



Cisco NCS 1010 System Setup and Software Installation Guide, IOS XR Release 7.11.x

First Published: 2023-12-08

Last Modified: 2024-05-29

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 NON-INFRINGEMENT 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.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2024 Cisco Systems, Inc. All rights reserved.



CONTENTS

Full Cisco Trademarks with Software License ?

CHAPTER 1

Cisco NCS 1010 Optical Line System - An Overview 1

NCS 1010 Chassis and Line Cards 1

Line Cards 2

OLT-C Line Card 2

OLT-R-C Line Card 2

ILA-C Line Card 3

ILA-R-C Line Card 3

ILA-2R-C Line Card 3

OLT-L Line Card 4

ILA-L Line Card 4

External Interface Timing Unit 4

Cisco NCS 1000 Passive Modules 5

Cisco NCS 1000 Breakout Patch Panel 5

NCS1K-BRK-8 5

NCS1K-BRK-24 6

Cisco NCS 1000 32-Channel Mux/Demux Patch Panels 6

Document Objective 7

Document Organization 7

CHAPTER 2

Bring-up Cisco NCS 1010 9

Boot NCS 1010 9

Boot NCS 1010 Using USB Drive 10

DHCP Configuration 11

Introduction to DHCP Relay 11

- Prerequisites for Configuring DHCP Relay Agent 12
- Limitations for DHCP Relay Feature 12
- Configuring and Enabling the DHCP Relay Agent 13
 - DHCP Client 13
- Boot Using iPXE 14
 - Setup DHCP Server 14
 - Boot Using iPXE 15
- Boot Using Zero Touch Provisioning 19
 - Fresh Boot Using DHCP 20
 - Build your Configuration File 21
 - Configure ZTP BootScript 22
 - Invoke ZTP Manually through CLI 24
 - Invoke ZTP Through Reload 25
 - ZTP Logging 26
 - Generate Tech Support Information for ZTP 27
- Configure Management Interface 28
 - Link Layer Discovery Protocol Support on Management Interface 29
- Configure Telnet 33
- Configure SSH 33
- Configure NTP Server 35
 - Understand NTP 35
 - Synchronize Clock with NTP Server 35
 - Verify the Status of the External Reference Clock 37
 - NTP Troubleshooting Information 38

CHAPTER 3 **Disaster Recovery 39**

- Overview 39
- CPU Replacement Considerations 39
- Health Check of Backup ISO Image 39

CHAPTER 4 **Perform Preliminary Checks 41**

- Verify Status of Hardware Components 41
- Verify Inventory 42
- Verify Software Version 43

Verify Firmware Version	44
Verify Management Interface Status	46
Verify Alarms	47
Verify Environmental Parameters	48
Verify Context	54
Verify Core Files	54
Verify Memory Information	54

CHAPTER 5	Upgrade Software and FPD	57
	Upgrade Software	57
	Software Upgrade and Downgrade Matrix	59
	Install Packages and RPMs	59
	NCS 1010 FPD	63
	Verify if an FPD Upgrade is Required	68
	Upgrade FPDs Manually	70
	Upgrade FPDs Automatically	71
	Release 7.10.1 Caveats	71

CHAPTER 6	Understanding Remote Node Management Using OSC	73
	Prerequisites	73
	DHCP Relay Configuration for OLT Node	73
	Loopback IP address for OSC Interface	75
	OSPF Neighbor Discovery	75
	Configure ILA Node	76
	Configure OLT Node	76

CHAPTER 7	Configuring BGP	77
	BGP Overview	77
	Prerequisites for Implementing BGP	78
	BGP Router Identifier	78
	Configuring BGP	79

CHAPTER 8	Configure CDP	83
	Enable CDP Globally	84

Disable CDP Globally 84
 Enable CDP on Interfaces 84
 Modify CDP Default Settings 85
 Monitor CDP 86

CHAPTER 9

Daisy Chain 89

Daisy Chain Overview 89
 Configure Daisy Chain on Management Ports 90
 Verify Daisy Chain 91
 Enable Storm Control on TOR Switch 92
 Disable DAD on Management Port 92

CHAPTER 10

Configure Access Control List 95

Understand Access Control List 95
 Ingress and Egress Access Control Lists 97
 How an Access Control List Works 97
 Configure IPv4 Standard Access Control List on Management Ethernet Interface 98
 Configure IPv6 Standard Access Control List on Management Ethernet Interface 101
 Configure an Extended Access Control List 103
 Modify an Access Control List 104



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, on page 1](#)
- [Cisco NCS 1000 Passive Modules, on page 5](#)
- [Document Objective, on page 7](#)
- [Document Organization, on page 7](#)

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

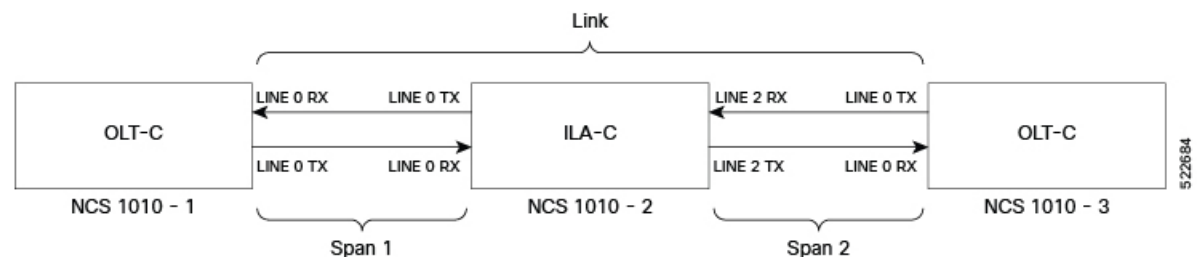
ILA-C Line Card

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.

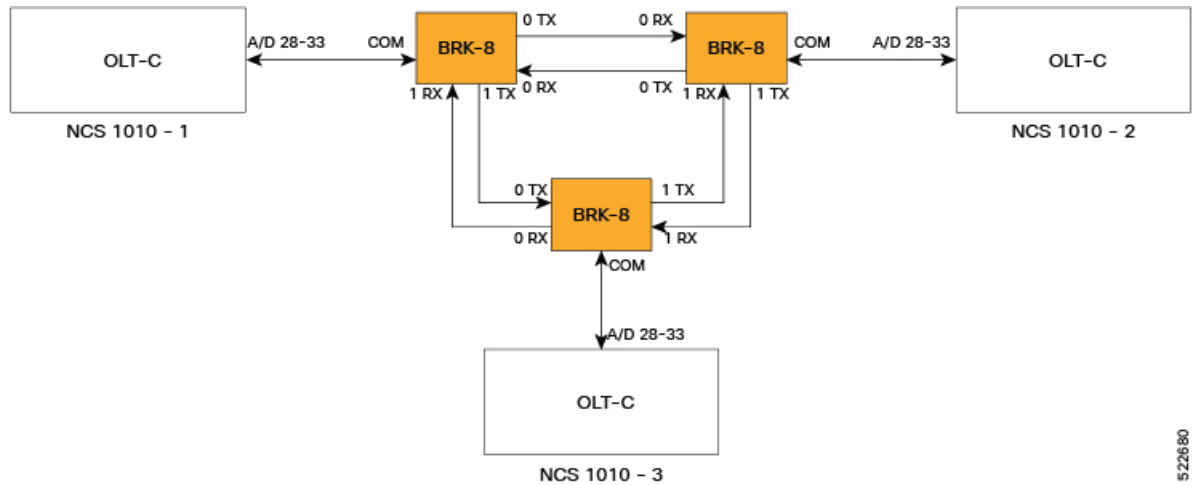
The breakout panel supports the following breakout modules:

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



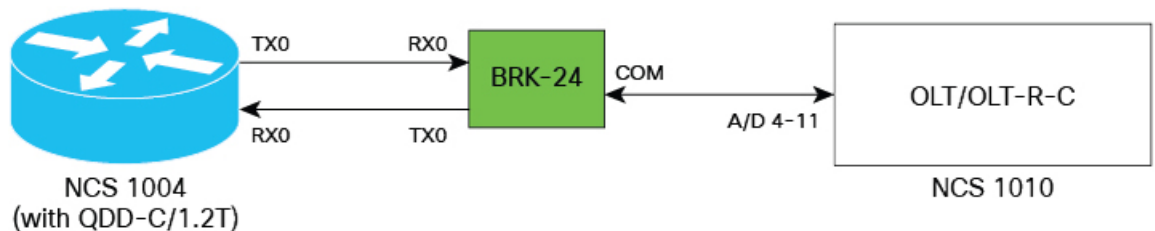
522680

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



522681

Cisco NCS 1000 32-Channel Mux/Demux Patch Panels

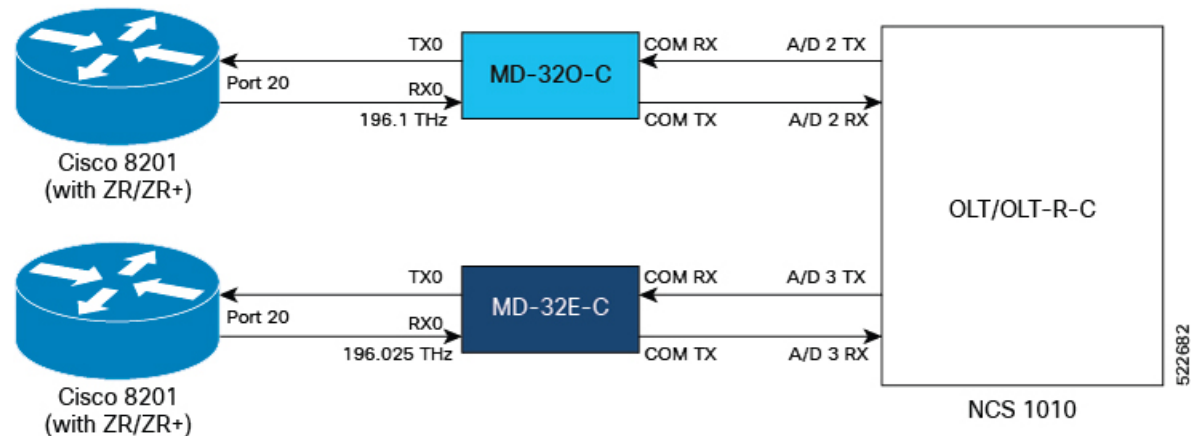
The Cisco NCS 1000 32-Channel Mux/Demux patch panels are a pair of passive Athermal Arrayed Waveguide Grating (AAWG) based modules (PIDs NCS1K-MD-32O-C and NCS1K-MD-32E-C). Each Mux/Demux panel has 32 channels and works as an add/drop unit for the OLT-C and OLT-R-C line cards. Each Mux/Demux panel allows 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
Bring-up Cisco NCS 1010, on page 9	Various methods such as iPXE, Zero Touch Provisioning, and USB drive to boot up the Cisco NCS 1010 system .
Disaster Recovery, on page 39	The disaster recovery process and the health check of backup ISO image.

Chapter	Description
Perform Preliminary Checks, on page 41	Preliminary checks to be performed after successfully logging into the console and the suggested corrective actions if any setup issue is detected.
Upgrade Software and FPD, on page 57	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 89	Describes how you can connect NCS 1010 devices in a Daisy Chain topology.
Configure Access Control List, on page 95	Procedures to configure access control lists (ACL).



CHAPTER 2

Bring-up Cisco NCS 1010

After installing the hardware, boot the Cisco NCS 1010 system. You can connect to the XR console port and power on the system. NCS 1010 completes the boot process using the pre-installed operating system (OS) image. If no image is available, NCS 1010 can be booted using the iPXE boot or an external bootable USB drive.

After booting, create the root username and password, and then use it to log on to the XR console.



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.

- [Boot NCS 1010, on page 9](#)
- [Configure Management Interface, on page 28](#)
- [Configure Telnet, on page 33](#)
- [Configure SSH, on page 33](#)
- [Configure NTP Server, on page 35](#)

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.

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 115,200 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.

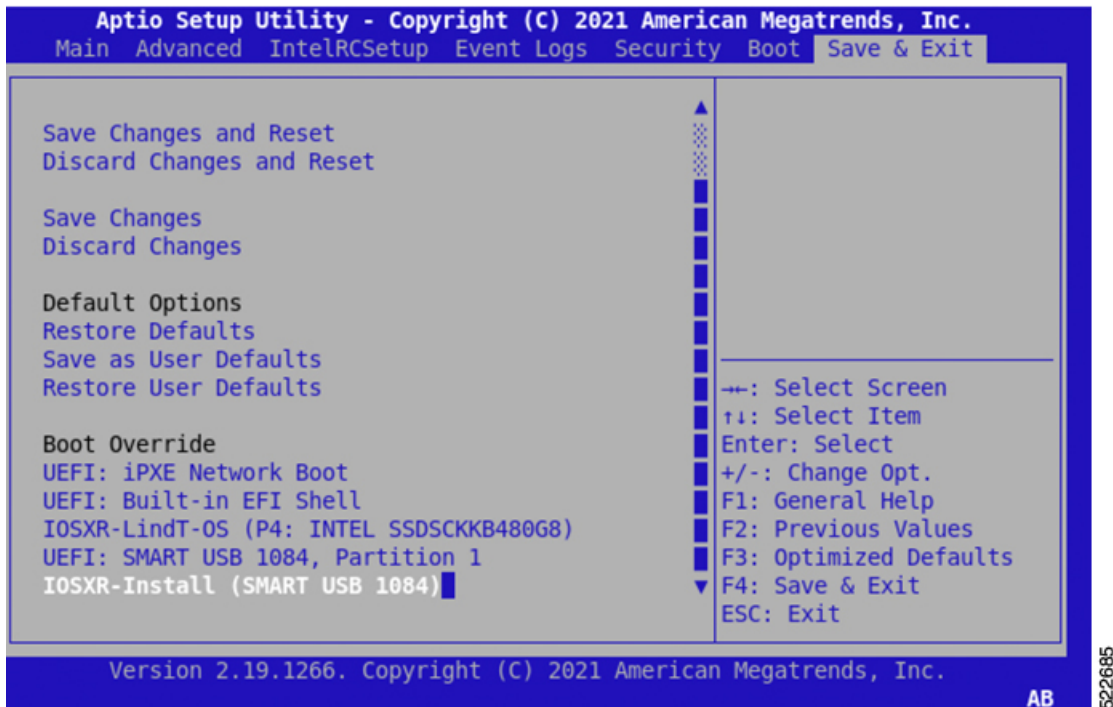
You can complete this task using the Windows, Linux, or MAC operating systems available on your local machine. The exact operation to be performed for each generic step that is outlined here depends on the operating system in use.

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*.

-
- 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.

DHCP Configuration

DHCP configuration is required for both manual configuration and ZTP configuration. Follow the below sections to set up DHCP for booting NCS 1010 using ZTP and iPXE.

Introduction to DHCP Relay

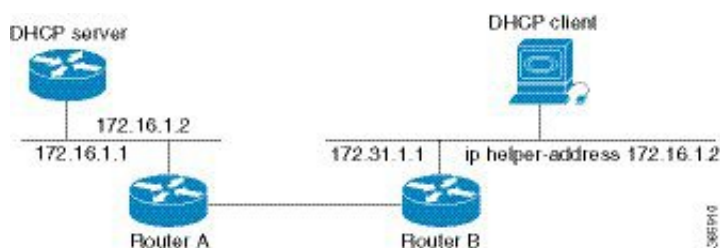
A DHCP relay agent is a host that forwards DHCP packets between clients and servers that do not reside on a shared physical subnet. Relay agent forwarding is distinct from the normal forwarding of an IP router where IP datagrams are switched between networks transparently.

DHCP clients use User Datagram Protocol (UDP) broadcasts to send DHCPDISCOVER messages when they lack information about the network to which they belong.

If a client is on a network segment that does not include a server, a relay agent is needed on that network segment to ensure that DHCP packets reach the servers on another network segment. UDP broadcast packets are not forwarded, because most routers are not configured to forward broadcast traffic. You can configure a DHCP relay agent to forward DHCP packets to a remote server by configuring a DHCP relay profile and configure one or more helper addresses in it. You can assign the profile to an interface or a VRF.

The figure below demonstrates the process. The DHCP client broadcasts a request for an IP address and additional configuration parameters on its local LAN. Acting as a DHCP relay agent, Router B picks up the broadcast, changes the destination address to the DHCP server's address and sends the message out on another interface. The relay agent inserts the IP address of the interface, on which the DHCP client's packets are received into the gateway address (giaddr) field of the DHCP packet, which enables the DHCP server to determine which subnet should receive the offer and identify the appropriate IP address range. The relay agent unicasts the messages to the server address, in this case 172.16.1.2 (which is specified by the helper address in the relay profile).

Figure 5: Forwarding UDP Broadcasts to a DHCP Server Using a Helper Address



Prerequisites for Configuring DHCP Relay Agent

The following are the prerequisites to configure a DHCP relay agent:

- 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.
- A configured and running DHCP client and DHCP server.
- Connectivity between the relay agent and DHCP server

Limitations for DHCP Relay Feature

These are the limitations for implementing DHCP relay feature:

- The multicast addresses are not supported. The **helper-address** command in DHCP relay profile submode supports valid unicast IP address as the helper address.



Note Configuring the **helper-address** command directly (not using profile) under a interface (such as BVI interface) is not supported.

- Only interface-id and remote-id DHCP option code are added by a relay agent while forwarding the packet to a DHCP server.



Note Configuring DHCP option code is not supported in DHCP relay profile submode.

Configuring and Enabling the DHCP Relay Agent

Configuration Example

```
RP/0/RP0/CPU0:ios# configure
RP/0/RP0/CPU0:ios(config)# dhcp ipv4
RP/0/RP0/CPU0:ios(config-dhcpv4)# profile r1 relay
RP/0/RP0/CPU0:ios(config-dhcpv4-relay-profile)# helper-address vrf default 198.51.100.1
giaddr 198.51.100.3
RP/0/RP0/CPU0:ios(config-dhcpv4-relay-profile)# !
RP/0/RP0/CPU0:ios(config-dhcpv4-relay-profile)# interface GigabitEthernet0/0/0/2 relay
profile r1
RP/0/RP0/CPU0:ios(config-dhcpv4)# commit
```

Running Configuration

```
RP/0/RP0/CPU0:ios# show running-config dhcp ipv4
Tue Aug 29 07:30:50.677 UTC
dhcp ipv4
  profile r1 relay
    helper-address vrf default 198.51.100.1 giaddr 198.51.100.3
  !
  interface GigabitEthernet0/0/0/2 relay profile r1
  !
```

DHCP Client

The Dynamic Host Configuration Protocol (DHCP) client functionality enables the router interfaces to dynamically acquire the IPv4 or DHCPv4 or DHCPv6 server, and forwards the responses back to the correct Layer 2 address so that the correct device gets the correct configuration information.

DHCP has the ability to allocate IP addresses only for a configurable period of time, called the lease period. If the client is required to retain this IP address for a longer period beyond the lease period, the lease period must be renewed before the IP address expires. The client renews the lease based on configuration that was sent from the server. The client unicasts a REQUEST message using the IP address of the server. When a server receives the REQUEST message and responds with an ACK message. The lease period of the client is extended by the lease time configured in the ACK message.

Enabling DHCP Client on an Interface

You can enable both the DHCPv4 and DHCPv6 clients at an interface level. The DHCP component receives a notification when DHCPv4 or DHCPv6 is enabled or disabled on an interface.

```
RP/0/RP0/CPU0:ios# configure
Tue Aug 29 09:26:12.468 UTC
RP/0/RP0/CPU0:ios(config)# int mgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:ios(config-if)# ipv4 address dhcp
RP/0/RP0/CPU0:ios(config-if)# commit
Tue Aug 29 09:26:21.715 UTC
RP/0/RP0/CPU0:ios(config-if)# exit
RP/0/RP0/CPU0:ios(config)# int mgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:ios(config-if)# ipv6 address dhcp
```

```

dhcp dhcp-client-options
RP/0/RP0/CPU0:ios(config-if)# ipv6 address dhcp
RP/0/RP0/CPU0:ios(config-if)# commit
Tue Aug 29 09:26:50.159 UTC

```

Boot Using iPXE

iPXE is a pre-boot execution environment that is included in the network card of the management interfaces and works at the system firmware (UEFI) level of the chassis. iPXE is used to reimage the system, and boot the chassis in case of boot failure or in the absence of a valid bootable partition. iPXE downloads the ISO image, proceeds with the installation of the image, and finally bootstraps inside the new installation.



Note The time taken for iPXE to download the ISO image depends on the network speed. Ensure that the network speed is sufficient to complete the image download in less than 10 minutes. The chassis reloads if the image is not downloaded by 10 minutes.

iPXE acts as a bootloader 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. You must define iPXE in the DHCP server configuration file.



Note To initiate the iPXE boot process, perform one of the following methods:

- Use the **reload bootmedia network location all** command. This method is the preferred method.
- Power cycle the NCS 1010 chassis and start the iPXE boot process in the BIOS interface.

Setup DHCP Server

A DHCP server must be configured for IPv4, IPv6, or both communication protocols.



Note For DHCPv6, a routing advertisement (RA) message must be sent to all nodes in the network that indicates which method is to be used to obtain the IPv6 address. Configure Router-advertise-daemon (radvd, install using `yum install radvd`) to allow the client to send the DHCP request. For example:

```

interface eth3
{
    AdvSendAdvert on;
    MinRtrAdvInterval 60;
    MaxRtrAdvInterval 180;
    AdvManagedFlag on;
    AdvOtherConfigFlag on;
    prefix 2001:1851:c622:1::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr off;
    };
};

```

To setup a DHCP server:

1. Create the `dhcpd.conf` file (for IPv4, IPv6 or both communication protocols), `dhcpv6.conf` file (for IPv6) or both in the `/etc/` 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 chassis.
2. Test the server once the DHCP server is running:

For example, for ipv4:

- a. Use MAC address of the chassis:

```
host ncs1010
{
hardware ethernet ab:cd:ef:01:23:45;
fixed-address <ip address>;
filename "http://<httpserver-address>/<path-to-image>/ncs1010-mini-x.iso";
}
```

Ensure that the above configuration is successful.

- b. Use serial number of the chassis:

```
host demo {
option dhcp-client-identifier "<chassis-serial-number>";
filename "http://<IP-address>/<hardware-platform>-mini-x.iso";
fixed-address <IP-address>;
}
```

The serial number of the chassis is derived from the BIOS and is used as an identifier.

Example

```
host 10.89.205.202 {
hardware ethernet 40:55:39:56:0c:e8;
option dhcp-client-identifier "<FCB2437B066>";
if exists user-class and option user-class = "iPXE" {
filename "http://10.89.205.127/box1/ncs1010-x64.iso";
} else {
filename "http://10.89.205.127/box1/StartupConfig.cfg";
}
fixed-address 10.89.205.202;
}
```

Boot Using iPXE

Before you use the iPXE boot, ensure that:

- DHCP server is set and is running.
- Management port of the NCS 1010 chassis is in *UP* state.

Use anyone of the following methods to invoke the iPXE boot process:

- via CLI terminal:

Run the following command to invoke the iPXE boot process to reimage the chassis:

```
reload bootmedia network location all
```

Example:

```
RP/0/RP0/CPU0:ios# reload bootmedia network location all
Wed Jul  6 15:11:33.791 UTC
Reload hardware module ? [confirm]
```

The following example shows the output of the command:

Preparing system for backup. This may take a few minutes especially for large configurations.

```
Status report: node0_RP0_CPU0: BACKUP INPROGRESS
RP/0/RP0/CPU0:PlD_DT# Status report: node0_RP0_CPU0: BACKUP HAS COMPLETED SUCCESSFULLY
```

```
[Done]
[FAILED] Failed unmounting /mnt/fuse/parser_server.
[ OK ] Unmounted /mnt/fuse/ftp.
[ OK ] Unmounted /mnt/fuse/nvgen_server.
[ OK ] Unmounted /boot/efi.
[ OK ] Unmounted /selinux.
```

```
.
```

Output Snipped

```
.
```

```
.
```

```
..          *** Sirius ***
```

```
System Initializing..
```

```
..
```

```
ERROR: Class:0; Subclass:10000; Operation: 1004
```

```
Shelf Assembly Reset
```

```
Shelf Assembly Reset for P1
```

```
..          *** Sirius ***
```

```
System Initializing..
```

```
..
```

```
ERROR: Class:0; Subclass:10000; Operation: 1004
```

```
.
```

```
.
```

Output Snipped

```
.
```

```
.
```

NCS1010, Initializing Devices

Booting from Primary Flash

Aldrin: Programmed MI 10

```
.
```

```
.
```

Output Snipped

```
.
```

```
.
```

Version 2.19.1266. Copyright (C) 2022 American Megatrends, Inc.

BIOS Date: 05/20/2022 10:47:39 Ver: 0ACHIO410

Press or <ESC> to enter setup.

TAM Chipguard Validate Observed DB Error: 0x48

WARNING!!! TAM: Empty Chip DB

Software Boot OK, Validated

iPXE initialising devices...ok

```

iPXE 1.0.0+ (c2215) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP TFTP VLAN EFI ISO9660 ISO9660_grub Menu
Trying net0-2051,net0-2052 and net0-2053...
net0-2051: 68:9e:0b:b8:71:1e using NII on NII-PCI06:00.0 (open)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Unknown (http://ipxe.org/1a086194)]
Configuring (net0-2051 68:9e:0b:b8:71:1e)..... Error 0x040ee186
(http://ipxe.org/040ee186)
net0-2052: 68:9e:0b:b8:71:1f using NII on NII-PCI06:00.0 (open)
  [Link:up, TX:0 TXE:0 RX:18 RXE:14]
  [RXE: 8 x "Operation not supported (http://ipxe.org/3c086083)"]
  [RXE: 3 x "Error 0x440e6083 (http://ipxe.org/440e6083)"]
  [RXE: 3 x "The socket is not connected (http://ipxe.org/380f6093)"]
Configuring (net0-2052 68:9e:0b:b8:71:1f)..... ok
net0: fe80::6a9e:bff:feb8:711e/64
net1: fe80::6a9e:bff:feb8:7121/64 (inaccessible)
net2: fe80::6a9e:bff:feb8:7122/64 (inaccessible)
net3: fe80::6a9e:bff:feb8:7123/64 (inaccessible)
net0-2051: fe80::6a9e:bff:feb8:711e/64
net0-2051: 2001:420:5446:2014::281:0/119 gw fe80::676:b0ff:fed8:c100 (no address)
net0-2051: 2002:420:54ff:93:6a9e:bff:feb8:711e/64 gw fe80::fa4f:57ff:fe72:a640
net0-2052: 10.4.33.44/255.255.0.0 gw 10.4.33.1
net0-2052: fe80::6a9e:bff:feb8:711e/64
net0-2053: fe80::6a9e:bff:feb8:711e/64
Filename: http://10.4.33.51/P1D_DT_05/ncs1010-x64.iso
http://10.4.33.51/P1D_DT_05/ncs1010-x64.iso... ok
.
.
Output Snipped
.
.
User Access Verification

Username: cisco
Password:

```

- via BIOS interface:
 1. Reboot NCS 1010 using power cycle or console.
 2. Press **Esc** to enter BIOS.
 3. Select the **Save & Exit** tab of BIOS.
 4. Choose **UEFI: iPXE Network Boot**.

The following example shows the output of the command:

```

Preparing system for backup. This may take a few minutes especially for large
configurations.
  Status report: node0_RP0_CPU0: BACKUP INPROGRESS
RP/0/RP0/CPU0:P1D_DT#  Status report: node0_RP0_CPU0: BACKUP HAS COMPLETED
SUCCESSFULLY
[Done]
[FAILED] Failed unmounting /mnt/fuse/parser_server.
[ OK ] Unmounted /mnt/fuse/ftp.
[ OK ] Unmounted /mnt/fuse/nvgen_server.
[ OK ] Unmounted /boot/efi.
[ OK ] Unmounted /selinux.
.
.
Output Snipped
.
.

```

```

..          *** Sirius ***
System Initializing..
..

ERROR: Class:0; Subclass:10000; Operation: 1004

Shelf Assembly Reset
Shelf Assembly Reset for P1

..          *** Sirius ***
System Initializing..
..

ERROR: Class:0; Subclass:10000; Operation: 1004
.
.
Output Snipped
.
.

NCS1010, Initializing Devices

Booting from Primary Flash
Aldrin: Programmed MI 10
.
.
Output Snipped
.
.
Version 2.19.1266. Copyright (C) 2022 American Megatrends, Inc.
BIOS Date: 05/20/2022 10:47:39 Ver: 0ACHI0410
Press <DEL> or <ESC> to enter setup.
TAM Chipguard Validate Observed DB Error: 0x48

WARNING!!! TAM: Empty Chip DB

Software Boot OK, Validated

iPXE initialising devices...ok

iPXE 1.0.0+ (c2215) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP TFTP VLAN EFI ISO9660 ISO9660_grub Menu
Trying net0-2051,net0-2052 and net0-2053...
net0-2051: 68:9e:0b:b8:71:1e using NII on NII-PCI06:00.0 (open)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Unknown (http://ipxe.org/1a086194)]
Configuring (net0-2051 68:9e:0b:b8:71:1e)..... Error 0x040ee186
(http://ipxe.org/040ee186)
net0-2052: 68:9e:0b:b8:71:1f using NII on NII-PCI06:00.0 (open)
  [Link:up, TX:0 TXE:0 RX:18 RXE:14]
  [RXE: 8 x "Operation not supported (http://ipxe.org/3c086083)"]
  [RXE: 3 x "Error 0x440e6083 (http://ipxe.org/440e6083)"]
  [RXE: 3 x "The socket is not connected (http://ipxe.org/380f6093)"]
Configuring (net0-2052 68:9e:0b:b8:71:1f)..... ok
net0: fe80::6a9e:bff:feb8:711e/64
net1: fe80::6a9e:bff:feb8:7121/64 (inaccessible)
net2: fe80::6a9e:bff:feb8:7122/64 (inaccessible)
net3: fe80::6a9e:bff:feb8:7123/64 (inaccessible)
net0-2051: fe80::6a9e:bff:feb8:711e/64
net0-2051: 2001:420:5446:2014::281:0/119 gw fe80::676:b0ff:fed8:c100 (no address)
net0-2051: 2002:420:54ff:93:6a9e:bff:feb8:711e/64 gw fe80::fa4f:57ff:fe72:a640

```



```

net0-2052: 10.4.33.44/255.255.0.0 gw 10.4.33.1
net0-2052: fe80::6a9e:bff:feb8:711e/64
net0-2053: fe80::6a9e:bff:feb8:711e/64
Filename: http://10.4.33.51/P1D_DT_05/ncs1010-x64.iso
http://10.4.33.51/P1D_DT_05/ncs1010-x64.iso... ok
.
.
Output Snipped
.
.
User Access Verification

Username: cisco
Password:

```

Boot Using Zero Touch Provisioning

ZTP allows you to provision the network device with day 0 configurations and supports both management ports and data ports.

ZTP provides multiple options, such as:

- Automatically apply specific configuration in a large-scale environment.
- Download and install specific IOS XR image.
- Install specific application package or third party applications automatically.
- Deploy containers without manual intervention.
- Upgrade or downgrade software versions effortlessly on thousands of network devices at a time

Benefits of Using ZTP

ZTP helps you manage large-scale service providers infrastructures effortlessly. Following are the added benefits of using ZTP:

- ZTP helps you to remotely provision a router anywhere in the network. Thus eliminates the need to send an expert to deploy network devices and reduces IT cost.
- Automated provisioning using ZTP can remove delay and increase accuracy and thus is cost-effective and provides better customer experience.

By automating repeated tasks, ZTP allows network administrators to concentrate on more important stuff.

- ZTP process helps you to quickly restore service. Rather than troubleshooting an issue by hand, you can reset a system to well-known working status.

Prerequisites:

ZTP does not execute, if a username is already configured in the system.

ZTP is initiated in one of the following ways:

• Automated Fresh Boot:

Fresh Boot: When you boot the device, the ZTP process initiates automatically if the device does not have a prior configuration. During the process, the router receives the details of the configuration file

from the DHCP server. Use this method for devices that has no pre-loaded configuration. See [Fresh Boot Using DHCP, on page 20](#).

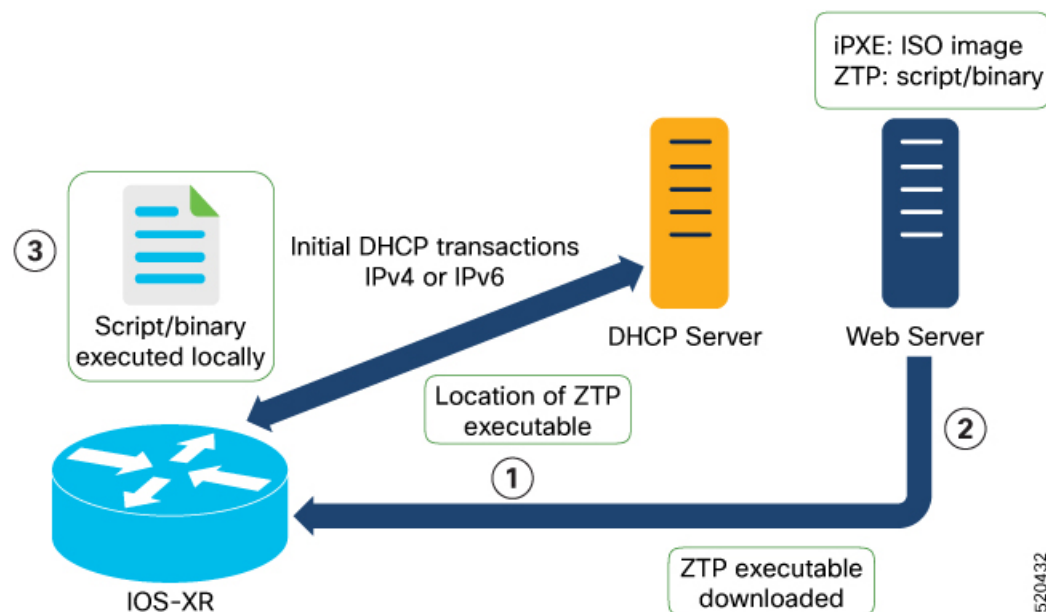
You must define the configuration file or the bootscript that is downloaded from the DHCP server:

- **Configuration File:** The first line of the file must contain **!! IOS XR configuration**", to process the file as a configuration. If you are trying to bring up ten new nodes, you have to define ten configuration files. See [Build your Configuration File, on page 21](#).
- **ZTP Bootscript:** Define the script to be executed on every boot. See [Configure ZTP BootScript, on page 22](#).
- **Manual Invocation using CLI:** Use this method when you want to forcefully initiate ZTP on a fully configured device, using CLI. See [Invoke ZTP Manually through CLI, on page 24](#).

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 7.7.1.22I
!! 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 10.127.60.160 255.255.255.0
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
router static
address-family ipv4 unicast
0.0.0.0/0 10.127.60.1
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.

```

Router#configure
Router(config)#ztp bootscript /disk0:/myscript
Router(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:

```

Router#configure
Router(config)#ztp bootscript preip /disk0:/myscript
Router(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://10.33.0.51/P1B_DT_08/ncs1010-x64.iso";
} else {
filename "http://10.33.0.51/P1B_DT_08/startup.cfg";
}
fixed-address 10.33.0.19;
}

```

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 7.7.1.22I
!! 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 10.127.60.160 255.255.255.0
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 router 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.

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.

Step 1 configure

Example:

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

Enters the configuration mode.

Step 2 commit replace

Example:

```
Fri Apr 29 06:48:46.236 UTC
RP/0/RP0/CPU0:P2B_DT_02(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)#
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.

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 data interface using the command `ztp initiate 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 router
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 10.33.0.63
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:

```

RP/0/RP0/CPU0:R1#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.

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.

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 10.1.1.1 255.0.0.0
```

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 `router static address-family ipv4 unicast 0.0.0.0/0 default-gateway`

Example:

```
RP/0/RP0/CPU0:ios(config)#router 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 [Configure SSH](#) or [Configure 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 Router.

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 1:

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:regen#configure terminal
RP/0/RP0/CPU0:regen(config)#lldp management enable
RP/0/RP0/CPU0:regen(config)#lldp holdtime 30
RP/0/RP0/CPU0:regen(config)#lldp reinit 2
RP/0/RP0/CPU0:regen(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:regen#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:regen#show lldp interface
Thu Nov 7 08:45:22.934 UTC
```

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

```
MgmtEth0/RP0/CPU0/1:
  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/CPU0:M-131#show lldp neighbors
Mon Dec 2 11:01:20.143 CET
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
[DISABLED]         MgmtEth0/RP0/CPU0/0  120        B           gi19
MYS-130            MgmtEth0/RP0/CPU0/1  120        R           MgmtEth0/RP0/CPU0/1
```

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. .

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.

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.
-

Configure NTP Server

Understand NTP

Table 2: Feature History

Feature Name	Release Information	Feature Description
NTP Support	Cisco IOS XR Release 7.11.1	<p>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.</p> <p>Commands added:</p> <ul style="list-style-type: none"> • 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, on page 28](#)

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 4.33.0.51 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:router (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.

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
*~4.33.0.51 10.64.58.50 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
```

```
4.33.0.51 configured, our_master, stratum 2
ref ID 10.64.58.50, 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
filtererror = 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
```

```
4.33.0.51 configured, our_master, stratum 2
ref ID 10.64.58.50, 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
filtererror = 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 4.33.0.51
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.

NTP Troubleshooting Information

For NTP troubleshooting information, see [here](#).



CHAPTER 3

Disaster Recovery

This chapter describes the disaster recovery process and the health check feature.

- [Overview, on page 39](#)
- [CPU Replacement Considerations, on page 39](#)
- [Health Check of Backup ISO Image, on page 39](#)

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.



CHAPTER 4

Perform Preliminary Checks

After successfully logging into the console, you must perform some preliminary checks to verify the default setup. If any setup issue is detected, take corrective action before making further configurations.



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 Status of Hardware Components, on page 41](#)
- [Verify Inventory, on page 42](#)
- [Verify Software Version, on page 43](#)
- [Verify Firmware Version, on page 44](#)
- [Verify Management Interface Status, on page 46](#)
- [Verify Alarms, on page 47](#)
- [Verify Environmental Parameters, on page 48](#)
- [Verify Context, on page 54](#)
- [Verify Core Files, on page 54](#)
- [Verify Memory Information, on page 54](#)

Verify Status of Hardware Components

To verify the status of all the hardware components installed on NCS 1010, perform the following procedure.

Before you begin

Ensure that all the required hardware components are installed on NCS 1010. For installation details, see *Cisco Network Convergence System 1010 Hardware Installation Guide*.

show platform

When you execute this command, the status of Cisco IOS XR is displayed.

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-CNTRLR-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

RP0/RP0/CPU0:ios#show platform
Thu Mar  2 12:35:01.883 IST
Node                Type                State              Config state
-----
0/RP0/CPU0         NCS1010-CNTRLR-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

```

Verify that all the components of NCS 1010 are displayed in output. The state must be in the OPERATIONAL state. The various states are:

- OPERATIONAL—Node is operating normally and is fully functional.
- POWERED_ON—Power is on and the node is booting up.
- FAILED—Node is powered on but has encountered an internal failure.
- PRESENT—Node is in intermediate state in the boot sequence.
- POWERED_OFF—Power is off and the node cannot be accessed.
- IOS XR RUN—Node is running IOS XR.
- OFFLINE—Input power is not connected to the power modules.

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.

show inventory

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

Example:

```

RP0/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-CNTRLR-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 Software Version

NCS 1010 is shipped with the Cisco IOS XR software preinstalled. Verify that the latest version of the software is installed. If a newer version is available, perform a [Upgrade Software, on page 57](#). This software upgrade installs the newer version of the software and provide the latest feature set on NCS 1010.

To verify the version of Cisco IOS XR Software running on NCS 1010, perform the following procedure.

show version

Displays the software version and details such as system uptime.

Example:

```
RP/0/RP0/CPU0:ios#show version
Sat Mar 25 11:38:23.614 IST
Cisco IOS XR Software, Version 7.9.1
Copyright (c) 2013-2023 by Cisco Systems, Inc.
```

Verify Firmware Version

```

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/7.9.1.30I.SIT_IMAGE/ncs1010/ws
Version : 7.9.1
Label : 7.9.1
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

```

Verify the software version to determine whether system upgrade is required. If the upgrade is required, see [Upgrade Software, on page 57](#).

Verify Firmware Version

The firmware version on various hardware components of NCS 1010 must be compatible with the installed Cisco IOS XR release. Incompatibility may cause NCS 1010 to malfunction.

To verify the firmware version, perform the following procedure.

Step 1 show hw-module fpd

Displays the firmware information of various hardware components of NCS 1010.

Example:

```

RP/0/RP0/CPU0:ios#show hw-module fpd
Thu Mar 2 12:35:06.602 IST

```

```

Auto-upgrade:Enabled
Attribute codes: B golden, P protect, S secure, A Anti Theft aware

```

Location	Card type	HWver	FPD device	ATR	Status	FPD Versions		Reload Loc
						Running	Programd	
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	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-PrimMCU		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-PrimMCU		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

Step 2 show fpd package

Displays the FPD image version available with this software release for each hardware component.

Example:

```
RP/0/RP0/CPU0:ios#show fpd package
Thu Mar  2 12:37:58.530 IST
```

```
=====
                                Field Programmable Device Package
                                =====
Card Type           FPD Description           Req   SW   Min Req   Min Req
                    Reinstall Ver       SW Ver   Board Ver
=====
NCS1010-AC-PSU     AP-PriMCU                 NO     1.03   1.03     0.0
                    AP-SecMCU                 NO     2.01   2.01     0.0
-----
NCS1010-CNTRLR-K9  ADMConfig                 NO     2.30   2.30     0.0
                    ADMConfig                 NO     2.30   2.30     0.0
                    ADMConfig                 NO     3.40   3.40     1.0
                    BIOS                       YES    4.20   4.20     0.0
                    BIOS                       YES    4.20   4.20     0.0
                    BIOS-Golden                YES    4.10   4.10     0.0
                    BIOS-Golden                YES    4.10   4.10     0.0
                    CpuFpga                   YES    1.11   1.11     0.0
                    CpuFpga                   YES    1.11   1.11     0.0
                    CpuFpgaGolden             YES    1.01   1.01     0.0
                    CpuFpgaGolden             YES    1.01   1.01     0.0
                    SsdIntelS4510             YES   11.32  11.32    0.0
                    SsdIntelS4510             YES   11.32  11.32    0.0
                    SsdMicron5300             YES    0.01   0.01     0.0
                    SsdMicron5300             YES    0.01   0.01     0.0
                    SsdSmartModular           YES   13.06  13.06    0.0
                    SsdSmartModular           YES   13.06  13.06    0.0
                    TamFw                     YES    6.13   6.13     0.0
                    TamFw                     YES    6.13   6.13     0.0
                    TamFwGolden              YES    6.11   6.11     0.0
                    TamFwGolden              YES    6.11   6.11     0.0
-----
NCS1010-SA         EITU-ADMConfig            NO     1.04   1.04     0.0
                    EITU-ADMConfig            NO     2.10   2.10     1.0
                    EITU-ADMConfig            NO     1.04   1.04     0.0
                    EITU-ADMConfig            NO     2.10   2.10     1.0
                    IoFpga                    NO     1.12   1.12     0.0
                    IoFpga                    NO     1.12   1.12     0.0
                    IoFpgaGolden              NO     1.01   1.01     0.0
                    IoFpgaGolden              NO     1.01   1.01     0.0
                    SsdIntelS4510             YES   11.32  11.32    0.0
                    SsdIntelS4510             YES   11.32  11.32    0.0
                    SsdMicron5300             YES    0.01   0.01     0.0
                    SsdMicron5300             YES    0.01   0.01     0.0
                    SsdSmartModular           YES   13.06  13.06    0.0
                    SsdSmartModular           YES   13.06  13.06    0.0
-----
NCS1K-ILA-2R-C     ILA                       NO     1.12   1.12     0.1
                    ILA                       NO     0.28   0.28    99.1
                    Raman-1                   NO     1.04   1.04     0.1
                    Raman-1                   NO     0.28   0.28    99.1
                    Raman-2                   NO     1.04   1.04     0.1
                    Raman-2                   NO     0.28   0.28    99.1
-----
NCS1K-ILA-C        ILA                       NO     1.12   1.12     0.1
                    ILA                       NO     0.28   0.28    99.1
-----
NCS1K-ILA-L        ILA                       NO     1.00   1.00     0.1
-----
NCS1K-ILA-R-C     ILA                       NO     1.12   1.12     0.1
=====
```

	ILA	NO	0.28	0.28	99.1
	Raman-1	NO	1.04	1.04	0.1
	Raman-1	NO	0.28	0.28	99.1

NCS1K-OLT-C	OLT	NO	1.12	1.12	0.1
	OLT	NO	0.28	0.28	99.1

NCS1K-OLT-L	OLT	NO	1.02	1.02	0.1

NCS1K-OLT-R-C	OLT	NO	1.12	1.12	0.1
	OLT	NO	0.28	0.28	99.1
	Raman-1	NO	1.04	1.04	0.1
	Raman-1	NO	0.28	0.28	99.1

Verify Management Interface Status

To verify the management interface status, perform the following 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 10.33.0.61/16
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
Wed May 25 11:50:02.558 UTC
Interface Type          Total    UP      Down    Admin Down
-----
ALL TYPES                9        5       0       4
-----
IFT_GETHERNET           1        1       0       0
IFT_LOOPBACK            2        2       0       0
IFT_ETHERNET            3        1       0       2
IFT_NULL                1        1       0       0
IFT_PTP_ETHERNET        2        0       0       2
```

Example:

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

Intf Name	Intf State	LineP State	Encap Type	MTU (byte)	BW (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	10.3.3.21	Up	Up	default
Loopback3	10.1.1.2	Up	Up	default
GigabitEthernet0/0/0/0	10.7.1.20	Up	Up	default
MgmtEth0/RP0/CPU0/0	10.4.33.63	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

Verify Alarms

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

```
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 (PM_OUTPUT_DISABLED)	Major	Environ	04/28/2022 06:03:50 UTC	Power Module Output Disabled
0	Major	Environ	04/28/2022 06:03:50 UTC	Power Group redundancy lost
0/PM0 Current State	Major	FPD_Infra	04/28/2022 06:04:08 UTC	One Or More FPDs Need Upgrade Or Not In
0/PM1 Current State	Major	FPD_Infra	04/28/2022 06:04:09 UTC	One Or More FPDs Need Upgrade Or Not In
0/0	Minor	Environ	04/28/2022 06:04:10 UTC	ILAC_CT_1: Invalid sensor read error.
0/0	Minor	Environ	04/28/2022 06:04:10 UTC	ILAC_CT_2: Invalid sensor read error.
0/0	Minor	Environ	04/28/2022 06:04:10 UTC	ILAC_CT_3: Invalid sensor read error.
0/0	Minor	Environ	04/28/2022 06:04:10 UTC	ILAC_CT_4: Invalid sensor read error.
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.

```
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
```

```
=====
```

Location	FRU Type	Fan speed (rpm)		
		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 *O/RP0* location.

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

```
Thu May 26 04:16:39.832 UTC
```

Location	TEMPERATURE Sensor	Value (deg C)	Crit (Lo)	Major (Lo)	Minor (Lo)	Minor (Hi)	Major (Hi)	Crit (Hi)
O/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 *O/O/NXR0* location.

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

```
Thu May 26 04:16:39.832 UTC
```

Location	TEMPERATURE Sensor	Value (deg C)	Crit (Lo)	Major (Lo)	Minor (Lo)	Minor (Hi)	Major (Hi)	Crit (Hi)
O/O/NXR0								
	OLTCT_LT_P0_iEDFA0	24	18	19	20	30	31	32
	OLTCT_LT_P0_iEDFA1	25	18	19	20	30	31	32
	OLTCT_LT_P0_iEDFA2	24	18	19	20	30	31	32
	OLTCT_LT_P2_iEDFA0	25	18	19	20	30	31	32
	OLTCT_LT_P3_iEDFA0	25	18	19	20	30	31	32
	OLTCT_LT_P0_eEDFA0	24	18	19	20	30	31	32
	OLTCT_CT_1	32	-10	-7	-5	75	77	80
	OLTCT_LT_P0_eEDFA1	24	18	19	20	30	31	32
	OLTCT_CT_2	27	-10	-7	-5	70	73	75
	OLTCT_CT_3	30	-10	-7	-5	70	73	75
	OLTCT_CT_4	30	-10	-7	-5	70	73	75
	OLTCT_FT_P0_iEDFA0	60	55	57	58	62	64	65
	OLTCT_FT_P2_iEDFA0	60	55	57	58	62	64	65
	OLTCT_FT_P3_iEDFA0	60	55	57	58	62	64	65
	OLTCT_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 Module	Supply Type	-----Input----	-----Output----	Status		
		Volts	Amps	Volts	Amps	
O/PM0	NCS1010-AC-PSU	228.5	0.5	12.1	5.6	OK

Verify Environmental Parameters

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-CNTRLR-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
    
```

Verify Environmental Parameters

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 Sensor	Value (mV)	Crit (Lo)	Minor (Lo)	Minor (Hi)	Crit (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 Sensor	Value (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

```

```

=====
Location      FRU Type                               Fan speed (rpm)
-----
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-CNTRLR-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.

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.

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.

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



CHAPTER 5

Upgrade Software and FPD

This chapter describes the procedures to upgrade software and FPDs.



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.

- [Upgrade Software, on page 57](#)
- [Software Upgrade and Downgrade Matrix, on page 59](#)
- [Install Packages and RPMs, on page 59](#)
- [NCS 1010 FPD, on page 63](#)
- [Release 7.10.1 Caveats, on page 71](#)

Upgrade Software

Upgrading the software is the process of installing a new version of the Cisco IOS XR operating system on NCS 1010. NCS 1010 is preinstalled with the Cisco IOS XR image. However, you can install a new version to keep features up to date. You can perform the software upgrade operation using an ISO image from the XR mode.

Before you begin

- [Configure Management Interface](#)
- 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.

Step 1 Execute one of these commands:

- **install replace /harddisk:/iso-image-name**
- **install package replace** *<ftp or http or https protocol>/package_path/ filename1 filename2 ...*

Note The **install package replace** command upgrades the ISO image but doesn't reload the RP automatically. But the **install replace** command upgrades the ISO image and reloads the RP.

Example:

```
RP/0/RP0/CPU0:ios#install replace /harddisk:/ncs1010-x64.iso
Mon Jul  4 10:15:07.697 UTC
Once the packaging dependencies have been determined, the install operation may have to reload the
system.
If you want to control the timing of system reload, you must not continue, but use the 'install
package replace' command instead, followed by 'install apply'.
Continue? [yes/no]:[yes] yes
Install replace operation 1.1 has started
Install operation will continue in the background
.....
.....
ios con0/RP0/CPU0 is now available
```

Installs the new ISO image from the harddisk or from the network server. The install operation takes between 20–40 minutes to complete.

Note Boot time FPD upgrade happens before XR boot. All the FPDs belonging to the RP location are upgraded during the boot time FPD upgrade.

Note Automatic FPD upgrade is enabled by default. When the automatic FPD upgrade is enabled, the install operation also upgrades the FPDs (except the Golden FPDs) that need to be upgraded.

Step 2 **show install request****Example:**

```
RP/0/RP0/CPU0:ios#show install request
Mon May  9 15:16:27.486 UTC
User request: install replace /harddisk:/ncs1010-x64.iso
Operation ID: 1.1
State:          In progress since 2022-05-09 15:13:08 UTC
Current activity:  Package add or other package operation
Next activity:    Apply
Time started:    2022-05-09 15:14:34 UTC
Timeout in:      38m 6s
Locations responded: 0/1
Location          Packaging operation stage Notification Phase Clients responded
-----
0/RP0/CPU0        Package operations          None in progress          N/A
```

Displays the current status of the install operation.

When the install operation completes successfully, the device automatically reloads.

Note In case of the **install package replace** command, you'll be prompted to enter the next command (**install apply reload** command).

Step 3 **install commit****Example:**

```
RP/0/RP0/CPU0:ios#install commit
Mon May  9 15:24:28.581 UTC
Install commit operation 1 has started
Install operation will continue in the background
```

Commits the new ISO image.

Step 4 **show install committed****Example:**


```
RP/0/RP0/CPU0:ios#show install committed
Mon May  9 15:24:55.672 UTC
Software Hash: 9dfe3b29058bd85eccc3910fb6ea66bf7bf9ccaa9e7ef38c8e3499ab1d0e91f8
Package                               Version
-----
xr-aaa                                 7.9.1
xr-acl                                 7.9.1
xr-apphosting                          7.9.1
xr-appmgr                               7.9.1
xr-bcdl                                 7.9.1
xr-bfd                                 7.9.1
```

Displays the list of committed packages.

Software Upgrade and Downgrade Matrix

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

Upgrade Path		Downgrade Path	
Source Release	Destination Release	Source Release	Destination Release
R7.7.1, R7.9.1, R7.10.1	R7.11.1	R7.11.1	R7.10.1, R7.9.1, R7.7.1

Install Packages and RPMs

Complete this task to install additional packages or rpm files. The rpm files that need to be installed must be placed in a folder.



Note This task can be used to install SMUs as well.

Before you begin

- Configure and connect to the management interface. You can access the installable file through the management interface. For details about configuring the management interface, see [Configure Management Interface, on page 28](#).
- Copy the package or rpm to be installed either on the NCS 1010 hard disk or on a network server to which NCS 1010 has access.

Step 1 `install package add source /harddisk:/ iso-image-name or rpm-folder-name`

Example:

```
RP/0/RP0/CPU0:ios#install package add source /harddisk:/rpm/
Mon Jul  4 11:37:31.526 UTC
Install add operation 2.1.1 has started
Install operation will continue in the background
```

Ensure to add the respective packages or rpm files as appropriate. This operation may take time depending on the size of the files that are added. The operation takes place in an asynchronous mode. The **install package add source** command runs in the background, and the EXEC prompt is returned.

Step 2 show install request

Example:

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

Mon Jul  4 11:44:48.411 UTC

User request: install package add source file:///harddisk:/rpm/
Operation ID: 2.1.1
State:        Success since 2022-07-04 11:38:57 UTC

Current activity:  Await user input
Time started:     2022-07-04 11:38:57 UTC

The following actions are available:
install package add
install package remove
install package upgrade
install package downgrade
install package abort latest
install package abort all-since-apply
install apply restart
install apply reload

Least impactful apply method: install apply restart
```

Displays the current status of the install operation.

Step 3 install apply reload

Example:

```
RP/0/RP0/CPU0:ios#install apply reload

Mon Jul  4 11:45:18.434 UTC
Install apply operation 2.1 has started
Install operation will continue in the background

Enables NCS 1010 to reload.
```

Step 4 show install request

Example:

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

Mon Jul  4 11:47:32.221 UTC

User request: install apply reload
Operation ID: 2.1
State:        Success since 2022-07-04 11:46:03 UTC

Current activity:  Await user input
Time started:     2022-07-04 11:46:03 UTC

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
install commit

```

Displays the current status of the install operation.

Step 5 **install commit**

Example:

```

RP/0/RP0/CPU0:ios#install commit
Mon Jul 4 11:48:47.745 UTC
Install commit operation 2 has started
Install operation will continue in the background

```

Commits the package or rpm files.

Step 6 **show install request**

Example:

```

RP/0/RP0/CPU0:ios#show install request

User request: install commit
Operation ID: 2
State:        In progress since 2022-07-04 11:48:48 UTC

Current activity:  Commit transaction
Next activity:    Transaction complete
Time started:     2022-07-04 11:48:48 UTC

```

No per-location information.

Displays the current status of the install operation. The above output indicates that the install operation is in progress.

Example:

```

RP/0/RP0/CPU0:ios#show install request

User request: install commit
Operation ID: 2
State:        Success since 2022-07-04 11:50:32 UTC

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

```

Displays the current status of the install operation. The above output indicates that the install operation is complete.

Step 7 **show install active summary**

Example:

```

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

```

```

Mon Jul  4 11:52:24.823 UTC
Active Packages:   XR: 145   All: 1265
Label:            7.9.1
Software Hash:    3ce63ce432d50358d7a0d654ec61e4377abccf265013132e310b4d34a7259b90

```

```

Optional Packages                               Version
-----
xr-bgp                                           7.9.1
xr-ipsla                                         7.9.1
xr-is-is                                         7.9.1
xr-lldp                                           7.9.1
xr-mppls-oam                                     7.9.1
xr-netsim                                        7.9.1
xr-olc                                           7.9.1
xr-ospf                                          7.9.1
xr-perfmgmt                                     7.9.1
xr-telnet                                       7.9.1
xr-track                                        7.9.1

```

Displays the list of active packages and rpm files.

Step 8 show install committed summary

Example:

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

```

Mon Jul  4 11:54:04.178 UTC
Committed Packages: XR: 145   All: 1265
Label:            7.9.1
Software Hash:    3ce63ce432d50358d7a0d654ec61e4377abccf265013132e310b4d34a7259b90

```

```

Optional Packages                               Version
-----
xr-bgp                                           7.9.1
xr-ipsla                                         7.9.1
xr-is-is                                         7.9.1
xr-lldp                                           7.9.1
xr-mppls-oam                                     7.9.1
xr-netsim                                        7.9.1
xr-olc                                           7.9.1
xr-ospf                                          7.9.1
xr-perfmgmt                                     7.9.1
xr-telnet                                       7.9.1
xr-track                                        7.9.1

```

Displays the list of committed packages and rpm files.

Related Commands

The following commands can be used to track the status of the install operation.

Related Commands	Purpose
show install active	Displays the list of active packages.
show install committed	Displays the list of committed packages.
show install log	Displays the log information for the install operation. This information is used for troubleshooting in case of installation failure.

Related Commands	Purpose
show install package	Displays the details of the packages that are added to the repository. Use this command to identify individual components of a package.
show install request	Displays the current status of the install operation.
show install which	Displays the package information on an installed file.

NCS 1010 FPD

Table 3: 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.

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.

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

Table 4: NCS 1010 FPDs

Location	FPDs
RP	<ul style="list-style-type: none"> • ADMConfig • CpuFpga • CpuFpgaGolden • BIOS • BIOS-Golden • SsdIntelS4510 • SsdMicron5300 • SsdSmartModular • TamFw • TamFwGolden
PM0 and PM1	<ul style="list-style-type: none"> • AP-PrimMCU • AP-SecMCU
LC	<ul style="list-style-type: none"> • ILA • OLT • Raman-1 • Raman-2
Rack	<ul style="list-style-type: none"> • IoFpga • IoFpgaGolden • EITU-ADMConfig • SsdIntelS4510 • SsdMicron5300 • SsdSmartModular
Breakout module	<ul style="list-style-type: none"> • BRK-8 • BRK-24
Multiplexer and demultiplexer modules	<ul style="list-style-type: none"> • 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.

Retrieve FPD Information

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
```

The following output shows the types of FPDs for each module.

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

```
Auto-upgrade:Enabled
```

```
Attribute codes: B golden, P protect, S secure, A Anti Theft aware
```

Location Reload Loc	Card type	HWver	FPD device	ATR Status	FPD Versions	
					Running	Programd
0/RP0/CPU0 NOT REQ	NCS1010-CNTRLR-K9	1.11	ADMConfig	CURRENT	3.40	3.40
0/RP0/CPU0 0/RP0	NCS1010-CNTRLR-K9	1.11	BIOS	S CURRENT	4.20	4.20
0/RP0/CPU0 0/RP0	NCS1010-CNTRLR-K9	1.11	BIOS-Golden	BS CURRENT		4.10
0/RP0/CPU0 0/RP0	NCS1010-CNTRLR-K9	1.11	CpuFpga	S CURRENT	1.11	1.11
0/RP0/CPU0 0/RP0	NCS1010-CNTRLR-K9	1.11	CpuFpgaGolden	BS CURRENT		1.01
0/RP0/CPU0 0/RP0	NCS1010-CNTRLR-K9	1.11	SsdIntelS4510	S CURRENT	11.32	11.32
0/RP0/CPU0 0/RP0	NCS1010-CNTRLR-K9	1.11	TamFw	S CURRENT	6.13	6.13
0/RP0/CPU0 0/RP0	NCS1010-CNTRLR-K9	1.11	TamFwGolden	BS CURRENT		6.11
0/PM0 NOT REQ	NCS1010-AC-PSU	0.0	AP-PrimMCU	CURRENT	1.03	1.03
0/PM0 NOT REQ	NCS1010-AC-PSU	0.0	AP-SecMCU	CURRENT	2.01	2.01
0/PM1 NOT REQ	NCS1010-AC-PSU	0.0	AP-PrimMCU	CURRENT	1.03	1.03
0/PM1 NOT REQ	NCS1010-AC-PSU	0.0	AP-SecMCU	CURRENT	2.01	2.01
0/0/NXR0 NOT REQ	NCS1K-OLT-L	1.0	OLT	S CURRENT	1.02	1.02
0/Rack NOT REQ	NCS1010-SA	2.1	EITU-ADMConfig	CURRENT	2.10	2.10
0/Rack NOT REQ	NCS1010-SA	2.1	IoFpga	S CURRENT	1.12	1.12
0/Rack NOT REQ	NCS1010-SA	2.1	IoFpgaGolden	BS CURRENT		1.01
0/Rack 0/Rack	NCS1010-SA	2.1	SsdIntelS4510	S CURRENT	11.32	11.32

The following output highlights the types of FPDs for the new controller card, new OLT line card, breakout module, and multiplexer/demultiplexer module.

```
Fri Feb 17 11:43:28.878 UTC
```

```
Auto-upgrade:Enabled
```

Attribute codes: B golden, P protect, S secure, A Anti Theft aware

Location Reload Loc	Card type	HWver	FPD device	ATR	Status	FPD Versions	
						Running	Programd
0/RP0/CPU0 NOT REQ	NCS1010-CTLR-B-K9	1.0	ADMConfig		CURRENT	2.30	2.30
0/RP0/CPU0 0/RP0	NCS1010-CTLR-B-K9	1.0	BIOS	S	CURRENT	4.40	4.40
0/RP0/CPU0 0/RP0	NCS1010-CTLR-B-K9	1.0	BIOS-Golden	BS	CURRENT		4.40
0/RP0/CPU0 0/RP0	NCS1010-CTLR-B-K9	1.0	CpuFpga	S	CURRENT	1.11	1.11
0/RP0/CPU0 0/RP0	NCS1010-CTLR-B-K9	1.0	CpuFpgaGolden	BS	CURRENT		1.01
0/RP0/CPU0 0/RP0	NCS1010-CTLR-B-K9	1.0	SsdIntels4510	S	CURRENT	11.32	11.32
0/RP0/CPU0 0/RP0	NCS1010-CTLR-B-K9	1.0	TamFw	S	CURRENT	6.13	6.13
0/RP0/CPU0 0/RP0	NCS1010-CTLR-B-K9	1.0	TamFwGolden	BS	CURRENT		6.11
0/PM0 NOT REQ	NCS1010-AC-PSU	0.0	AP-PrimMCU		CURRENT	1.03	1.03
0/PM0 NOT REQ	NCS1010-AC-PSU	0.0	AP-SecMCU		CURRENT	2.01	2.01
0/PM1 NOT REQ	NCS1010-AC-PSU	0.0	AP-PrimMCU		CURRENT	1.03	1.03
0/PM1 NOT REQ	NCS1010-AC-PSU	0.0	AP-SecMCU		NEED UPGD	1.06	1.06
0/0/NXR0 NOT REQ	NCS1K-E-OLT-R-C	1.0	OLT	S	CURRENT	1.16	1.16
0/0/NXR0 NOT REQ	NCS1K-E-OLT-R-C	1.0	Raman-1	S	CURRENT	1.04	1.04
0/Rack NOT REQ	NCS1010-SA	0.1	EITU-ADMConfig		CURRENT	1.04	1.04
0/Rack NOT REQ	NCS1010-SA	0.1	IoFpga	S	CURRENT		1.12
0/Rack NOT REQ	NCS1010-SA	0.1	IoFpgaGolden	BS	NEED UPGD	1.12	0.08
0/Rack 0/Rack	NCS1010-SA	0.1	SsdIntels4510	S	CURRENT	11.32	11.32
0/1 NOT REQ	NCS1K-MD-32E-C	0.1	MD-32-NEO	S	CURRENT	2.02	2.02
0/2 NOT REQ	NCS1K-MD-32O-C	10.2	MD-32-ACC	S	CURRENT	2.18	2.18
0/3/0 NOT REQ	NCS1K-BRK-8	1.0	BRK-8	S	CURRENT	2.08	2.08
0/3/3 NOT REQ	NCS1K-BRK-24	1.0	BRK-24	S	CURRENT	2.08	2.08

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

Table 5: 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.

Field	Description
HWver	Hardware version where the FPD resides.
FPD device	Name of the FPD.
ATR	Attribute codes. The possible values are: <ul style="list-style-type: none"> • B - Golden Image • S - Secure Image • P - Protect Image <p>The attribute code of the primary FPDs is S and the Golden FPDs is BS.</p>
Status	Status of the FPD. See Table 6: Description of FPD Status Values in show hw-module fpd Command , on page 67.
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 6: 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.
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, <ul style="list-style-type: none"> • FPD version is up to date and compatible. • FPD image is protected.

FPD Status	Description
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.

Verify if an FPD Upgrade is Required

Step 1 Use the **show hw-module fpd** command to check whether all the FPDs are in the Current state.

If the status of any FPD is **NEED UPGD**, then the upgrade is required for that FPD.

Step 2 Use the **show fpd package** command to determine the FPDs that are supported with the current software release and the minimum hardware requirements for each FPD.

```
RP/0/RP0/CPU0:ios#show fpd package
```

The following output highlights the FPD packages for the breakout and multiplexer/demultiplexer modules.

```
Thu Mar 2 12:37:58.530 IST
```

```

=====
                                Field Programmable Device Package
=====
Card Type          FPD Description          Req   SW   Min Req  Min Req
=====  =====  =====  =====  =====  =====
NCS1010-AC-PSU    AP-PrimCU                NO    1.03  1.03     0.0
                  AP-SecMCU                NO    2.01  2.01     0.0
-----
NCS1010-CNTRLR-K9  ADMConfig                NO    2.30  2.30     0.0
                  ADMConfig                NO    2.30  2.30     0.0
                  ADMConfig                NO    3.40  3.40     1.0
                  BIOS                     YES    4.20  4.20     0.0
                  BIOS                     YES    4.20  4.20     0.0
                  BIOS-Golden              YES    4.10  4.10     0.0
                  BIOS-Golden              YES    4.10  4.10     0.0
                  CpuFpga                  YES    1.11  1.11     0.0
                  CpuFpga                  YES    1.11  1.11     0.0
                  CpuFpgaGolden            YES    1.01  1.01     0.0
                  CpuFpgaGolden            YES    1.01  1.01     0.0
                  SsdIntelS4510            YES   11.32  11.32    0.0
                  SsdIntelS4510            YES   11.32  11.32    0.0
                  SsdMicron5300            YES    0.01  0.01     0.0
                  SsdMicron5300            YES    0.01  0.01     0.0
                  SsdSmartModular          YES   13.06  13.06    0.0
                  SsdSmartModular          YES   13.06  13.06    0.0

```

	TamFw	YES	6.13	6.13	0.0
	TamFw	YES	6.13	6.13	0.0
	TamFwGolden	YES	6.11	6.11	0.0
	TamFwGolden	YES	6.11	6.11	0.0

NCS1010-SA	EITU-ADMConfig	NO	1.04	1.04	0.0
	EITU-ADMConfig	NO	2.10	2.10	1.0
	EITU-ADMConfig	NO	1.04	1.04	0.0
	EITU-ADMConfig	NO	2.10	2.10	1.0
	IoFpga	NO	1.12	1.12	0.0
	IoFpga	NO	1.12	1.12	0.0
	IoFpgaGolden	NO	1.01	1.01	0.0
	IoFpgaGolden	NO	1.01	1.01	0.0
	SsdIntelS4510	YES	11.32	11.32	0.0
	SsdIntelS4510	YES	11.32	11.32	0.0
	SsdMicron5300	YES	0.01	0.01	0.0
	SsdMicron5300	YES	0.01	0.01	0.0
	SsdSmartModular	YES	13.06	13.06	0.0
	SsdSmartModular	YES	13.06	13.06	0.0

NCS1K-ILA-2R-C	ILA	NO	1.12	1.12	0.1
	ILA	NO	0.28	0.28	99.1
	Raman-1	NO	1.04	1.04	0.1
	Raman-1	NO	0.28	0.28	99.1
	Raman-2	NO	1.04	1.04	0.1
	Raman-2	NO	0.28	0.28	99.1

NCS1K-ILA-C	ILA	NO	1.12	1.12	0.1
	ILA	NO	0.28	0.28	99.1

NCS1K-ILA-L	ILA	NO	1.00	1.00	0.1

NCS1K-ILA-R-C	ILA	NO	1.12	1.12	0.1
	ILA	NO	0.28	0.28	99.1
	Raman-1	NO	1.04	1.04	0.1
	Raman-1	NO	0.28	0.28	99.1

NCS1K-OLT-C	OLT	NO	1.12	1.12	0.1
	OLT	NO	0.28	0.28	99.1

NCS1K-OLT-L	OLT	NO	1.02	1.02	0.1

NCS1K-OLT-R-C	OLT	NO	1.12	1.12	0.1
	OLT	NO	0.28	0.28	99.1
	Raman-1	NO	1.04	1.04	0.1
	Raman-1	NO	0.28	0.28	99.1

NCS1K-BRK-24	BRK-24	NO	2.08	2.08	0.0

NCS1K-BRK-8	BRK-8	NO	2.08	2.08	0.0

NCS1K-MD-32E-C	MD-32-ACC	NO	2.18	2.18	0.0
	MD-32-NEO	NO	2.02	2.02	0.0

NCS1K-MD-32O-C	MD-32-ACC	NO	2.18	2.18	0.0
	MD-32-NEO	NO	2.02	2.02	0.0

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

Table 7: 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.
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.

FPD can be upgraded using two methods:

- [Upgrade FPDs Manually](#)
- [Upgrade FPDs Automatically](#)

Upgrade FPDs Manually

Use the following procedure to upgrade the FPDs manually.



Note The Golden FPDs cannot be upgraded using the CLI.

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.

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.

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
```

Release 7.10.1 Caveats

The following upgrade caveats are applicable for the Cisco NCS 1010 platform for Release 7.10.1 and later:

Table 8: Upgrade Caveats for Cisco NCS 1010 Platform

From	To	Bridge SMUs Required	Caveats
7.7.1	7.10.1 and later	None	1*
7.9.1	7.10.1 and later	None	1*

1* : You can't roll back using the **install rollback** command.

The following downgrade caveats are applicable for the Cisco NCS 1010 platform for Release 7.10.1 and later:

Table 9: Downgrade Caveats for Cisco NCS 1010 Platform

From	To	Bridge SMUs Required	Caveats
7.10.1 and later	7.7.1	Yes	***, A*, B*, C*, D*
7.10.1 and later	7.9.1	Yes	***, C*, D*

- You don't need to run the **install commit** command after a downgrade operation because the operation is automatically committed.
- You can't roll back after a downgrade. To revert to the previous IOS XR previous version, replace or reimage to the relevant ISO.
- IOS XR configuration history is lost after a downgrade, but the NCS 1010 platform preserves the latest configuration.
- Install history from the last transaction is preserved after a downgrade operation.
- Downtime takes a longer time as the operation is performed through reimage.
- You can't downgrade using the **install package replace** command. Instead, use the **install replace** command to downgrade.
- Ensure that you reinstall third-party application once you complete the downgrade.
- PXE recovery is required if the image downgrading isn't bootable.
- You must re-install the *Type 6 masterkey* and reapply the configuration encrypted by it because they are lost after the downgrade.
- You must regenerate crypto keys and certificates after a downgrade.

A*: You can't downgrade to the base ISO. You can downgrade to a GISO containing the bridge SMU (CSCwd71524).



Note CSCwd71524:

- When you upgrade from earlier than Release 7.10.1 to Release 7.10.1, system supports the installation process seamlessly.
- When you downgrade from Release 7.10.1, system preserves the present configuration and the install history from last transaction.

B*: You must recover the NCS 1010 platform through PXE if a power cycle occurs during the downgrade.

C*: FPD upgrade may cause your NCS 1010 platform to reload an extra time during boot-up.

D*: Traffic is impacted.

Use the **show install upgrade-matrix running** command to view the caveats.



CHAPTER 6

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, on page 73](#)
- [DHCP Relay Configuration for OLT Node, on page 73](#)
- [Loopback IP address for OSC Interface, on page 75](#)
- [OSPF Neighbor Discovery, on page 75](#)
- [Configure ILA Node, on page 76](#)
- [Configure OLT Node, on page 76](#)

Prerequisites

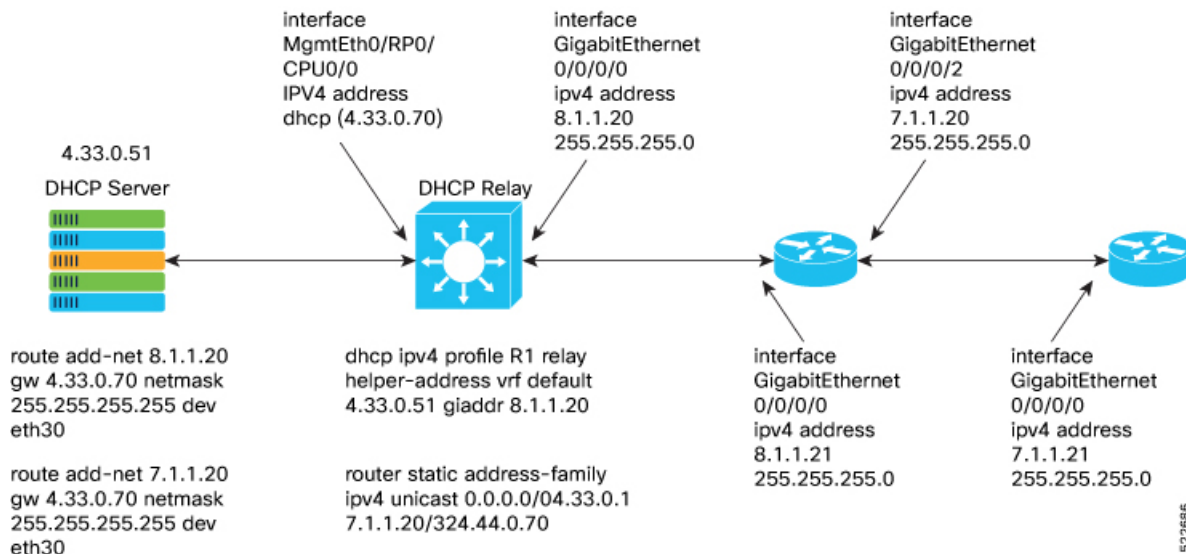
The remote node management for ILA node works only if the following conditions are met:

- The DSCP relay configuration for OLT node must be present. See [DHCP Relay Configuration for OLT Node, on page 73](#)
- The loopback address and IP address must be present for OSC interface. See [Loopback IP address for OSC Interface, on page 75](#)
- The OSPF neighbor discovery must be successful between OLT and ILA nodes. See [OSPF Neighbor Discovery, on page 75](#)

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 6: DHCP Relay Configuration for OLT Node



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:P2C_DT_02#sh 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
10.3.3.20        1    FULL/ -         00:00:35    10.8.1.20         GigabitEthernet0/0/0/0
Neighbor is up for 00:00:42

Total neighbor count: 1
RP/0/RP0/CPU0:P2C_DT_02#

```

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 7

Configuring BGP

- [BGP Overview, on page 77](#)
- [Prerequisites for Implementing BGP, on page 78](#)
- [BGP Router Identifier, on page 78](#)
- [Configuring BGP, on page 79](#)

BGP Overview

Table 10: Feature History

Feature Name	Release Information	Feature Description
Configuring BGP	Cisco IOS XR Release 7.11.1	BGP routers form 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.

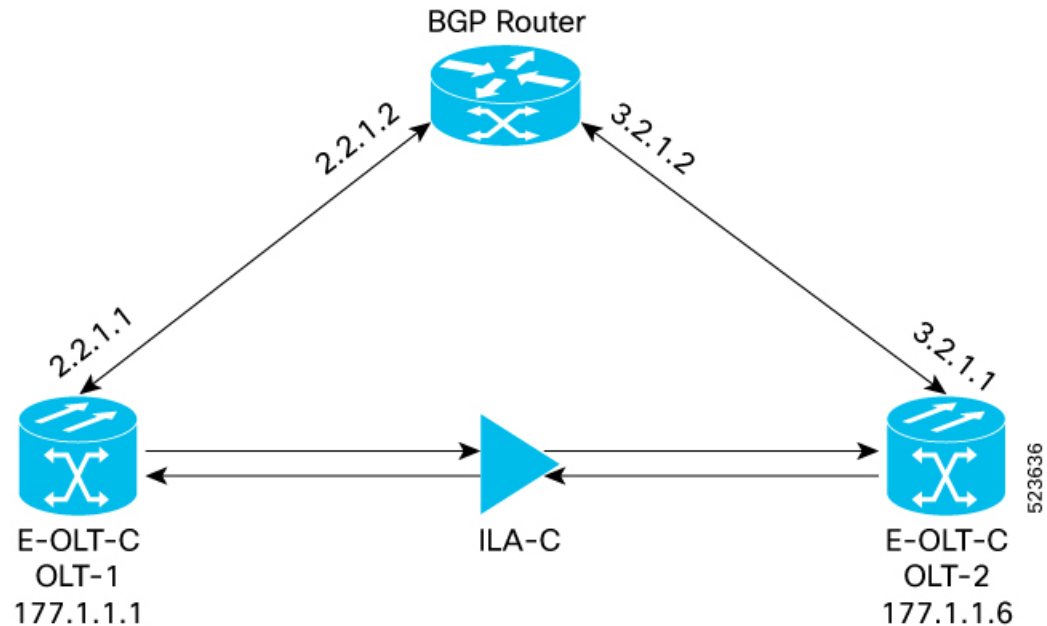
After BGP has obtained a router ID, it continues to use it even if a better router ID becomes available. This usage avoids unnecessary flapping for all BGP sessions. However, if the router ID currently in use becomes invalid (because the interface goes down or its configuration is changed), BGP selects 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 7: 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 177.1.1.1
  address-family ipv4 unicast
    redistribute connected
    redistribute ospf 1 route-policy pass-all
  !
  address-family vpnv4 unicast
  !
  neighbor 2.2.1.2
    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 177.1.1.1
  address-family ipv4 unicast
    redistribute connected
  
```

```

    redistribute ospf 1 route-policy pass-all
    !
    address-family vpnv4 unicast
    !
    neighbor 2.2.1.2
    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 177.1.1.6
  address-family ipv4 unicast
  redistribute connected
  redistribute ospf 1 route-policy pass-all
  !
  address-family vpnv4 unicast
  !
  neighbor 3.2.1.2
  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 15:59:12.210 IST
router bgp 1
  bgp router-id 177.1.1.6
  address-family ipv4 unicast
  redistribute connected
  redistribute ospf 1 route-policy pass-all
  !
  address-family vpnv4 unicast
  !
  neighbor 3.2.1.2
  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 1.1.1.99
  address-family ipv4 unicast
  redistribute connected
  !

```

```
neighbor 2.2.1.1
  remote-as 1
  address-family ipv4 unicast
    route-policy pass-all in
    route-policy pass-all out
  !
!
neighbor 3.2.1.1
  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 1.1.1.99
  address-family ipv4 unicast
    redistribute connected
  !
  neighbor 2.2.1.1
    remote-as 1
    address-family ipv4 unicast
      route-policy pass-all in
      route-policy pass-all out
    !
  !
  neighbor 3.2.1.1
    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 8

Configure CDP

Table 11: 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, on page 84](#)
- [Disable CDP Globally, on page 84](#)

- [Enable CDP on Interfaces, on page 84](#)
- [Modify CDP Default Settings, on page 85](#)
- [Monitor CDP, on page 86](#)

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.

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 9

Daisy Chain

This chapter describes the Daisy Chain optical application for Cisco NCS 1010.

- [Daisy Chain Overview, on page 89](#)
- [Configure Daisy Chain on Management Ports, on page 90](#)
- [Verify Daisy Chain, on page 91](#)
- [Enable Storm Control on TOR Switch, on page 92](#)
- [Disable DAD on Management Port, on page 92](#)

Daisy Chain Overview

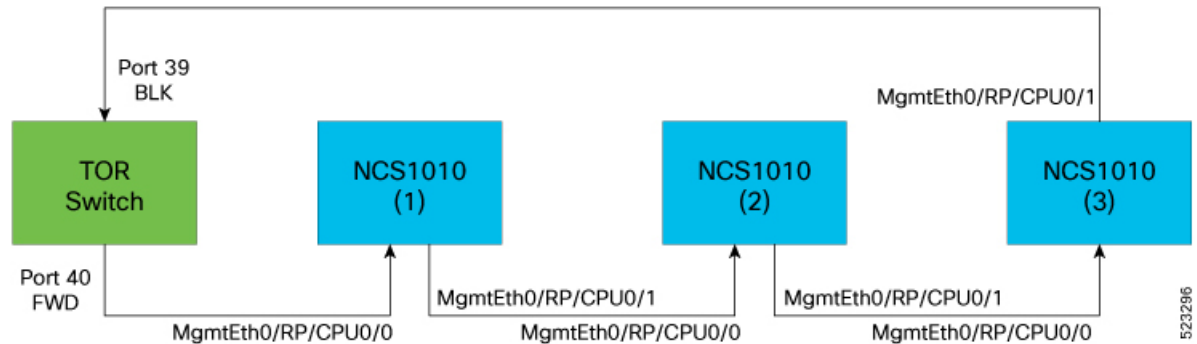
Table 12: Feature History

Feature Name	Release Information	Description
Daisy Chain on NCS 1010 Management Ports	Cisco IOS XR Release 7.10.1	<p>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.</p> <p>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.</p>

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 8: 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 Daisy Chain on Management Ports](#)
- [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 typeRack/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 interfacetype
```

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 10

Configure Access Control List

This chapter describes the Access Control List (ACL) and the procedures to configure ACL.

- [Understand Access Control List, on page 95](#)
- [Ingress and Egress Access Control Lists, on page 97](#)
- [How an Access Control List Works, on page 97](#)
- [Configure IPv4 Standard Access Control List on Management Ethernet Interface, on page 98](#)
- [Configure IPv6 Standard Access Control List on Management Ethernet Interface, on page 101](#)
- [Configure an Extended Access Control List, on page 103](#)
- [Modify an Access Control List, on page 104](#)

Understand Access Control List

Table 13: 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: <ul style="list-style-type: none">• ipv4-access-list• ipv4-access-group• show access-lists-ipv4• ipv6-access-list• ipv6-access-group• show access-lists-ipv6

Access Control List

Access Control List 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 to 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 List

Table 14: 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.
Extended ACL	<ul style="list-style-type: none"> • source address • destination address • UDP or TCP port numbers • 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.
- control vty access.

Guidelines to Create an Access Control List

- 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 List

- You can configure an ACL name with a maximum of 64 characters.
- You can configure an ACL name to comprise only letters and numbers.
- You can configure an ACL to control traffic ingressing or egressing a device but not traffic originating at the device.

Ingress and Egress Access Control Lists

Entries within the ACL are based on the direction of the flow of traffic from the point of view of the management interface of NCS 1000. ACL is applied either on the ingress or egress interfaces. After receiving a packet, the software checks the source address of the packet against the ingress or egress ACL.

Table 15: Ingress and Egress ACLs

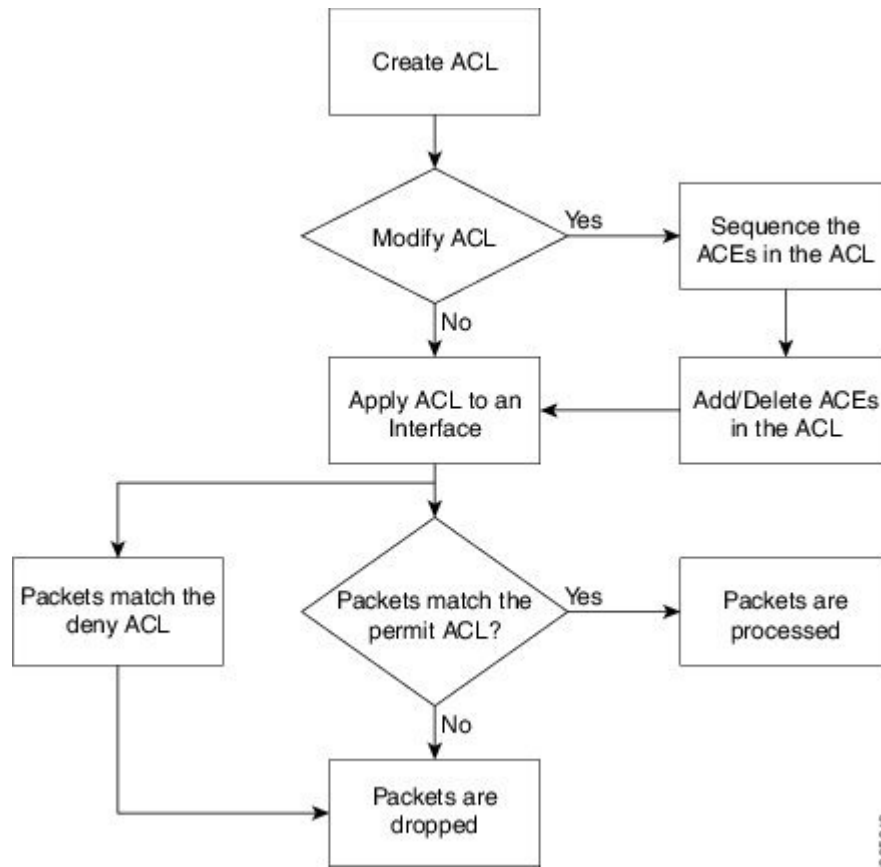
ACL	Traffic Control	When ACL permits the source address	When ACL rejects the source address
Ingress ACL	controls traffic that comes from a network to the management interface.	software continues to process the packet.	software discards the packet and returns an Internet Control Message Protocol (ICMP) host unreachable message.
Egress ACL	controls traffic that comes from the management interface to the network.	software sends the packet.	software discards the packet and returns an ICMP host unreachable message.

How an Access Control List Works

ACL Workflow

The following image illustrates the workflow of an ACL.

Figure 9: ACL Workflow



Configure IPv4 Standard Access Control List on Management Ethernet Interface

Use this task to configure an ingress or egress IPv4 standard ACL on the management Ethernet interface.

Before you begin

Review the "Guidelines" and "Restrictions" sections of [Understand Access Control List, on page 95](#).

Step 1 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 4.33.0.57 255.255.255.0
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 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	3.3.3.16	Up	Up	default
GigabitEthernet0/0/0/0	7.1.11.5	Up	Up	default
GigabitEthernet0/0/0/2	9.1.11.5	Up	Up	default
MgmtEth0/RP0/CPU0/0	4.33.0.57	Up	Up	default
PTP0/RP0/CPU0/0	unassigned	Shutdown	Down	default
MgmtEth0/RP0/CPU0/1	8.1.1.1	Up	Up	default
PTP0/RP0/CPU0/1	unassigned	Shutdown	Down	default
MgmtEth0/RP0/CPU0/2	192.0.2.1	Down	Down	default

Step 3 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 [Verify](#) the ACL creation.

The entries highlighted in bold show the successful creation of ingress ACL and egress 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 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 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 4.33.0.57/16
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 unreachable 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 4.33.0.57/16
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 unreachable 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

Use this task to configure an ingress or egress IPv6 standard ACL on the management Ethernet interface.

Before you begin

Review the "Guidelines" and "Restrictions" sections of [Understand Access Control List, on page 95](#).

Step 1 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 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 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 Verify the ACL creation.

The entries highlighted in bold show the successful creation of ingress ACL and egress 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 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 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 unreachablees 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 */

RP0/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 unreachablees 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 [Understand Access Control List, on page 95](#).

Step 1 Configure the [ACL](#).

Example:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ipv4 access-list acl_1
```

Step 2 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 [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 [Understand Access Control List, on page 95](#).

Step 1 Configure the [ACL](#).

Example:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ipv4 access-list acl_1
```

Step 2 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 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 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 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.

