)#commit
Fri Oct 20 05:28:46.664 UTC
```

```
/* Configure an IPv6 egress ACL */
Router(config)# ipv6 access-list V6-EGRESS-ACL
Router(config-ipv6-acl)# 10 permit ipv6 any any
Router(config-ipv6-acl)# 20 deny udp any any
Router(config-ipv6-acl)# commit
Thu Jul 11 09:50:40.566 UTC
```

Step 4 Run the **show access-lists ipv6** command to verify the ACL creation.

The entries highlighted in bold show the successful creation of ingress ACL and egresss ACL.

Example:

```
/* Verify the ingress ACL creation */
RP/0/RP0/CPU0:ios(config)# show access-lists ipv6
Fri Oct 20 05:29:01.125 UTC

ipv6 access-list V6-INGRESS-ACL
 10 permit ipv6 any any
 20 deny udp any any


/* Verify the egress ACL creation */
RP/0/RP0/CPU0:ios(config)# show access-lists ipv6
Fri Oct 20 05:29:01.125 UTC

ipv6 access-list V6-EGRESS-ACL
 10 permit ipv6 any any
 20 deny udp any any
```

Step 5 Run the **ipv6 access-group** command to apply the ACL to the management Ethernet interface.

Example:

```
/* Apply the ingress ACL to the mgmtEth interface */
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/1
RP/0/RP0/CPU0:ios(config-if)#ipv6 access-group V6-INGRESS-ACL ingress
RP/0/RP0/CPU0:ios(config-if)#commit
Fri Oct 20 05:37:32.738 UTC
RP/0/RP0/CPU0:ios(config-if)#exit

/* Apply the egress ACL to the mgmtEth interface */
Router(config)# interface mgmtEth 0/RP0/CPU0/1
Router(config-if)# ipv6 access-group V6-EGRESS-ACL egress
Router(config-if)# commit
Thu Jul 11 09:52:57.751 UTC
Router(config-if)# exit
```

Step 6 Run the **show ipv6 interface** command to verify whether the ACL has been successfully applied to the management Ethernet interface.

The entry highlighted in bold shows the ACL has been successfully applied to the management Ethernet interface.

Example:

```
/* Verify if the ingress ACL has been successfully applied to the mgmtEth interface */
RP/0/RP0/CPU0:ios#show ipv6 interface mgmtEth 0/RP0/CPU0/1
Fri Oct 20 05:38:00.753 UTC
MgmtEth0/RP0/CPU0/1 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
  IPv6 is enabled, link-local address is fe80::3afd:f8ff:fe66:872
  Global unicast address(es):
    2001::1, subnet is 2001::/64
  Joined group address(es): ff02::1:ff00:1 ff02::1:ff66:872 ff02::2
      ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND cache entry limit is 1000000000
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 160 to 240 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
  Outgoing access list is not set
  Inbound  common access list is not set, access list is V6-INGRESS-ACL
  Table Id is 0xe0800000
  Complete protocol adjacency: 0
  Complete glean adjacency: 0
  Incomplete protocol adjacency: 0
  Incomplete glean adjacency: 0
  Dropped protocol request: 0
  Dropped glean request: 0
  RA DNS Server Address Count: 0
  RA DNS Search list Count: 0

/* Verify if the egress ACL has been successfully applied to the mgmtEth interface */

RP/0/RP0/CPU0:ios#show ipv6 interface mgmtEth 0/RP0/CPU0/1
Fri Oct 20 05:38:00.753 UTC
MgmtEth0/RP0/CPU0/1 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
  IPv6 is enabled, link-local address is fe80::3afd:f8ff:fe66:872
  Global unicast address(es):
    2001::1, subnet is 2001::/64
  Joined group address(es): ff02::1:ff00:1 ff02::1:ff66:872 ff02::2
      ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND cache entry limit is 1000000000
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 160 to 240 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
  Outgoing access list is V6-EGRESS-ACL
  Inbound  common access list is not set, access list is not set
  Table Id is 0xe0800000
  Complete protocol adjacency: 0
  Complete glean adjacency: 0
  Incomplete protocol adjacency: 0
  Incomplete glean adjacency: 0
  Dropped protocol request: 0
  Dropped glean request: 0
  RA DNS Server Address Count: 0
  RA DNS Search list Count: 0
```

You have successfully configured an IPv6 standard ACL on the management Ethernet interface.

Configure an Extended Access Control List

To configure an extended ACL, you must create an ACL and specify the condition to allow or deny the network traffic.

Before you begin

Review the "Guidelines" and "Restrictions" sections of Access Control List, on page 138.

Procedure

Step 1 Run the **ipv4 access-list** command to configure the ACL.

Example:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ipv4 access-list acl_1
```

Step 2 Run the **permit** and **deny** commands to specify the condition to allow or deny the network traffic

Example:

```
RP/0/RP0/CPU0:ios(config-ipv4-acl)#10 permit 172.16.0.0 0.0.255.255
RP/0/RP0/CPU0:ios(config-ipv4-acl)#20 deny 192.168.34.0 0.0.0.255
RP/0/RP0/CPU0:ios(config-ipv4-acl)#commit
```

Step 3 Run the **show access-lists** command to verify the ACL creation.

Example:

```
RP/0/RP0/CPU0:ios#show access-lists ipv4 acl_1
Fri Oct 20 06:22:17.223 UTC
ipv4 access-list acl_1
 10 permit ipv4 172.16.0.0 0.0.255.255 any
 20 deny ipv4 192.168.34.0 0.0.0.255 any
```

You have successfully configured an extended ACL on the management Ethernet interface.

Modify an Access Control List

Before you begin

Review the "Guidelines" and "Restrictions" sections of Access Control List, on page 138.

Procedure

Step 1 Run the **ipv4 access-list** command to configure the ACL.

Example:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ipv4 access-list acl_1
```

Step 2 Run the **permit** command to add entries to the ACL.

Example:

```
RP/0/RP0/CPU0:ios(config-ipv4-acl)#10 permit ip host 10.3.3.3 host 172.16.5.34
RP/0/RP0/CPU0:ios(config-ipv4-acl)#20 permit icmp any any
RP/0/RP0/CPU0:ios(config-ipv4-acl)#30 permit tcp any host 10.3.3.3
RP/0/RP0/CPU0:ios(config-ipv4-acl)#end
```

Step 3 Run the **show access-lists** command to verify the ACL creation.

Example:

```
RP/0/RP0/CPU0:ios#show access-lists ipv4 acl_1
ipv4 access-list acl_1
10 permit ip host 10.3.3.3 host 172.16.5.34
20 permit icmp any any
30 permit tcp any host 10.3.3.3
```

Step 4 Run the **permit** command to modify the ACL.

Example:

```
*/Add new entries, one with a sequence number "15" and another without a sequence number to the ACL.
 Delete an entry with the sequence number "30":*/
RP/0/RP0/CPU0:ios(config)#ipv4 access-list acl_1
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 15 permit 10.5.5.5 0.0.0.255
RP/0/RP0/CPU0:ios(config-ipv4-acl)# no 30
RP/0/RP0/CPU0:ios(config-ipv4-acl)# permit 10.4.4.4 0.0.0.255
RP/0/RP0/CPU0:ios(config-ipv4-acl)# commit
```

When an entry is added without a sequence number, it is automatically given a sequence number that puts it at the end of the access list. Because the default increment is 10, the entry will have a sequence number 10 higher than the last entry in the existing access list.

Step 5 Run the **show access-lists** command to verify the ACL creation.

Example:

```
RP/0/RP0/CPU0:ios#show access-lists ipv4 acl_1
ipv4 access-list acl_1
 10 permit ipv4 host 10.3.3.3 host 172.16.5.34

15 permit 10.5.5.5 0.0.0.255---*/newly added ACL (with the sequence number)*/
20 permit icmp any any
30 permit ipv4 10.4.4.0 0.0.0.255 any ---*/newly added ACL (without the sequence number)*/

*/The entry with the sequence number 30, that is, "30 permit tcp any host 10.3.3.3" is deleted from
 the ACL*/
```

You have successfully modified the ACL in operation.

CHAPTER 14

Remote Node Management in NCS 1010

This chapter describes how to manage an ILA node remotely in NCS 1010 by the remote node management feature.

Understanding Remote Node Management Using OSC

The remote node management feature in NCS 1010 allows you to remotely manage an ILA node that is not connected to a management network through an OLT gateway node over Optical Supervisory Channel (OSC) interface. The OLT node is connected to a management network and manages ILA node remotely. If the OLT node link is down, the ILA node cannot be accessible.

Prerequisites

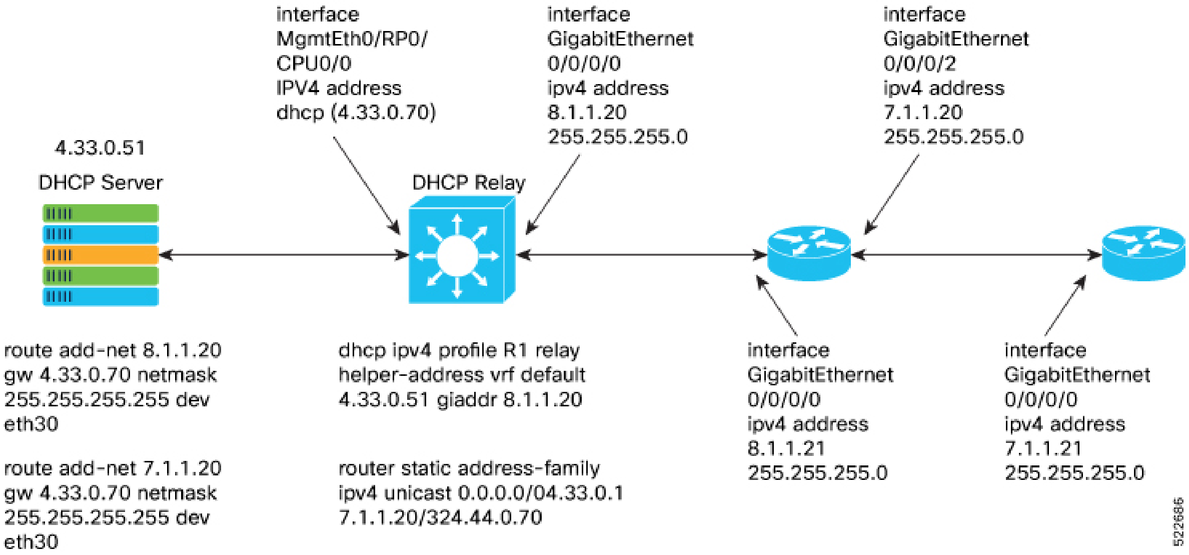
The remote node management for ILA node works only if the following conditions are met:

- The DSCP relay configuration for OLT node must be present. See DHCP Relay Configuration for OLT Node, on page 150
- The loopback address and IP address must be present for OSC interface. See Loopback IP address for OSC Interface, on page 151
- The OSPF neighbor discovery must be successful between OLT and ILA nodes. See OSPF Neighbor Discovery, on page 151

DHCP Relay Configuration for OLT Node

The OLT node must be configured with the DHCP management connection to manage the ILA node remotely over OSC interface.

Figure 17: DHCP Relay Configuration for OLT Node



interface
MgmtEth0/RP0/
CPU0/0
IPV4 address
dhcp (4.33.0.70)

interface
GigabitEthernet
0/0/0/0
ipv4 address
8.1.1.20
255.255.255.0

interface
GigabitEthernet
0/0/0/2
ipv4 address
7.1.1.20
255.255.255.0

route add-net 8.1.1.20
gw 4.33.0.70 netmask
255.255.255.255 dev
eth30

route add-net 7.1.1.20
gw 4.33.0.70 netmask
255.255.255.255 dev
eth30

dhcp ipv4 profile R1 relay
helper-address vrf default
4.33.0.51 giaddr 8.1.1.20

router static address-family
ipv4 unicast 0.0.0.0/04.33.0.1
7.1.1.20/324.44.0.70

interface
GigabitEthernet
0/0/0/0
ipv4 address
8.1.1.21
255.255.255.0

interface
GigabitEthernet
0/0/0/0
ipv4 address
7.1.1.21
255.255.255.0

Following is the sample DHCP relay configuration for the OLT gateway node:

```
RP/0/RP0/CPU0:P2B_DT_02#sh running-config int mgmtEth 0/RP0/CPU0/2
Thu Jun  9 06:37:59.071 UTC
interface MgmtEth0/RP0/CPU0/2
 ipv4 address 192.168.1.1 255.255.255.252
!

RP/0/RP0/CPU0:P2B_DT_02#

RP/0/RP0/CPU0:P2C_DT_02#

RP/0/RP0/CPU0:P2B_DT_02#sh running-config dhcp ipv4
Thu Jun  9 06:28:51.879 UTC
dhcp ipv4
 profile R1 relay
  helper-address vrf default 10.4.33.51 giaddr 10.8.1.20
 !
 interface GigabitEthernet0/0/0/0 relay profile R1
!
```

In the above sample CLI,

- **10.4.33.51** is the DHCP server IP address
- **10.8.1.20** is the OSC interface IP address that going to ILA node from OLT node
- **0/0/0/0** is the interface number
- **R1** is the profile

Sample command for DHCP server:

```
3) Config on dhcp server:
route add -net <OLT-OSCip> gw <OLT-MGMTip> netmask 255.255.255.255 dev eth3

route add -net 10.8.1.20 gw 10.4.33.70 netmask 255.255.255.255 dev eth3
route add -net 10.7.1.20 gw 10.4.33.70 netmask 255.255.255.255 dev eth3
Config on OLT:
dhcp ipv4 profile R1 relay helper-address vrf default 10.4.33.51 giaddr 10.8.1.20
router static
address-family ipv4 unicast
  0.0.0.0/0 10.4.33.1
  10.7.1.20/32 10.4.44.70
interface MgmtEth0/RP0/CPU0/0
ipv4 address dhcp
interface GigabitEthernet0/0/0/0
ipv4 address 10.8.1.20 255.255.255.0
```

Loopback IP address for OSC Interface

The loopback IP address must be mapped for the OSC interface.

Note The loopback IP address is essential as it acts as a router ID for the OSPF configuration. Many communication protocols such as: SSH, GRPC and optical applications, and remote login need the router ID for OSPF configuration. .

Following is the sample of loopback and IP address for OSC interface:

```
RP/0/RP0/CPU0:P2B_DT_02#sh running-config interface loopback 0
Thu Jun  9 06:29:00.447 UTC
interface Loopback0
 ipv4 address 10.3.3.20 255.255.255.255
!
```

OSPF Neighbor Discovery

The OSPF neighbor discovery indicates the successful connection between OLT and ILA node.

Following is the sample CLI:

```
RP/0/RP0/CPU0:IOS#show ospf neighbor
Tue Jul 26 07:31:29.532 UTC
* Indicates MADJ interface
# Indicates Neighbor awaiting BFD session up
Neighbors for OSPF 1
Neighbor ID     Pri   State           Dead Time   Address         Interface
198.51.100.1        1    FULL/  -        00:00:35    10.8.1.21      GigabitEthernet0/0/0/0
Neighbor is up for 00:00:42

Total neighbor count: 1
RP/0/RP0/CPU0:IOS#
```

In the above CLI,

- **198.51.100.1** is the neighbor IP address

- **10.8.1.21** is the OSC interface IP address

Configure ILA Node

The following is a sample command for ILA node configuration:

```
interface GigabitEthernet0/0/0/0
ipv4 address 10.8.1.21 255.255.255.0
!
interface GigabitEthernet0/0/0/2
ipv4 address 10.7.1.21 255.255.255.0

router ospf 1
distribute link-state
network point-to-point
redistribute connected
area 0
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/0
  !
  interface GigabitEthernet0/0/0/2
```

Configure OLT Node

The following is a sample command to configure the OLT node with loopback ip:

```
Configure

interface Loopback0
ipv4 address 10.3.3.21 255.255.255.255
!
interface GigabitEthernet0/0/0/0
ipv4 address 10.7.1.20 255.255.255.0
router ospf 1
distribute link-state
network point-to-point
area 0
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/0
```

-

CHAPTER 15

Automated File Management

Automated File Management System

Table 17: Feature History

Feature Name	Release Information	Feature Description
Automated File Management System	Cisco IOS XR Release 24.4.1	The new Automated File Management System is designed for efficient file handling on each node. This system automatically archives older files and removes them from local nodes to free up valuable SSD space. It manages the following types of files: • System-generated log files • Showtech-related residual files

The automated file management system archives older files to free up valuable SSD space by deleting them from the local nodes.

Types of files

The SSD stores two types of files that are generated by

- **User**: creates and owns files for requirement purposes and deletes the files when are no longer needed.
- **System**: organizes files automatically based on the file content and the application that created the content such as .log and showtech-related residual files.

  Automated file management is applicable for the system-generated files.

How the automated file management system works

These stages describe how the automated file management works for various files.

Log files

The NCS 1014 system uses the log rotation configurations to manage log files as required.

1. The system checks for the *.log* files exceeding 10 MB file size.

Note This threshold is applicable for `/tmp` folder files only. For files in other folders, the system uses a different threshold and follows the same process.

2. After locating the file, the system
   a. archives that file with .gz extension, and
   b. creates a new .log file with the same name.
3. The system then maintains one active log file and two archived files.
4. When new files are archived, the system deletes the oldest archived file to make sure that only the two most recently archived files and the current log file are retained.

showtech-related residual log files

1. The system generates the residual files during the collection of logs when **Showtech logs** is in operation.
2. After collecting the logs, the system automatically removes these residual files to conserve space.