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

Cisco Network Convergence System 1010
Updated September 11, 2026

© 2026 Cisco Systems, Inc. All rights reserved.

Full Cisco Trademarks with Software License

Full Cisco Trademarks with Software License

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

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

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

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

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

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

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

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

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)

Topics included

Full Cisco Trademarks with Software License....ii

1 Cisco NCS 1010 Optical Line System....9

Cisco NCS 1010 chassis and line cards....10
Optical line card variants....10
External interface timing unit....13
Cisco NCS 1000 passive modules....14
Cisco NCS 1000 breakout patch panel....14
Cisco NCS 1000 32-channel mux/demux patch panels....15

2 Software Setup Fundamentals....17

Types of software releases for Cisco NCS 1010....18
Downloadable software files for Cisco NCS 1010....18
Command modes for Cisco NCS 1010....19

3 Setup Procedures....21

Pre-setup requirements for Cisco NCS 1010....22
Connect the console port to a terminal....22
Configure the management interface....23
Cisco NCS 1010 setup workflow....33
Boot Cisco NCS 1010....33
Install a new image without Golden ISO....43
Build a Golden ISO boot image for Cisco NCS 1010....44
Network Time Protocol....46
Cisco NCS 1010 software and hardware verification....51
Verify the software version....51
Verify hardware modules....52
Verify interface status....59
Verify node status....60
Verify hardware inventory....61
Verify management interface status....63
Verify alarms....65
Verify environmental parameters....66
Verify core dump context....76
Verify core files....77
Verify memory information....77
Cisco NCS 1010 post-setup tasks....78
Create a user profile....79

Create user groups....80

4 TACACS+ and RADIUS Configuration....83

TACACS+ and RADIUS configuration details....84
Deprecation of type 7 password and type 5 secret....84
TACACS+ protocol....91
Configure TACACS+ server....91
Configure and verify TACACS+ server groups....92
Configure TACACS+ server group commands....93
Verify TACACS+ server group configuration....94
RADIUS protocol....95
Configure and verify RADIUS server groups....95
Configure RADIUS server group commands....96
Verify RADIUS server group configuration....97

5 Setup and Upgrade Workflow....99

NCS 1010 upgrade....100
Supported upgrade and downgrade releases....100
Software upgrade plan....101
Back up the current configuration....101
Field programmable devices....102
Upgrade FPDs automatically....107
Upgrade FPDs manually....107
FPD upgrades using YANG data models....109
FPD downgrade scenarios....109
System stability checks....110
Install file sources....111
Standard ISO and RPMs....112
Download install files from Cisco Software Center....112
Software upgrade methods....112
CLI software upgrade method for Cisco NCS 1010....113
Install ISO and RPMs....113
Install golden ISO....115
Data model software upgrade method....118
Access install-related data models....118
Manageability agent connections to Cisco NCS 1010....120
Generate RPC messages to install an IOS XR image....120
Software upgrade verification....121
System stability checks before upgrade....122

6 Classic ZTP Deployment....125

DHCP configuration....126
DHCP relay....126

Prerequisites for configuring DHCP relay agent....126
Limitations for DHCP relay feature....127
DHCP relay agent configuration....127
DHCP client....128
Provision Cisco NCS 1010 using DHCP-based ZTP fresh boot....128
Configuration file requirements....130
Configure a ZTP bootscript....131
Invoke ZTP manually through CLI....134
Invoke ZTP through reload....135
Data port authentication....137
DHCP server setup for ZTP....139
ZTP initialization file options....141
How classic ZTP provisioning works....144
ZTP logs....145
Generate tech support information for ZTP....146

7 Software Maintenance....149
Install additional RPMs and bug fixes....150
Install RPMs using command line interface....150
Install RPMs using YANG data model....154
Downgrade software version....154
Downgrade to a previously installed package....157
Telemetry sensor paths for install operations....159

8 Setup and Upgrade Troubleshooting....161
NCS 1010 boot failure recovery....162
Boot the NCS 1010 using USB drive....162
Boot the NCS 1010 using iPXE....164
Recover password....168
Resolve insufficient disk space during software installation....170
Recover frozen console prompt....172

9 Disaster Recovery....173
Disaster recovery partitions....174
CPU replacement considerations....174
Backup ISO image health check....174
Automated file management behavior....175

10 Border Gateway Protocol Configuration....177
BGP routing....178
BGP implementation prerequisites....178
BGP router identifier....179
Configure BGP....179

11 Cisco Discovery Protocol Configurations....185
CDP support on Cisco NCS 1010....186
Enable CDP globally....186
Disable CDP globally....187
Enable CDP on interfaces....187
Modify CDP default settings....188
Monitor CDP....190

12 Daisy Chain Topology....195
Daisy chain topology....196
Configure daisy chain on management ports....197
Verify daisy chain....198
Enable storm control on TOR switch....198
Disable DAD on the management port....199

13 Access Control Lists....201
Access control lists....202
How access control lists work....203
Access control list application behavior....205
ACL configuration procedures....205
Configure an ingress IPv4 ACL on management Ethernet interface....206
Configure an egress IPv4 ACL on the management Ethernet interface....208
Configure an ingress IPv6 ACL on the management Ethernet interface....210
Configure an egress IPv6 ACL on the management Ethernet interface....213
Configure extended access lists....215
Modify ACLs....216

14 Remote Node Management....219
Remote node management details for NCS 1010....220
Remote node management with OSC....220
Remote node management prerequisites....220
DHCP relay configuration for OLT node....220
Loopback IP address for OSC interface....222
OSPF neighbor discovery....222
ILA node configuration....223
OLT node configuration....223

15 Remote Console Connection Workflow....225
Remote console connection details....226
Remote console connection support for Cisco NCS 1010....226
Find the MAC address of all the nodes....227

Connect to a remote node....230

16 Automated File Management....233

Automated file management behavior....234

17 Audit Log Storage and Monitoring....235

Audit logging and monitoring details....236
Audit logs....236
How audit logging works....237
Guideline: Use audit logging....238
Note: Review audit log storage behavior....238
Configure audit logging....238
AuditD process keepalive....240
Guidelines for AuditD process keepalive....241
Restrictions for AuditD process keepalive....241
Configure the AuditD process keepalive....242

18 Process Memory Management....245

Process memory management details....246
Core dump folders....246
Requirement: Maintain core dump folder disk space and usage thresholds....247
How core dump folder limit works....247
Configure core dump folder limit....251

1 Cisco NCS 1010 Optical Line System

Topics:

- Cisco NCS 1010 chassis and line cards
- Cisco NCS 1000 passive modules

This chapter provides details about the Cisco NCS 1010 optical line system, including chassis components, line card variants, EITU interfaces, passive modules, breakout modules, and mux/demux patch panels.

Cisco NCS 1010 chassis and line cards

This section provides details about the Cisco NCS 1010 chassis and line cards, including WDM router connectivity, fiber multiplexing, degree scaling, transmission bands, and field-replaceable modules.

Use this reference to review Cisco NCS 1010 chassis and line cards information for the Cisco NCS 1010 optical line system.

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.

Optical line card variants

This section provides details about Cisco NCS 1010 optical line card variants and identifies the C-band, L-band, Raman, and in-line amplifier functions supported by each card.

Use this reference to review Optical line card variants information for the Cisco NCS 1010 optical line system.

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

This section provides details about the OLT-C line card, including amplifier functions, monitoring capabilities, optical service channel support, and other features used in the Cisco NCS 1010 optical line system.

Use this reference to review OLT-C line card information for the Cisco NCS 1010 optical line system.

The C-band Optical Line Terminal without Raman (OLT-C) line card includes these 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) and up to 30 EXP ports
- Dedicated Tunable Laser (TL) enabling Connection Verification (CV) and patch cord discovery features
- Embedded Optical Service Channel at Fast Ethernet (FE)

The OLT-C line card also includes these features:

- 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

This section provides details about the OLT-R-C line card, including OLT-C capabilities and Raman amplifier features such as pump wavelengths, monitoring, reflection detection, and eye-safety control.

Use this reference to review OLT-R-C line card information for the Cisco NCS 1010 optical line system.

The OLT-R-C line card includes the following details.

- 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

This section provides details about the ILA-C line card, including EDFA gain functions, dynamic gain equalization, OTDR monitoring, OCM channel monitoring, and optical service channel support.

Use this reference to review ILA-C line card information for the Cisco NCS 1010 optical line system.

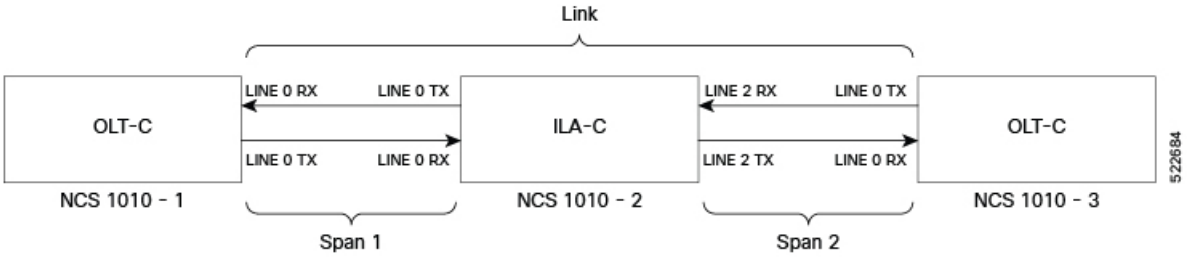
The C-band In Line Amplifier without Raman (ILA-C) line card includes the following features:

- 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 and 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

This 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

This section provides details about the ILA-R-C line card and its support for ILA-C functions with Raman amplification in the Cisco NCS 1010 optical line system.

Use this reference to review ILA-R-C line card information for the Cisco NCS 1010 optical line system.

The ILA-R-C line card includes 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

This section provides details about the ILA-2R-C line card and its support for ILA-C functions with Raman amplification on both transmission directions.

Use this reference to review ILA-2R-C line card information for the Cisco NCS 1010 optical line system.

The ILA-2R-C line card includes 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

This section provides details about the OLT-L line card, including L-band amplifier functions, monitoring capabilities, optical service channel support, and C-band upgrade considerations.

Use this reference to review OLT-L line card information for the Cisco NCS 1010 optical line system.

The L-band Optical Line Terminal (OLT-L) line card includes these 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

The OLT-L line card also includes these features:

- 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

This section provides details about the ILA-L line card, including L-band EDFA gain functions, dynamic gain equalization, OCM channel monitoring, and optical service channel support.

Use this reference to review ILA-L line card information for the Cisco NCS 1010 optical line system.

The L-band In Line Amplifier (ILA-L) line card includes these 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

This section provides details about the user interfaces available on the External Interface Timing Unit, including timing, management, console, optical PTP, UDC, USB, and GPS interfaces.

Use this reference to review External interface timing unit information for the Cisco NCS 1010 optical line system.

The External interface timing unit

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

This list describes 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

This section provides details about Cisco NCS 1000 passive modules and explains how they support long-haul and metro topology deployments with the Cisco NCS 1010 chassis.

Use this reference to review Cisco NCS 1000 passive modules information for the Cisco NCS 1010 optical line system.

The Cisco NCS 1000 passive modules includes these details.

- 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

This section provides details about the Cisco NCS 1000 breakout patch panel, including power connections, breakout module connectivity, channel support, rack options, and fiber protection.

Use this reference to review Cisco NCS 1000 breakout patch panel information for the Cisco NCS 1010 optical line system.

Cisco NCS 1000 Breakout Patch Panel is colorless breakout-modular patch panel that has these details:

- 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.

NCS1K-BRK-8 breakout module

This section provides details about the NCS1K-BRK-8 breakout module, including MPO-to-LC adaptation, optical splitter behavior, monitoring support, and port connection information for deployments.

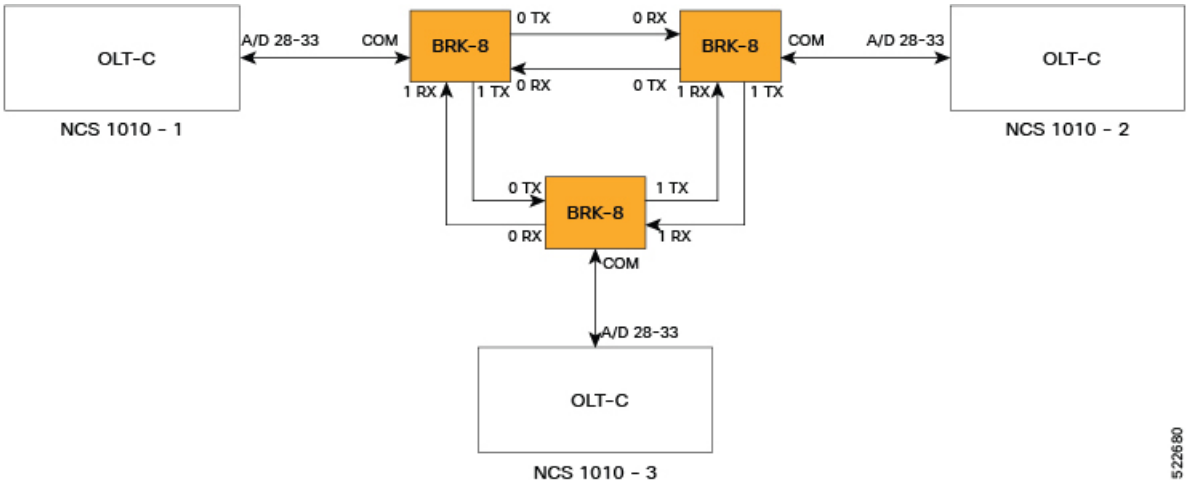
Use this reference to review NCS1K-BRK-8 breakout module information for the Cisco NCS 1010 optical line system.

The NCS1K-BRK-8 breakout module has these details.

- 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.

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

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



NCS1K-BRK-24 breakout module

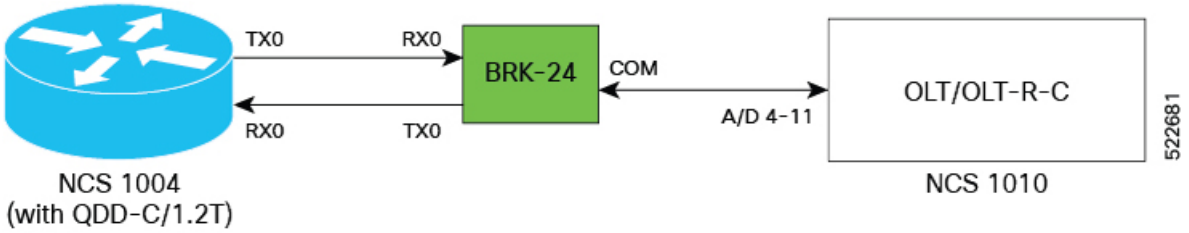
This section provides details about the NCS1K-BRK-24 breakout module, including MPO-to-LC breakout behavior, splitter functions, monitoring support, and port connection information for deployments.

Use this reference to review NCS1K-BRK-24 breakout module information for the Cisco NCS 1010 optical line system.

The NCS1K-BRK-24 breakout module has the following details.

- The NCS1K-BRK-24 module provides the breakout of 16 fibers from an MPO-24 connector to 24 duplex LC connectors. The signals on each fiber from the MPO input ports are split over three LC output ports by a 1x3 optical splitter. The signals from the three adjacent input LC ports are combined into a single MPO fiber output port through a 1x3 optical coupler. For each port (MPO and LC), power monitors with tone detection capability are available. A filtered optical loopback (191.175 THz) from one MPO input port (fiber-1) to all MPO output ports is available for connection verification.
- The following image displays port connections between BRK-24 panel and NCS 1010 and NCS 1004 chassis.

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



Cisco NCS 1000 32-channel mux/demux patch panels

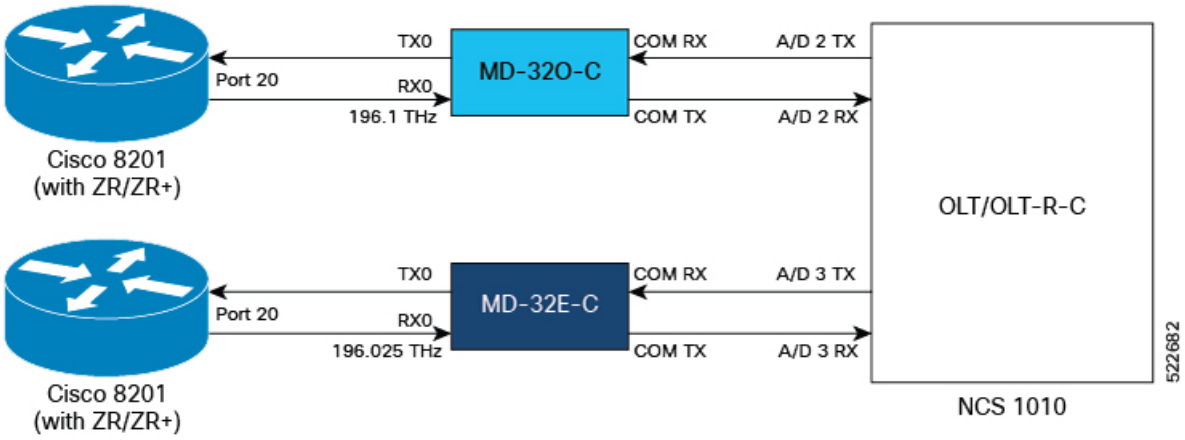
This section provides details about Cisco NCS 1000 32-channel mux/demux patch panels, including channel support, passive operation, monitoring capabilities, and port connection information.

Use this reference to review Cisco NCS 1000 32-channel mux/demux patch panels information for the Cisco NCS 1010 optical line system.

The Cisco NCS 1000 32-channel mux/demux patch panels has the following details.

- 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



2 Software Setup Fundamentals

Topics:

- Types of software releases for Cisco NCS 1010
- Downloadable software files for Cisco NCS 1010
- Command modes for Cisco NCS 1010

This chapter explains Cisco NCS 1010 software setup fundamentals, including release types, software download files, and command modes. Use this information to prepare for setup and software upgrade procedures.

Types of software releases for Cisco NCS 1010

This section describes the feature, maintenance, and SMU releases available for Cisco NCS 1010. Use this information to compare release types and locate downloads or documentation.

The Cisco IOS XR software model for NCS 1010 offers three primary types of software releases. Software images are available for download at the Cisco Software Download page. You can use these descriptions to identify release types, locate command outputs in documentation, and find supporting resources during NCS 1010 setup or troubleshooting.

Table 1: Types of software releases

Release Type	Description	Example	Where to Find
Feature Release (FR)	Contains new features and supports new hardware; features are added and installation notes are provided in release notes.	X.X.1 format (e.g., 24.1.1, 24.2.1)	Release notes, Cisco download site
Maintenance Release	Delivers groups of critical bug fixes to feature releases.	Referenced in release notes	Release notes, Cisco download site
Software Maintenance Unit (SMU)	Provides specific fixes until the release's End of Maintenance (EoM); fixes are merged into the next shipping release. SMUs are customized and posted as "Cisco IOS XR Software Maintenance Upgrade."	Customized for a specific release	Cisco download site, under *Cisco IOS XR Software Maintenance Upgrade*.

For detailed information about numbering, release types, and timelines, see: Software Lifecycle Support Statement - IOS XR

Downloadable software files for Cisco NCS 1010

This section provides details about Cisco Software Download files for Cisco NCS 1010. Use the information to compare values, confirm command output, or find supporting resources during setup.

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

Table 2: IOS XR Software Installation Files in Cisco Software Download Page

Package File	Example	Description
ncs1010-x64-<**rel. no.**>.iso	ncs1010-x64-24.3.1.iso	This file includes the required core packages. These packages comprise the operating system, Admin, Base, Forwarding, SNMP Agent, FPD, Alarm Correlation, Netconf-yang, Telemetry, Extensible Markup Language (XML) parser, and HTTP server packages.

Individually Installable Packages

Package File	Example	Description
xrtelnet-<**rel. no.**>x86_64.rpm xr-telnet-ncs1010-<**rel. no.**>.x86_64.rpm	xr-telnet-24.3.1.v1.0.0-1.x86_64.rpm xr-telnet-ncs1010-24.3.1.v1.0.0-1.x86_64.rpm	Install these packages to support Telnet.
xrcdp-rpms.<**rel. no.**>.rpm xr-cdp-ncs1010-<**rel. no.**>.x86_64.rpm	xr-cdp-24.3.1.v1.0.0-1.x86_64.rpm xr-cdp-ncs1010-24.3.1.v1.0.0-1.x86_64	Install these packages to support CDP.

Command modes for Cisco NCS 1010

This section lists the command modes available in Cisco NCS 1010 IOS XR and describes their functions.

The NCS 1010 operates using virtualized Cisco IOS XR software. CLI commands are executed on the IOS XR virtual machine.

The command modes apply to the Cisco NCS 1010 Series. The table lists available command modes for IOS XR.

Command Mode	Description
IOS XR EXEC mode (IOS XR execution mode)	Use commands on IOS XR to display the operational state of NCS 1010. **Example:** `RP/0/RP0/CPU0:ios#`

IOS XR Config mode
(IOS XR configuration mode)
	Configure security, routing, and other XR features on IOS XR.
	Example:
	`RP/0/RP0/CPU0:ios#configure`
	`RP/0/RP0/CPU0:ios(config)`

3 Setup Procedures

Topics:

- Pre-setup requirements for Cisco NCS 1010
- Cisco NCS 1010 setup workflow
- Cisco NCS 1010 software and hardware verification
- Cisco NCS 1010 post-setup tasks

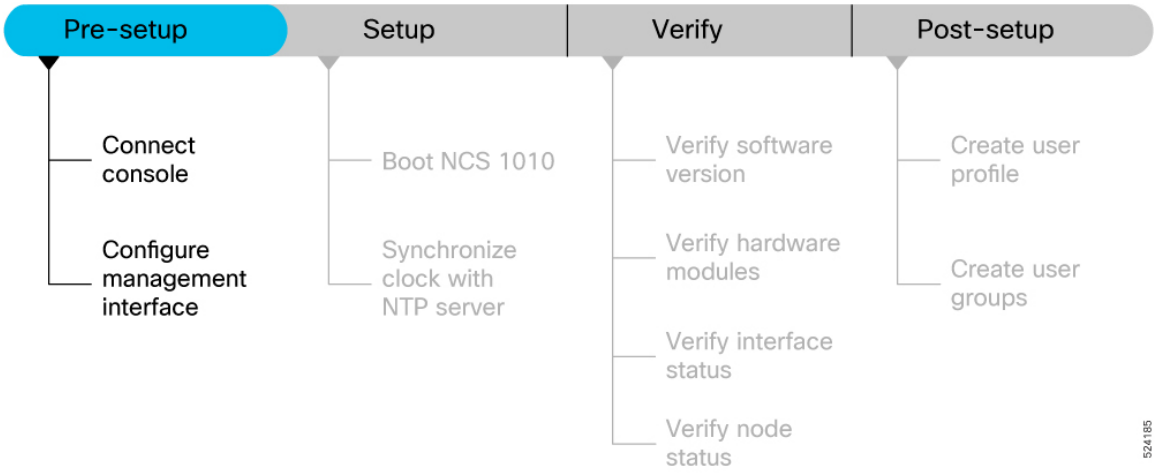
This chapter provides a navigation point for Cisco NCS 1010 setup tasks and references that prepare the router for boot, management access, verification, and administration.

Pre-setup requirements for Cisco NCS 1010

Provides pre-setup requirements for Cisco NCS 1010, including console access, management interface preparation, and required setup tasks before boot verification.

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

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



Connect the console port to a terminal

Connect the console port to a terminal to access Cisco NCS 1010 before management networking is available and continue with initial setup.

Connect your terminal or PC to the Cisco NCS 1010 console port and configure terminal settings for direct device access.

The console port allows you to log into the NCS 1010 without a network connection, using an emulation program such as HyperTerminal.

Procedure

1. Connect the console (or rollover) cable to the console port on the NCS 1010.
2. Use the correct adapter to connect the other end of the cable to your terminal or PC.
3. Launch the terminal session.
4. In the **COM1 Properties** window, select **Port Settings** tab, and enter these settings:

Setting	Value
Speed	9600
Data Bits	8
Parity	None
Stop bits	1
Flow Control	None

5. Click **OK**.

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

The terminal or PC is connected to the console port, and the terminal session is ready for initial access.

Configure the management interface

Configure the management interface to assign network access details and enable remote connections through SSH or Telnet for device administration.

Use this procedure to configure the management interface.

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

Before you begin

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

Procedure

---

1. Enter configuration mode.

   configure

   Example

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

   Enters IOS XR configuration mode.
2. Enter management interface configuration mode.

   interface mgmtEth 0/RP0/CPU0/0

   Example

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

   Enters interface configuration mode for the management interface.
3. Configure the IPv4 address and subnet mask.

   ipv4 address 192.0.2.254 255.255.255.0

   Example

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

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

   no shutdown

Example

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

Places the management interface in an "up" state.

5. Exit interface configuration mode.

   exit

   Example

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

   Exits the management interface configuration mode.

6. Configure a static route.

   ncs1010 static address-family ipv4 unicast 0.0.0.0/0 198.51.100.4

   Example

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

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

7. Save or exit the configuration session.

   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.

The configure the management interface task is complete.

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.

LLDP support on the management interface

Provides LLDP support details for the Cisco NCS 1010 management interface, including supported behavior, neighbor discovery, interoperability, supported OIDs, configuration attributes, limitations, and default interface behavior.

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

Advantages of LLDP

- Provides support on non-Cisco devices.

- Enables neighbor discovery between non-Cisco devices.

Limitation

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

> **Note**
>
> By default, LLDP is enabled for NCS 1010. But when you enable and disable LLDP in the global configuration mode, LLDP gets disabled on all the interfaces.
>
> **Workaround**: You must enable LLDP globally or reload the NCS1010.

Cisco Discovery Protocol (CDP) vs LLDP

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

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

Interoperability between non-Cisco devices using LLDP

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

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

This table describes the global LLDP attributes that you can configure:

Table 3:

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)

Configure LLDP globally

Configure LLDP globally to enable LLDP on all three management interfaces and set the global LLDP holdtime and reinitialization attributes.

Enable LLDP globally on all three management interfaces.

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.

Procedure

1. Enter global configuration mode.

   configure terminal

   Example

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

2. Enable LLDP on the management interfaces.

   lldp management enable

   Example

   ```
   RP/0/RP0/CPU0:ios(config)#lldp management enable
   ```

3. Set the LLDP holdtime.

   lldp holdtime 30

   Example

   ```
   RP/0/RP0/CPU0:ios(config)#lldp holdtime 30
   ```

4. Set the LLDP reinitialization delay.

   lldp reinit 2

   Example

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

5. Commit the configuration.

   commit

   Example

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

The global LLDP configuration enables LLDP on all the three management interfaces.

Verify LLDP configuration and data

Verify LLDP running configuration, interface data, and neighbor data on the Cisco NCS 1010 management interface after LLDP configuration changes.

Verify the LLDP configuration and operational data.

Procedure

1. Verify the LLDP running configuration.

   show running-config lldp

   Example

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

2. Verify the LLDP interface data.

   show lldp interface

   Example

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


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


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


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

3. Verify the LLDP neighbor data.

   show lldp neighbors

Example

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

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

Total entries displayed: 2
```

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

> **Note**
>
> If the RCOM interface is enabled, the output of **show lldp neighbors** command would include the entries for both LLDP neighbours and remote connect neighbours.

The LLDP running configuration, interface data, and neighbor data are verified.

Configure LLDP on a management interface

Configure LLDP at the management interface level to selectively enable LLDP on a specified Cisco NCS 1010 management interface for neighbor discovery.

Enable LLDP at the management interface level.

Procedure

---

1. Enter the management interface configuration mode.

   interface mgmtEth 0/RP0/CPU0/X

   Example

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

2. Enable LLDP on the management interface.

   lldp enable

   Example

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

3. Commit the configuration.

   commit

Example

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

LLDP is configured at the management interface level.

Disable LLDP transmit and receive operations

Disable LLDP transmit or receive operations on a specified Cisco NCS 1010 management interface to control LLDP behavior for that interface.

Disable LLDP transmit operations, receive operations, or both at a specified management interface.

Procedure

1. Enter the management interface configuration mode.

   interface mgmtEth 0/RP0/CPU0/X

   Example

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

2. Disable LLDP transmit operations at the specified management interface.

   lldp transmit disable

   Example

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

3. Disable LLDP receive operations at the specified management interface.

   lldp receive disable

   Example

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

4. Commit the configuration.

   commit

   Example

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

LLDP transmit or receive operations are disabled at the specified management interface.

LLDP traffic and debug commands

Provides LLDP traffic and debug commands that help identify LLDP behavior, neighbor discovery status, packet exchange, and interface-level troubleshooting details.

Use these commands for debugging issues in the LLDP functionality.

Table 4: LLDP troubleshooting commands

Command	Description
show lldp traffic	Displays statistics for LLDP traffic.
debug lldp all	Enables all LLDP debugging information.
debug lldp errors	Enables debugging information for LLDP errors.
debug lldp events	Enables debugging information for LLDP events.
debug lldp packets	Enables debugging information for LLDP packets.
debug lldp tlvs	Enables debugging information for LLDP TLVs.
debug lldp trace	Enables LLDP trace debugging information.
debug lldp verbose	Enables verbose LLDP debugging information.

Configure Telnet

Configure Telnet on Cisco NCS 1010 to provide remote management access after the required packages are installed and management connectivity is available.

Use this procedure to configure Telnet.

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

Before you begin

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

Procedure

1. Enter configuration mode.

   configure

   Example

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

   Enters the configuration mode.

2. Specify the number of allowable Telnet servers.

   telnet ipv4 server max-servers 10

   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.

3. Save or exit the configuration session.

   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.

The configure Telnet task is complete.

Configure SSH

Configure SSH on Cisco NCS 1010 to enable secure remote access to the management interface after the required management connectivity is available.

Use this procedure to configure SSH.

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

Before you begin

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

Procedure

---

1. Enter configuration mode.

   configure

   Example

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

   Enters the configuration mode.

2. Enable SSH server version 2.

   ssh server v2

   Example

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

   Enables the SSH server to accept only SSHv2 client connections.

3. Save or exit the configuration session.

   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.

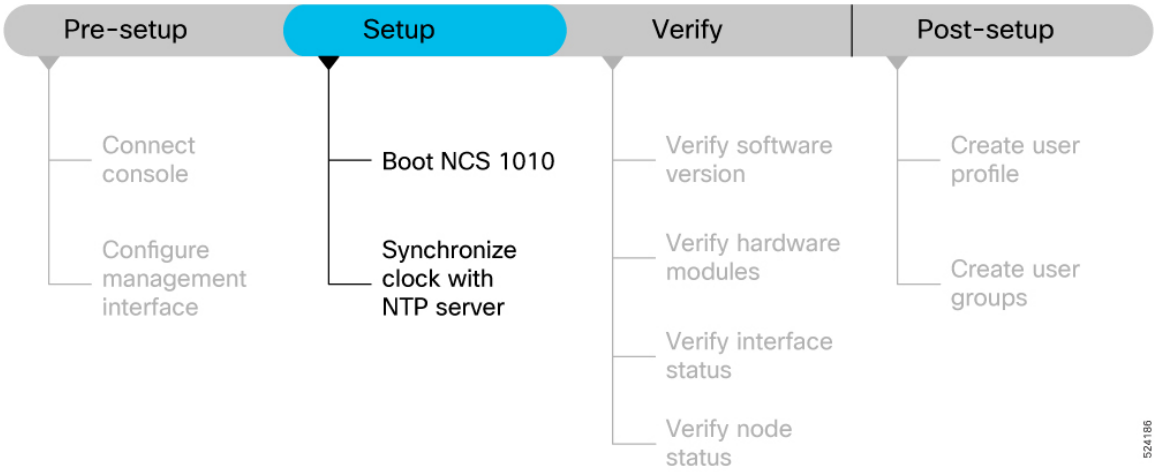
The configure SSH task is complete.

Cisco NCS 1010 setup workflow

Provides the Cisco NCS 1010 setup workflow and related setup task sequence used to bring up the system for further configuration.

Complete these tasks to bring up your NCS 1010 for further configuration.

Figure 6: Setup workflow for Cisco NCS 1010



Boot Cisco NCS 1010

Boot Cisco NCS 1010 to start the system and prepare it for the remaining setup, verification, management access, and administration tasks.

Use this procedure to boot Cisco NCS 1010.

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

Procedure

1. Connect a terminal to the console port of the RP.
2. Start the terminal emulation program on your workstation.

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

> **Note**
>
> 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.

The boot Cisco NCS 1010 task is complete.

Boot Cisco NCS 1010 using a USB drive

Boot Cisco NCS 1010 from a USB drive when the setup workflow requires local boot media, then continue with the preparation and startup subtasks.

Use this supertask to prepare a bootable USB drive and boot Cisco NCS 1010 from that 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*.

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.

Procedure

1. Prepare a USB boot drive.

   See Prepare a USB boot drive.

2. Start Cisco NCS 1010 from a USB drive.

   See Start Cisco NCS 1010 from a USB drive.

Cisco NCS 1010 boots from the USB image and reboots after installation.

Prepare a USB boot drive

Prepare a USB boot drive with the required image so Cisco NCS 1010 can start from local boot media during setup.

Use this procedure to format the USB drive, copy the compressed boot file, verify the file, and extract the contents at the root of the 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*.

The prepared USB drive contains the extracted boot files that make the drive bootable.

Procedure

1. Connect the USB drive to your local machine and format it with the FAT32 file system.
2. Copy the compressed boot file to the USB drive.
3. Verify that the copy operation is successful. To verify, compare the file size at source and destination. Also, verify the MD5 checksum value.
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.

The USB drive is ready to boot Cisco NCS 1010.

Start Cisco NCS 1010 from a USB drive

Start Cisco NCS 1010 from a USB drive after preparing boot media so the chassis can load the required setup image.

Use this procedure to insert the prepared USB drive, select the BIOS boot option, and remove the drive after the image loads.

Before you begin

Prepare the USB boot drive before you start Cisco NCS 1010 from the drive. See Prepare a USB boot drive.

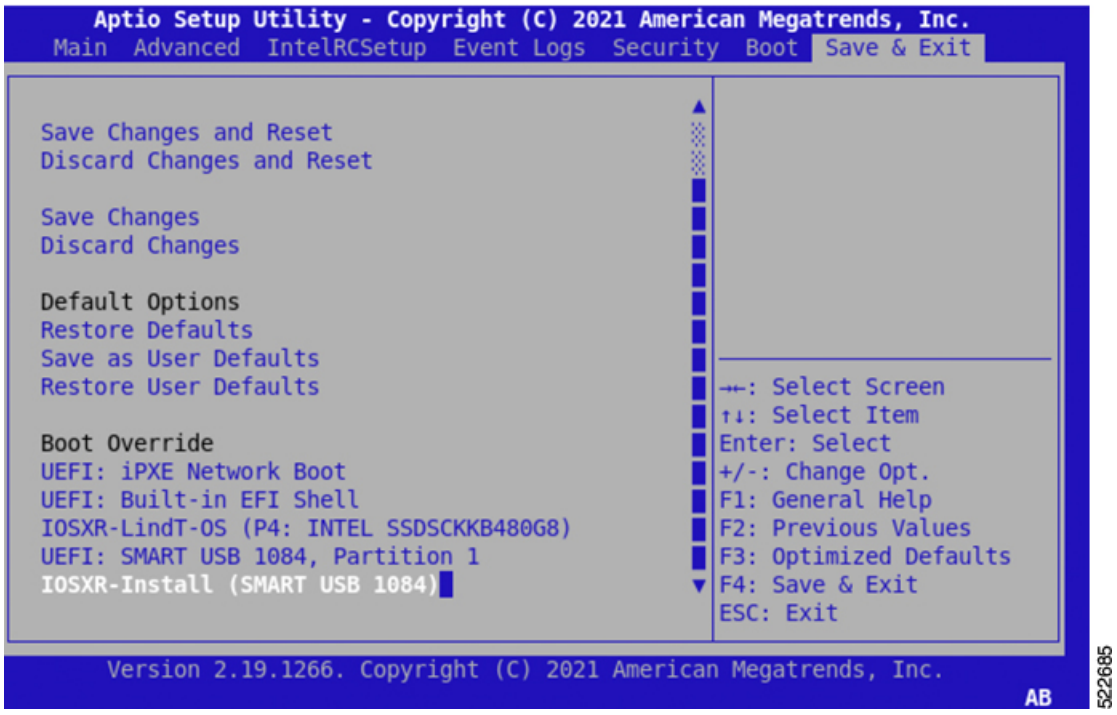
Use the console or BIOS boot option when you need to boot Cisco NCS 1010 from the prepared USB drive.

Procedure

1. Insert the USB drive in one of the USB ports of NCS 1010 line card/controller card.
2. 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.

3. Press **Esc** to enter BIOS.
4. Select the **Save & Exit** tab of BIOS.



5. 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...
```

6. 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.

Cisco NCS 1010 starts from the USB drive and reboots after installation.

Boot Cisco NCS 1010 using iPXE

Configure iPXE boot for Cisco NCS 1010 so that the chassis can download an ISO image from a network server and reimage the system.

Use iPXE boot to reimage Cisco NCS 1010 through a network boot workflow.

Use iPXE boot when the router fails to boot or when no valid bootable partition is available. iPXE enables network boot for an offline router. The iPXE bootloader downloads and installs the ISO image located on an HTTP, FTP, or TFTP server and reimages the router.

iPXE boot can be invoked through the CLI terminal or through the BIOS interface.

Before you begin

- Ensure that the DHCP server is set and running. For details, see Configure a DHCP server for iPXE boot.
- Ensure that the management port of the NCS 1010 chassis is in *UP* state.

Procedure

1. Invoke iPXE boot through the CLI terminal.

   For details, see Boot Cisco NCS 1010 using iPXE from the CLI.

2. Invoke iPXE boot through the BIOS interface.

   For details, see Boot Cisco NCS 1010 using iPXE from BIOS.

The iPXE boot process downloads the ISO image and reimages the Cisco NCS 1010 chassis.

Configure a DHCP server for iPXE boot

Configure a DHCP server for Cisco NCS 1010 iPXE boot so that the chassis can locate the ISO image or provisioning configuration file during boot.

Configure a DHCP server to provide Cisco NCS 1010 iPXE boot information.

A DHCP server must be configured for IPv4, IPv6, or both communication protocols before Cisco NCS 1010 can use iPXE boot.

For DHCPv6, send a routing advertisement (RA) message to all nodes in the network to indicate the method used to obtain the IPv6 address.

Procedure

1. If you use DHCPv6, configure Router Advertisement Daemon to allow the client to send the DHCP request.

   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;
        };
};
```

2. Create the dhcpd.conf file, dhcpv6.conf file, or both files in the /etc/ directory.

   The configuration file stores network information, such as the script path, ISO install file location, provisioning configuration file location, serial number, and chassis MAC address.

3. Add a host entry that uses the chassis MAC address.

   Example

```
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 DHCP host configuration is successful after the DHCP server is running.

4. If you identify the chassis by serial number, add a host entry that uses the chassis serial number.

   Example

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

   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;
    }
```

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

The DHCP server provides the iPXE boot image or provisioning configuration file information to Cisco NCS 1010.

Boot Cisco NCS 1010 using iPXE from the CLI

Configure iPXE boot from the CLI to reimage Cisco NCS 1010 by using the reload bootmedia network location all command.

Invoke iPXE boot from the CLI terminal to reimage the chassis.

Use this method to start the iPXE boot process from the CLI terminal.

Before you begin

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

Procedure

1. Run the command to invoke the iPXE boot process and 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]
```

2. Review the iPXE boot output.

   Example

```
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:
```

The iPXE boot process downloads the ISO image and displays the user access verification prompt.

Cisco NCS 1010 boots through iPXE from the CLI terminal and starts the reimage workflow.

Boot Cisco NCS 1010 using iPXE from BIOS

Configure iPXE boot from BIOS to reimage Cisco NCS 1010 by choosing the UEFI iPXE Network Boot option from the BIOS interface.

Invoke iPXE boot from the BIOS interface to reimage the chassis.

Use this method to start the iPXE boot process from the BIOS interface.

Before you begin

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

Procedure

---

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**.

Example

```
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:
```

The iPXE boot process downloads the ISO image and displays the user access verification prompt.

Cisco NCS 1010 boots through iPXE from the BIOS interface and starts the reimage workflow.

Install a new image without Golden ISO

Configure mini ISO boot, software package activation, and IOS XR configuration to install a new image without Golden ISO on NCS 1010.

Install a new image without using the Golden ISO feature.

Before the introduction of Golden ISO, you had to perform this sequence to install a new image.

Before you begin

- Ensure that the mini ISO is available.
- Ensure that all relevant SMUs, optional packages, and IOS XR configuration are available.

Procedure

1. Boot the system with mini ISO.

   You can use iPXE or USB boot.

2. Install, add, and activate all relevant SMUs and optional packages on NCS 1010.
   NCS 1010 reloads when any SMU reloads.
3. Apply IOS XR configuration.

The new image is installed without using Golden ISO.

Build a Golden ISO boot image for Cisco NCS 1010

Configure a Golden ISO boot image for Cisco NCS 1010 by using mini ISO, required SMUs, and IOS XR configuration in the gisobuild.py script.

Build a customized Golden ISO image that includes the mini ISO, required SMUs, and IOS XR configuration.

Golden ISO is a feature that enables you to build a customized ISO using mini ISO, required SMUs, and IOS XR configuration.

Golden ISO saves installation effort and time. It makes the system available in a single command and boot.

The `gisobuild.py` script is available at `/pkg/bin/gisobuild.py`.

> **Note**
>
> Install operation over IPv6 is not supported.

Before you begin

- For details about the image installation sequence used before Golden ISO was introduced, see Install a new image without Golden ISO.

  Copy the `/pkg/bin/gisobuild.py` script from NCS 1010 to the Linux environment.
- Ensure that the mini ISO, required SMUs, and IOS XR configuration file are available.

Procedure

1. Build the Golden ISO image.

   gisobuild.py -i *mini-iso* **-r** *rpm-directory* **-c** *xr-config* **-l** *label*

   - *rpm-directory* - Directory where SMUs (xr, calvados, and host) are copied.
   - *xr-config* - IOS XR configuration to be applied to the system after booting.

- *label* - Label of the Golden ISO.

Example

```
gisobuild.py -i./ncs1010-mini-x.iso -r ./rpm-directory -c ./xr-config -l label
```

2. Review the Golden ISO build output.

Example

```
python gisobuild.py -i ./ncs1010-mini-x-7.0.1.04I.iso -r. -c startup_new.cfg
 -l v2
System requirements check [PASS]
Golden ISO build process starting...

Platform: ncs1010 Version: 7.0.1.04I

XR-Config file (/bh/bosshogg_images/r701/701_04I_DT_IMAGE/giso/startup_new.cfg)
 will be encapsulated in Golden ISO.

Scanning repository [/bh/bosshogg_images/r701/701_04I_DT_IMAGE/giso]...

Building RPM Database...
Total 1 RPM(s) present in the repository path provided in CLI

XR x86_64 rpm(s) used for building Golden ISO:

(+) ncs1010-k9sec-192.0.2.1-r70104I.x86_64.rpm

...RPM compatibility check [PASS]

Building Golden ISO...
Summary .....

XR rpms:
ncs1010-k9sec-192.0.2.1-r70104I.x86_64.rpm

XR Config file:
router.cfg

...Golden ISO creation SUCCESS.

Golden ISO Image Location:
/bh/bosshogg_images/r701/701_04I_DT_IMAGE/giso/ncs1010-goldenk9-x-7.0.1.04I-v2.iso

Detail logs:
/bh/bosshogg_images/r701/701_04I_DT_IMAGE/giso/Giso_build.log-2019-03-20:15:47:19.516203
```

The command output shows that Golden ISO creation succeeded and displays the Golden ISO image location.

3. Verify the Golden ISO file format.

Use these Golden ISO filename formats:

- *platform-name*-golden-x.iso-*version.label* does not contain the security (*k9sec*.rpm) rpm.
- *platform-name*-goldenk9-x.iso-*version.label* contains the security (*k9sec*.rpm) rpm.

Example

Example 1: `ncs1010-golden-x-7.0.1.014I-V1.iso`

Example 2: `ncs1010-goldenk9-x-7.0.1.014I-V1.iso`

The Golden ISO boot image is built and its filename format is verified.

Network Time Protocol

Explains Network Time Protocol for Cisco NCS 1010, including clock synchronization behavior, stratum concepts, and dependencies for related setup procedures.

A Network Time Protocol implementation is a time synchronization function that

- uses UDP and Coordinated Universal Time to synchronize device clocks,
- forms configured associations with NTP servers to exchange timing messages, and
- supports accurate event timing for network management, security, planning, and debugging.

Details

Table 5: Feature History

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

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

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

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

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

Synchronize the clock with an NTP server

Synchronize the clock with an NTP server to align Cisco NCS 1010 system time before verification, troubleshooting, and post-setup activities.

Use this procedure to synchronize the clock with an NTP server.

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

Before you begin

Configure Management Interface

Procedure

1. Enter configuration mode.

   configure

   Example

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

   Enters the configuration mode.

2. Enter NTP configuration mode.

   ntp

   Example

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

   Enters NTP configuration mode.

3. Configure an NTP server.

   server 198.51.100.1 version 4 prefer iburst

   server 2001:DB8::1 version 4 prefer iburst

   Example
   IPv4:

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

   IPv6:

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

   Synchronizes the console clock with the specified NTP server.

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

4. Save or exit the configuration session.

   end

   commit

   - **end**
   - **commit**

   Example

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

   or

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

   Saves configuration changes.

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

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

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

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

5. Verify the running NTP configuration.

   show running-config ntp

   Example

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

   ```
   Sun Nov  5 15:14:24.969 UTC
   ```

```
ntp
 server 192.0.2.51 burst iburst
!
```

Displays the running configuration.

The synchronize the clock with an NTP server task is complete.

Verify NTP synchronization status

Verify NTP synchronization status to confirm the system clock state, association details, and synchronization health with the configured NTP server.

Use this procedure to verify NTP synchronization status.

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

Procedure

---

1. Verify NTP associations.

   show ntp associations

   Example

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

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

   selected, + candidate, - outlayer, x falseticker, ~ configured

   Displays the status of NTP associations.

2. Verify detailed NTP association information.

   show ntp associations detail

   Example

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

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

```
filtoffset = 7.857 7.802 8.065 8.063 8.332 8.397 8.664 8.684
filterror = 0.000 0.060 1.995 2.055 4.050 4.110 6.060 6.120
```

3. Verify detailed NTP association information for a location.

   show ntp associations detail location 0/RP0/CPU0

   Example

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

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

4. Verify NTP status.

   show ntp status

   Example

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

Clock is synchronized, stratum 3, reference is 192.0.2.1
nominal freq is 1000000000.0000 Hz, actual freq is 44881851.3383 Hz, precision
 is 2**24
reference time is E8F22D7A.AB020D97 (15:04:26.668 UTC Sun Nov 5 2023)
clock offset is 9.690 msec, root delay is 2.553 msec
root dispersion is 24.15 msec, peer dispersion is 2.13 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.0000212807
s/s
system poll interval is 128, last update was 610 sec ago
authenticate is disabled, panic handling is disabled,
hostname resolution retry interval is 1440 minutes.
```

   Verifies that the clock is synchronized with the NTP server.

The verify NTP synchronization status task is complete.

NTP troubleshooting reference for Cisco NCS 1010

Provides NTP troubleshooting resources for Cisco NCS 1010, including links to issue-resolution information that helps diagnose synchronization and clock status problems.

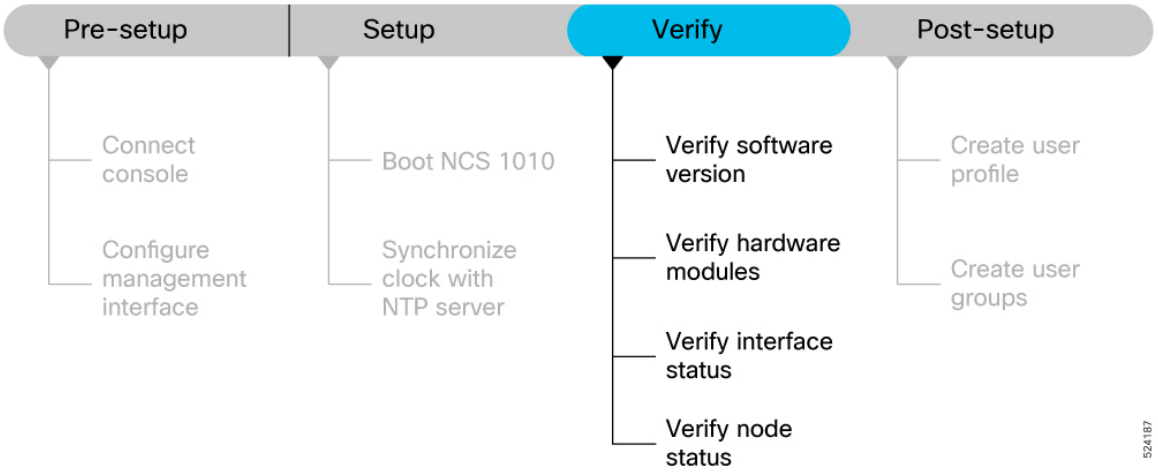
For details about NTP issue resolution, see Troubleshoot Network Time Protocol Issues.

Cisco NCS 1010 software and hardware verification

Provides the Cisco NCS 1010 software and hardware verification task sequence used to check the default setup after console login.

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

Figure 7: Verification Workflow for the Cisco NCS 1010 Setup



Complete the procedures in Cisco NCS 1010 setup workflow before you proceed with the verification tasks.

Note

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

Verify the software version

Verify the software version on Cisco NCS 1010 to confirm the installed release before hardware verification and post-setup task procedures begin.

Use this procedure to verify the software version.

View the software version installed on the NCS 1010.

Procedure

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

show version

Example

```
RP/0/RP0/CPU0:ios#show version
Sat Mar 25 11:38:23.614 IST
Cisco IOS XR Software, Version 24.3.1
Copyright (c) 2013-2023 by Cisco Systems, Inc.
```

```
Build Information:
Built By : ingunawa
Built On : Tue Mar 07 02:22:55 UTC 2023
Build Host : iox-ucs-063
Workspace : /auto/iox-ucs-063-san2/prod/203.0.113.1I.SIT_IMAGE/ncs1010/ws
Version : 24.3.1
Label : 24.3.1
cisco NCS1010 (C3758 @ 2.20GHz)
cisco NCS1010-SA (C3758 @ 2.20GHz) processor with 32GB of memory
OLT-C-R-SITE-1 uptime is 2 weeks, 12 hours, 59 minutes
NCS 1010 - Chassis
```

Note

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

For more information about upgrading the software version, see Software upgrade plan on page 101.

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

The verify the software version task is complete.

Verify hardware modules

Verify hardware modules on Cisco NCS 1010 to confirm module status and FPD information after the system completes setup checks.

Use this procedure to verify hardware modules.

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

Procedure

1. Verify the status of the hardware modules.

   show platform

   Example

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

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

2. View the list of hardware and firmware modules that are detected on the NCS 1010.

> **Note**
>
> From R26.2.1, the SSD FPD name displayed in the output of show hw-module fpd command is changed from a vendor or model-specific name to a generic format: 'CPU-SSD' for RP locations and 'CHASSIS-SSD' for rack locations.

show hw-module fpd command output until R26.2.1

Example

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


Mon Dec 22 10:01:56.338 UTC
Auto-upgrade:Enabled,PM excluded
Attribute codes: B golden, P protect, S secure, A Anti Theft aware
                                                                                FPD
 Versions

==============
Location   Card type            HWver FPD device       ATR Status    Running
  Programd   Reload Loc
-------------------------------------------------------------------------------
0/RP0/CPU0 NCS1010-CNT-B-K9      0.1   ADMCONFIG            CURRENT      1.00
     1.00        NOT REQ
0/RP0/CPU0 NCS1010-CNT-B-K9      0.1   BIOS             S   CURRENT      6.10
     6.10          0/RP0
0/RP0/CPU0 NCS1010-CNT-B-K9      0.1   BIOS-Golden       BS  CURRENT
      1.30          0/RP0
0/RP0/CPU0 NCS1010-CNT-B-K9      0.1   CpuFpga          S   CURRENT      1.12
     1.12          0/RP0
0/RP0/CPU0 NCS1010-CNT-B-K9      0.1   CpuFpgaGolden     BS  CURRENT
      0.07          0/RP0
0/RP0/CPU0 NCS1010-CNT-B-K9      0.1   SsdMicron5300     S   CURRENT      0.01
     0.01          0/RP0
0/RP0/CPU0 NCS1010-CNT-B-K9      0.1   TamFw            S   CURRENT      9.07
     9.07          0/RP0
0/RP0/CPU0 NCS1010-CNT-B-K9      0.1   TamFwGolden       BS  CURRENT
      9.05          0/RP0
0/PM0      NCS1K4-AC-PSU-2       1.0   PO-PriMCU             NOT READY
                   N/A
0/PM0      NCS1K4-AC-PSU-2       1.0   PO-SecMCU            CURRENT      1.05
     1.05        NOT REQ
0/PM1      NCS1K4-AC-PSU-2       1.0   PO-PriMCU            CURRENT      1.03
     1.03        NOT REQ
0/PM1      NCS1K4-AC-PSU-2       1.0   PO-SecMCU            CURRENT      1.05
     1.05        NOT REQ
0/0/NXR0   NCS1K-E-ILA-RE-C      0.1   ILA              S   CURRENT      3.44
```

```
     3.44         NOT REQ
0/0/NXR0   NCS1K-E-ILA-RE-C      0.1   Raman-E-1        S   CURRENT      4.04
     4.04         NOT REQ
0/1/NXR0   NCS1K-E2-OLT-RE-C     0.1   E2_OLT           S   CURRENT      4.01
     4.01         NOT REQ
0/1/NXR0   NCS1K-E2-OLT-RE-C     0.1   Raman-E-1        S   CURRENT      4.04
     4.04         NOT REQ
0/Rack     NCS1020-SA            0.1   ADMCONFIG            CURRENT      1.00
     1.00         NOT REQ
0/Rack     NCS1020-SA            0.1   IoFpgaLow        S   CURRENT      1.12
     1.12         NOT REQ
0/Rack     NCS1020-SA            0.1   IoFpgaLowGolden  BS  CURRENT
      0.07         NOT REQ
0/Rack     NCS1020-SA            0.1   IoFpgaUp         S   CURRENT      1.10
     1.10         NOT REQ
0/Rack     NCS1020-SA            0.1   IoFpgaUpGolden   BS  CURRENT
      0.06         NOT REQ
0/Rack     NCS1020-SA            0.1   SsdIntelSC2KB    S   CURRENT      1.30
     1.30          0/Rack
```

show hw-module fpd command output from R26.2.1

Example

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


Mon Dec 22 09:27:51.696 UTC
Auto-upgrade:Enabled,PM excluded
Attribute codes: B golden, P protect, S secure, A Anti Theft aware
                                                                                FPD
 Versions

==============
Location   Card type             HWver FPD device       ATR Status    Running
  Programd   Reload Loc
-----------------------------------------------------------------------------------
0/RP0/CPU0 NCS1010-CNT-B-K9      0.1   ADMCONFIG            CURRENT      1.00
     1.00         NOT REQ
0/RP0/CPU0 NCS1010-CNT-B-K9      0.1   BIOS             S   CURRENT      6.10
     6.10          0/RP0
0/RP0/CPU0 NCS1010-CNT-B-K9      0.1   BIOS-Golden      BS  CURRENT
      1.30          0/RP0
0/RP0/CPU0 NCS1010-CNT-B-K9      0.1   CPU-SSD          S   CURRENT      0.01
     0.01          0/RP0
0/RP0/CPU0 NCS1010-CNT-B-K9      0.1   CpuFpga          S   CURRENT      1.12
     1.12          0/RP0
0/RP0/CPU0 NCS1010-CNT-B-K9      0.1   CpuFpgaGolden    BS  CURRENT
      0.07          0/RP0
0/RP0/CPU0 NCS1010-CNT-B-K9      0.1   TamFw            S   CURRENT      9.07
     9.07          0/RP0
0/RP0/CPU0 NCS1010-CNT-B-K9      0.1   TamFwGolden      BS  CURRENT
      9.05          0/RP0
0/PM0      NCS1K4-AC-PSU-2       1.0   PO-PriMCU            NOT READY
                    N/A
0/PM0      NCS1K4-AC-PSU-2       1.0   PO-SecMCU            CURRENT      1.05
     1.05         NOT REQ
0/PM1      NCS1K4-AC-PSU-2       1.0   PO-PriMCU            CURRENT      1.03
     1.03         NOT REQ
0/PM1      NCS1K4-AC-PSU-2       1.0   PO-SecMCU            CURRENT      1.05
```

```
     1.05        NOT REQ
0/0/NXR0   NCS1K-E-ILA-RE-C      0.1   ILA              S   CURRENT      3.44
     3.44        NOT REQ
0/0/NXR0   NCS1K-E-ILA-RE-C      0.1   Raman-E-1        S   CURRENT      4.04
     4.04        NOT REQ
0/1/NXR0   NCS1K-E2-OLT-RE-C     0.1   E2_OLT           S   CURRENT      4.01
     4.01        NOT REQ
0/1/NXR0   NCS1K-E2-OLT-RE-C     0.1   Raman-E-1        S   CURRENT      4.04
     4.04        NOT REQ
0/Rack     NCS1020-SA            0.1   ADMCONFIG            CURRENT      1.00
     1.00        NOT REQ
0/Rack     NCS1020-SA            0.1   CHASSIS-SSD      S   CURRENT      1.30
     1.30         0/Rack
0/Rack     NCS1020-SA            0.1   IoFpgaLow        S   CURRENT      1.12
     1.12        NOT REQ
0/Rack     NCS1020-SA            0.1   IoFpgaLowGolden  BS  CURRENT
      0.07        NOT REQ
0/Rack     NCS1020-SA            0.1   IoFpgaUp         S   CURRENT      1.10
     1.10        NOT REQ
0/Rack     NCS1020-SA            0.1   IoFpgaUpGolden   BS  CURRENT
      0.06        NOT REQ
```

show hw-module fpd command output from R26.2.1 with another PSU vendor.

Example

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


Wed Jun  3 20:23:01.254 IST

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

==============
Location   Card type             HWver FPD device       ATR Status    Running
  Programd   Reload Loc
-----------------------------------------------------------------------------------
0/RP0/CPU0 NCS1010-CNTLR-K9      1.0   ADMConfig            CURRENT      3.40
     3.40        NOT REQ
0/RP0/CPU0 NCS1010-CNTLR-K9      1.0   BIOS             S   CURRENT      6.10
     6.10          0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.0   BIOS-Golden      BS  CURRENT
      4.40          0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.0   CPU-SSD          S   CURRENT     11.51
    11.51          0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.0   CpuFpga          S   CURRENT      1.16
     1.16          0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.0   CpuFpgaGolden    BS  CURRENT
      1.01          0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.0   TamFw            S   CURRENT      6.13
     6.13          0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.0   TamFwGolden      BS  CURRENT
      6.11          0/RP0
0/PM0      NCS1010-AC-PSU        1.0   LI-PriMCU            CURRENT      2.03
     2.03        NOT REQ
0/PM0      NCS1010-AC-PSU        1.0   LI-SecMCU            CURRENT      2.03
     2.03        NOT REQ
```

```
0/PM1      NCS1010-AC-PSU        1.0   LI-PriMCU            CURRENT      2.03
     2.03         NOT REQ
0/PM1      NCS1010-AC-PSU        1.0   LI-SecMCU            CURRENT      2.03
     2.03         NOT REQ
0/0/NXR0   NCS1K-ILA-C           0.1   ILA              S   CURRENT      3.48
     3.48         NOT REQ
0/Rack     NCS1010-SA            1.0   CHASSIS-SSD      S   CURRENT     11.51
    11.51          0/Rack
0/Rack     NCS1010-SA            1.0   EITU-ADMConfig       CURRENT      2.10
     2.10         NOT REQ
0/Rack     NCS1010-SA            1.0   IoFpga           S   CURRENT      1.30
     1.30         NOT REQ
0/Rack     NCS1010-SA            1.0   IoFpgaGolden     BS  CURRENT
      1.01         NOT REQ
```

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

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

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

Table 6: Status and Description of the Firmware Upgrade

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

3. Upgrade the required firmware as required,.

   upgrade hw-module location all fpd all

   Example

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

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

Note

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

4. Verify status of the modules after upgrade.

   show hw-module fpd REQ

   Example

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

 REQ
Wed Jun 29 08:50:21.057 UTC

Auto-upgrade:Disabled
                                                                                FPD
 Versions

==============
Location   Card type             HWver FPD device       ATR Status    Running
   Programd  Reload Loc
-------------------------------------------------------------------------------
0/RP0/CPU0 NCS1010-CNTLR-K9       1.0   ADMConfig            CURRENT    3.40
   3.40        NOT REQ
0/RP0/CPU0 NCS1010-CNTLR-K9       1.0   BIOS                  S  CURRENT    4.10
   4.10          0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9       1.0   BIOS-Golden          BS  CURRENT
   4.10          0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9       1.0   CpuFpga               S  CURRENT    1.02
   1.02          0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9       1.0   CpuFpgaGolden        BS  CURRENT
   1.01          0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9       1.0   SsdIntelS4510         S  CURRENT   11.32
  11.32          0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9       1.0   TamFw                 S  CURRENT    6.13
   6.13          0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9       1.0   TamFwGolden          BS  CURRENT
   6.11          0/RP0
```

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

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

5. Reload the individual nodes that require an upgrade.

   reload location node-location

   Example

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

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

   Example

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

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

==============
Location   Card type             HWver FPD device       ATR Status   Running
 Programd  Reload Loc
----------------------------------------------------------------------------------------
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  ADMConfig            CURRENT    3.40
   3.40       NOT REQ
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  BIOS              S  CURRENT    4.20
   4.20         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  BIOS-Golden      BS  CURRENT
   4.10         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  CpuFpga           S  CURRENT    1.11
   1.11         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  CpuFpgaGolden    BS  CURRENT
   1.01         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  SsdIntelS4510     S  CURRENT   11.32
  11.32         0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  TamFw             S  CURRENT    6.13
```

```
   6.13          0/RP0
0/RP0/CPU0 NCS1010-CNTLR-K9      1.11  TamFwGolden      BS  CURRENT
   6.11          0/RP0
0/PM0      NCS1010-AC-PSU        0.0   AP-PriMCU            CURRENT    1.03
   1.03       NOT REQ
0/PM0      NCS1010-AC-PSU        0.0   AP-SecMCU            CURRENT    2.01
   2.01       NOT REQ
0/PM1      NCS1010-AC-PSU        0.0   AP-PriMCU            CURRENT    1.03
   1.03       NOT REQ
0/PM1      NCS1010-AC-PSU        0.0   AP-SecMCU            CURRENT    2.01
   2.01       NOT REQ
0/0/NXR0   NCS1K-OLT-L           1.0   OLT              S   CURRENT    1.02
   1.02       NOT REQ
0/Rack     NCS1010-SA            2.1   EITU-ADMConfig       CURRENT    2.10
   2.10       NOT REQ
0/Rack     NCS1010-SA            2.1   IoFpga           S   CURRENT    1.12
   1.12       NOT REQ
0/Rack     NCS1010-SA            2.1   IoFpgaGolden     BS  CURRENT
   1.01       NOT REQ
0/Rack     NCS1010-SA            2.1   SsdIntelS4510    S   CURRENT   11.32
  11.32        0/Rack
```

The verify hardware modules task is complete.

Verify interface status

Verify interface status on Cisco NCS 1010 to confirm interface state and platform information after the setup workflow completes for management readiness.

Use this procedure to verify interface status.

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

Procedure

View the interfaces discovered by the system.

show ipv4 interfaces brief

Example

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

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

                Lo0       up          up              Loopback       1500
      0
                Lo3       up          up              Loopback       1500
      0
                Nu0       up          up               Null           1500
       0
           Gi0/0/0/0      up          up               ARPA           1514
```

```
1000000
        Mg0/RP0/CPU0/0    up          up                ARPA          1514
1000000
       Mg0/RP0/CPU0/1    admin-down  admin-down        ARPA  1514    1000000

       Mg0/RP0/CPU0/2    admin-down  admin-down        ARPA  1514    1000000

       PT0/RP0/CPU0/0    admin-down  admin-down        ARPA  1514    1000000

       PT0/RP0/CPU0/1    admin-down  admin-down        ARPA  1514    1000000
```

Example

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

Interface                  IP-Address      Status          Protocol Vrf-Name
Loopback0                  198.51.100.1       Up            Up       default

Loopback3                  203.0.113.1        Up            Up       default

GigabitEthernet0/0/0/0      192.0.2.1          Up            Up       default

MgmtEth0/RP0/CPU0/0         192.0.2.255        Up            Up       default

PTP0/RP0/CPU0/0             unassigned      Shutdown        Down     default

MgmtEth0/RP0/CPU0/1         unassigned      Down            Down     default

PTP0/RP0/CPU0/1             unassigned      Shutdown        Down     default

MgmtEth0/RP0/CPU0/2         unassigned      Down            Down     default
```

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

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

The verify interface status task is complete.

Verify node status

Verify node status on Cisco NCS 1010 to confirm active node state and readiness after the setup workflow completes successfully checks.

Use this procedure to verify node status.

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

Procedure

Verify the operational status of the node.

show platform

Example

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

Example

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

The verify node status task is complete.

What to do next

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

Verify hardware inventory

Verify hardware inventory on Cisco NCS 1010 to confirm installed components and inventory details after the system setup workflow completes successfully.

Use this procedure to verify hardware 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 this procedure.

Procedure

Verify hardware inventory.

show inventory

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

Example

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Example

```
RP/0/RP0/CPU0:ios#show inventory
Wed Jun  3 20:23:34.452 IST

NAME: "Rack 0", DESCR: "NCS 1010 Shelf Assembly"
PID: NCS1010-SA        , VID: V00, SN: FCB2546B08R

NAME: "0/RP0/CPU0", DESCR: "NCS1010 - Controller"
PID: NCS1010-CNTLR-K9  , VID: V00, SN: FCB2535B0AB
```

```
NAME: "0/FT0", DESCR: "NCS1010 - Shelf Fan"
PID: NCS1010-FAN       , VID: V00, SN: FCB2528B1M3

NAME: "0/FT1", DESCR: "NCS1010 - Shelf Fan"
PID: NCS1010-FAN       , VID: V00, SN: FCB2528B1LZ

NAME: "0/PM0", DESCR: "NCS 1010 - AC Power Supply Unit"
PID: NCS1010-AC-PSU    , VID: V00, SN: LIT2944334E

NAME: "0/PM1", DESCR: "NCS 1010 - AC Power Supply Unit"
PID: NCS1010-AC-PSU    , VID: V00, SN: LIT2944336S

NAME: "0/0/NXR0", DESCR: "NCS 1010 - In-Line Amplifier - C-band"
PID: NCS1K-ILA-C       , VID: V00, SN: FCB2533B0NB

NAME: "0/RP0-PTP0", DESCR: "Cisco Pluggable Optics Module"
PID: GLC-SX-MMD        , VID: V01, SN: OPM22160121

NAME: "0/RP0-PTP1", DESCR: "Cisco Pluggable Optics Module"
PID: GLC-SX-MMD        , VID: V01, SN: OPM221600ZM

NAME: "0/RP0-UDC0", DESCR: "Cisco Pluggable Optics Module"
PID: SFP-GE-S          , VID: V01, SN: FNS12281ZGA

NAME: "0/RP0-UDC1", DESCR: "Cisco Pluggable Optics Module"
PID: SFP-GE-S          , VID: V01, SN: FNS12510W8C
```

The verify hardware inventory task is complete.

Verify management interface status

Verify management interface status on Cisco NCS 1010 to confirm interface state, IPv4 address details, and management connectivity after setup completion.

Use this procedure to verify management interface status.

To verify the management interface status, perform this procedure.

Procedure

1. Verify the management interface configuration.

   show interfaces MgmtEth 0/RP0/CPU0/0

   Displays the management interface configuration.

   Example

```
RP/0/RP0/CPU0:ios#show interfaces MgmtEth 0/RP0/CPU0/0
Wed May 25 11:49:18.118 UTC
MgmtEth0/RP0/CPU0/0 is up, line protocol is up
  Interface state transitions: 1
  Hardware is Management Ethernet, address is 38fd.f866.0964 (bia
38fd.f866.0964)
  Internet address is 192.0.2.254/16
  MTU 1514 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 1000Mb/s, CX, link type is autonegotiation
```

```
  loopback not set,
  Last link flapped 15:05:21
  ARP type ARPA, ARP timeout 04:00:00
  Last input never, output 00:00:00
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     53138 packets input, 6636701 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 12145 broadcast packets, 40082 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     217288 packets output, 60964220 bytes, 0 total output drops
     Output 1 broadcast packets, 15 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions
```

2. Verify the management interface status.

   show interfaces summary

   show interfaces brief

   show ipv4 interfaces brief

   Verifies the management interface status.

   Example

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

   Example

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

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

                Lo0          up          up           Loopback  1500
  0
                Nu0          up          up               Null  1500
  0
         Gi0/0/0/0           up          up               ARPA  1514
100000
     Mg0/RP0/CPU0/0         up          up              ARPA  1514    1000000
```

```
     Mg0/RP0/CPU0/1          up          up              ARPA  1514    1000000
     Mg0/RP0/CPU0/2          up          up              ARPA  1514    1000000
     PT0/RP0/CPU0/0  admin-down  admin-down             ARPA  1514    1000000
     PT0/RP0/CPU0/1  admin-down  admin-down             ARPA  1514    1000000
    Mg0/RP0/RCOM0/0          up          up              ARPA  1514    1000000
```

Example

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

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

The verify management interface status task is complete.

Verify alarms

Verify alarms on Cisco NCS 1010 to confirm current alarm status and identify conditions that require attention after setup or recovery.

Use this procedure to verify alarms.

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

Procedure

Verify active alarms.

show alarms brief system active

Displays alarms in brief or detail.

Example

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

```
Thu Apr 28 06:16:50.524 UTC

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

Note

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

The verify alarms task is complete.

Verify environmental parameters

Verify environmental parameters on Cisco NCS 1010 to check fan, temperature, power, voltage, current, altitude, and overall environmental readiness status.

Use this procedure to verify environmental parameters.

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

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

Procedure

1. Verify fan environmental parameters.

show environment fan

Example

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

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

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

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

Displays the environmental parameters for the selected command output.

2. Verify route processor temperature parameters.

show environment temperature location 0/RP0

Example

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

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

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

3. Verify NXR temperature parameters.

   show environment temperature location 0/0/NXR0

   Example

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

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

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

4. Verify power environmental parameters.

   show environment power

Example

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

```
RP/0/RP0/CPU0:ios#show environment power
Thu May 26 04:17:55.592 UTC
===============================================================================
CHASSIS LEVEL POWER INFO: 0
===============================================================================

   Total output power capacity (Group 0 + Group 1) :    1050W +     1050W
   Total output power required                     :     700W
   Total power input                               :     228W
   Total power output                              :     140W

Power Group 0:
===============================================================================

   Power       Supply         ------Input----   ------Output---     Status
   Module      Type            Volts     Amps    Volts     Amps
===============================================================================

   0/PM0       NCS1010-AC-PSU  228.5     0.5     12.1      5.6      OK

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

Power Group 1:
===============================================================================

   Power       Supply         ------Input----   ------Output---     Status
   Module      Type            Volts     Amps    Volts     Amps
===============================================================================

   0/PM1       NCS1010-AC-PSU  228.5     0.5     12.1      6.1      OK

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

===============================================================================

   Location     Card Type                Power       Power        Status
                                         Allocated   Used
                                         Watts       Watts
===============================================================================

   0/RP0/CPU0   NCS1010-CNTLR-K9         90          14           ON
   0/FT0        NCS1010-FAN              110         17           ON
   0/FT1        NCS1010-FAN              110         15           ON
   0/0/NXR0     NCS1K-OLT-C              350         61           ON
   0/Rack       NCS1010-SA               40          19           ON


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

Example

```
RP/0/RP0/CPU0:ios#show environment power
Wed Jun  3 20:27:37.231 IST
===============================================================================
CHASSIS LEVEL POWER INFO: 0
===============================================================================
```

```
   Total output power capacity (Group 0 + Group 1) :    1050W +     1050W
   Total output power required                     :     511W
   Total power input                               :     136W
   Total power output                              :     118W

Power Group 0:
=============================================================================

   Power       Supply            ------Input----   ------Output---     Status

   Module      Type                Volts     Amps    Volts     Amps

=============================================================================

   0/PM0       NCS1010-AC-PSU     228.0     0.3     12.1      5.2      OK

Total of Group 0:               68W/0.3A         62W/5.2A

Power Group 1:
=============================================================================

   Power       Supply            ------Input----   ------Output---     Status

   Module      Type                Volts     Amps    Volts     Amps

=============================================================================

   0/PM1       NCS1010-AC-PSU     227.5     0.3     12.1      4.7      OK

Total of Group 1:               68W/0.3A         56W/4.7A

=============================================================================

   Location     Card Type                Power       Power        Status
                                         Allocated   Used
                                         Watts       Watts

=============================================================================

   0/RP0/CPU0   NCS1010-CNTLR-K9         111         10           ON
   0/FT0        NCS1010-FAN              110         12           ON
   0/FT1        NCS1010-FAN              110         13           ON
   0/0/NXR0     NCS1K-ILA-C              140         37           ON
   0/Rack       NCS1010-SA               40          12           ON
```

5. Verify voltage environmental parameters.

   show environment voltage location 0/RP0

   Example

   This 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

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

6. Verify current environmental parameters.

   show environment current

   Example

   This 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
```

7. Verify altitude environmental parameters.

   show environment altitude

   Example

   This 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
```

8. Verify all environmental parameters.

   show environment all

   Example

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

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

```
Tue Jul  5 08:37:28.412 UTC
================================================================================================
Location  TEMPERATURE                            Value     Crit    Major
Minor    Minor    Major    Crit
          Sensor                                (deg C)     (Lo)     (Lo)
(Lo)     (Hi)     (Hi)     (Hi)
------------------------------------------------------------------------------------------------
0/RP0/CPU0
          RP_TEMP_PCB                             29      -10       -5
 0       70       75       80
          RP_TEMP_HOT_SPOT                        32      -10       -5
 0       70       75       80
          RP_TEMP_LTM4638                         45      -10       -5
 0       80       85       90
          RP_TEMP_LTM4644_0                       35      -10       -5
 0       80       85       90
          RP_TEMP_LTM4644_1                       38      -10       -5
 0       80       85       90
          RP_JMAC_1V0_VCCP_TMON                   30      -10       -5
 0       80       85       90
          RP_JMAC_1V0_VNN_TMON                    29      -10       -5
 0       80       85       90
          RP_JMAC_1V0_VCC_RAM_TMON                30      -10       -5
 0       80       85       90
          RP_JMAC_1V2_DDR_VDDQ_TMON               31      -10       -5
 0       80       85       90
0/PM0
```

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

```
================================================================================
Location  VOLTAGE                                 Value      Crit     Minor
Minor     Crit
          Sensor                                   (mV)       (Lo)     (Lo)
(Hi)      (Hi)
--------------------------------------------------------------------------------
0/RP0/CPU0
          RP_ADM1266_12V0                       12094    10800    11280    12720
    13200
          RP_ADM1266_1V8_CPU                     1801     1670     1750
1850     1930
          RP_ADM1266_1V24_VCCREF                 1238     1150     1200
```

```
1280     1330
         RP_ADM1266_1V05_CPU                 1054      980     1020
1080     1120
         RP_ADM1266_1V2_DDR_VDDQ             1207     1120     1160
1240     1280
         RP_ADM1266_1V0_VCC_RAM               988      650      700
1250     1300
         RP_ADM1266_1V0_VNN                   858      550      600
1250     1300
         RP_ADM1266_1V0_VCCP                 1008      450      500
1250     1300
         RP_ADM1266_0V6_DDR_VTT               603      560      580
620      640
         RP_ADM1266_3V3_STAND_BY             3310     3070     3200
3400     3530
         RP_ADM1266_5V0                      4996     4650     4850
5150     5350
         RP_ADM1266_3V3                      3328     3070     3200
3400     3530
         RP_ADM1266_2V5_PLL                  2489     2330     2430
2580     2680
         RP_ADM1266_2V5_FPGA                 2500     2330     2430
2580     2680
         RP_ADM1266_1V2_FPGA                 1197     1120     1160
1240     1280
         RP_ADM1266_3V3_CPU                  3332     3070     3200
3400     3530
         RP_ADM1266_2V5_CPU                  2502     2330     2430
2580     2680
0/Rack
         SA_ADM1266_12V_BUS_EITU            12057    10800    11280    12720
    13200
         SA_ADM1266_5V0                      5022     4650     4800
5200     5350
         SA_ADM1266_1V8_ZARLINK_DPLL         1806     1670     1730
1870     1930
         SA_ADM1266_1V0_PHY                  1009      930      960
1040     1070
         SA_ADM1266_1V0_ALDRIN_CORE           982      910      930
1070     1090
         SA_ADM1266_1V0_ALDRIN_SERDES        1007      930      960
1040     1070
         SA_ADM1266_1V0_FPGA                 1008      930      960
1040     1070
         SA_ADM1266_1V2_FPGA                 1205     1120     1150
1250     1280
         SA_ADM1266_1V8                      1804     1670     1730
1870     1930
         SA_ADM1266_2V5                      2505     2330     2400
2600     2680
         SA_ADM1266_3V3                      3323     3070     3170
3430     3530
         SA_ADM1275_12V_SA_BP               12058    10800    11280    12720
    13200
         SA_ADM1275_12V_CPU_BP              12032    10800    11280    12720
    13200
         SA_ADM1275_12V_MOD0_BP             12063    10800    11280    12720
    13200
         SA_ADM1275_12V_MOD1_BP             12048    10800    11280    12720
    13200
         SA_ADM1275_12V_MOD2_BP             12027    10800    11280    12720
    13200
```

```
          SA_ADM1275_12V_FAN0_BP              12032    10800    11280    12720
    13200
          SA_ADM1275_12V_FAN1_BP              12042    10800    11280    12720
    13200
==============================================================================
Location  CURRENT                                 Value
          Sensor                                  (mA)
------------------------------------------------------------------------------
0/RP0/CPU0
          RP_CURRMON_LTM4638                       395
          RP_CURRMON_LTM4644_0                     179
          RP_CURRMON_LTM4644_1                     307
          RP_JMAC_1V0_VCCP_IMON                    125
          RP_JMAC_1V0_VNN_IMON                      62
          RP_JMAC_1V0_VCC_RAM_IMON                   0
          RP_JMAC_1V2_DDR_VDDQ_IMON                156
0/Rack
          SA_ADM1275_12V_MOD0_IMON                3412
          SA_ADM1275_12V_MOD1_IMON                  30
          SA_ADM1275_12V_MOD2_IMON                  43
          SA_ADM1275_12V_FAN0_IMON                1418
          SA_ADM1275_12V_FAN1_IMON                1394
          SA_INA230_5V0_IMON                       129
          SA_INA230_3V3_IMON                      3020
          SA_INA230_1V0_XGE_CORE_IMON             2464
          SA_INA230_1V0_FPGA_CORE_IMON             787
          SA_ADM1275_12V_SA_IMON                  1640
          SA_ADM1275_12V_CPU_IMON                 1157
==============================================================================
                                                Fan speed (rpm)
Location      FRU Type                          FAN_0    FAN_1   FAN_2
------------------------------------------------------------------------------

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

==============================================================================
CHASSIS LEVEL POWER INFO: 0
==============================================================================

   Total output power capacity (Group 0 + Group 1) :    1050W +        0W
   Total output power required                     :     700W
   Total power input                               :     159W
   Total power output                              :     129W

Power Group 0:
==============================================================================

   Power       Supply         ------Input----   ------Output---     Status
   Module      Type            Volts     Amps    Volts     Amps
==============================================================================

   0/PM1       NCS1010-AC-PSU  0.0       0.0     0.0       0.0      OFFLINE

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

Power Group 1:
```

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

   Power       Supply         ------Input----   ------Output---     Status
   Module      Type            Volts     Amps    Volts     Amps
==============================================================================

   0/PM0       NCS1010-AC-PSU  228.5     0.7     12.1      10.7     OK

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

==============================================================================

   Location     Card Type                Power       Power        Status
                                         Allocated   Used
                                         Watts       Watts
==============================================================================

   0/RP0/CPU0   NCS1010-CNTLR-K9         90          14           ON
   0/FT0        NCS1010-FAN              110         17           ON
   0/FT1        NCS1010-FAN              110         16           ON
   0/0/NXR0     NCS1K-ILA-C              350         54           ON
   0/Rack       NCS1010-SA               40          19           ON
```

The command output shows the fan, temperature, power, voltage, current, altitude, and overall environmental status.

Verify core dump context

Verify core dump context on Cisco NCS 1010 to check available context information before you collect or review core file details.

Use this procedure to verify core dump context.

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

Procedure

Verify the context.

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.

The verify core dump context task is complete.

Verify core files

Verify core files on Cisco NCS 1010 to confirm whether the system generated core files that require further analysis after setup verification.

Use this procedure to verify core files.

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

Procedure

Run the shell.

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
```

The verify core files task is complete.

Verify memory information

Verify memory information on Cisco NCS 1010 to confirm platform memory details and identify memory status after setup verification checks complete successfully.

Use this procedure to verify memory information.

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

Procedure

Verify memory information.

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
```

The verify memory information task is complete.

Cisco NCS 1010 post-setup tasks

Provides Cisco NCS 1010 post-setup task information, including AAA context and links to user profile and user group configuration tasks.

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

AAA services

Every user is authenticated using a username and a password.

The authentication, authorization, and accounting (AAA) commands help with these services:

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

User access behavior

IOS-XR and Linux have separate AAA services. IOS XR AAA is the primary AAA system.

- A user created through IOS-XR can log in directly to the EXEC prompt on the NCS 1010.
- A user created through Linux can connect to the NCS 1010 and log in to the bash prompt. The user must log in to IOS XR explicitly to access the IOS-XR EXEC prompt.

AAA authorization

Configure IOS-XR AAA authorization to restrict uncontrolled user access.

If AAA is not configured, the command rules and data rules that are associated with the assigned groups are ignored.

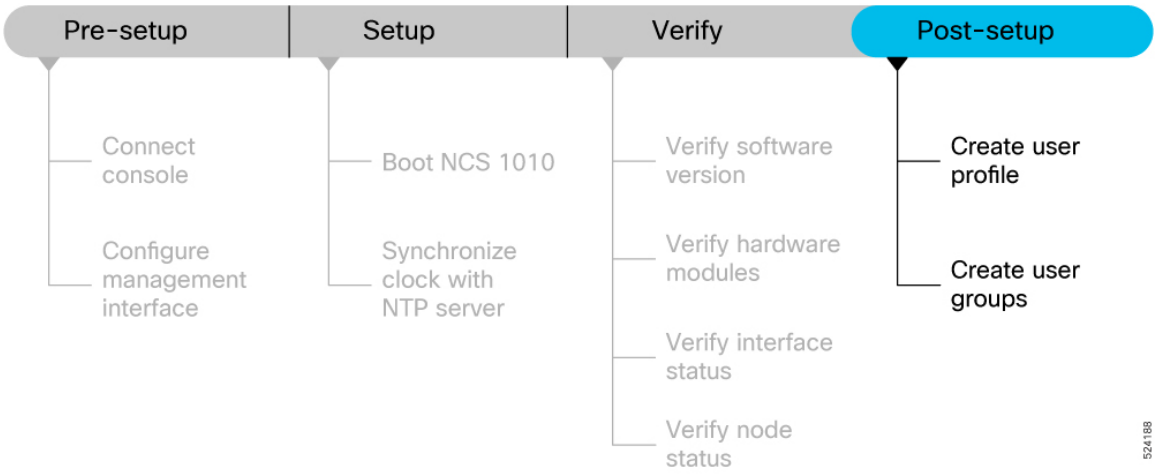
A user can then have full read and write access to IOS XR configuration through NETCONF, gRPC, or other YANG-based agents.

Enable AAA before you set up any configuration. For more information about AAA services, see AAA services.

Post-setup workflow

The image shows the tasks that are involved in the Cisco NCS 1010 Series NCS 1010 post-setup procedure.

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



Before you begin

Before you perform the post-setup tasks, complete these prerequisite tasks:

- Cisco NCS 1010 setup workflow
- Cisco NCS 1010 software and hardware verification tasks

Post-setup task functions

- Create a user profile: Create users and include the users in user groups with certain privileges.
- Create user groups: Associate command rules and data rules with a user group and enforce those rules on users in the group.

Create a user profile

Create a user profile to assign a password and group privileges for Cisco NCS 1010 access after initial setup and verification.

Use this procedure to create a user profile.

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

Create a user profile with these steps:

Procedure

1. Create a user, provide a password, and assign the user to a group.

   config

   username *<user-name>*

   password *password*

   group root-lr

   Example

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

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

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

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

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

   Enable display of login banner: The US Department of Defense (DOD)-approved login banner provides information such as number of successful and unsuccessful login attempts, time stamp, login method, and so on.

   The banner is displayed before granting access to devices and helps maintain privacy and security that is consistent with applicable federal laws.

   The system tracks logins from system boot or from the time the user profile is created.

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

> **Note**
>
> Login notifications get reset during a NCS 1010 reload.

2. Verify the state of login banner.

   show running-config username NAME1

   Example

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

3. Commit the configuration.

   commit

   Example

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

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

The create a user profile task is complete.

Create user groups

Create user groups to associate command rules and data rules with users who require specific Cisco NCS 1010 privilege levels.

Use this procedure to create user groups.

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

Before you begin

Ensure that you have created a user profile. See Create a user profile.

Procedure

---

1. Create a new user group.

   config

   group group1

   username user1

Example

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

/* Create a new user group, group1 */
ios#(config)#group group1

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

2. Commit the configuration.

   commit

   Example

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

The create user groups task is complete.

What to do next

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

4 TACACS+ and RADIUS Configuration

Topics:

- TACACS+ and RADIUS configuration details
- Deprecation of type 7 password and type 5 secret
- TACACS+ protocol
- Configure TACACS+ server
- Configure and verify TACACS+ server groups
- RADIUS protocol
- Configure and verify RADIUS server groups

This chapter describes TACACS+ and RADIUS configuration for Cisco NCS 1010. It helps readers use AAA concepts, RADIUS and TACACS+ references, and related configuration procedures.

TACACS+ and RADIUS configuration details

This section provides details about TACACS+ and RADIUS configuration, including context that helps readers use AAA concepts, RADIUS and TACACS+ references, and related procedures.

Use this reference to review configure AAA.

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

- **Note**

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

Deprecation of type 7 password and type 5 secret

This section provides details about the deprecation of type 7 password and type 5 secret for Cisco NCS 1010. Use it to review password storage behavior for AAA configuration.

Use this reference to review deprecation of type 7 password and type 5 secret.

Password configuration options before Release 24.4.1

- Until Release 24.4.1, there were two options for configuring a password:
  - Password: Uses Type 7 encryption to store the password.
  - Secret: Supports Type 5, 8, 9, or 10 hashing algorithms to store the password securely.

- **Deprecation notice**

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

  - #unique_68/unique_68_Connect_42_section_x5q_wm4_4dc
  - #unique_68/unique_68_Connect_42_section_msp_v44_4dc
  - #unique_68/unique_68_Connect_42_section_udk_2p4_4dc
  - #unique_68/unique_68_Connect_42_aaa-passwoed
  - #unique_68/unique_68_Connect_42_section
  - #unique_68/unique_68_Connect_42_masked-secret

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

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

Changes:

- All the options that were present until the Release 24.4.1are removed except LINE (to accept cleartext).
- **During upgrade**: Any configuration using the Type 7 password configuration is automatically converted to Type 10 secret.
- **Post-upgrade**: You can still use the Type 7 password configurations option after new commits, but the password will be stored as Type 10 secret.
- New **syslog** has been added to indicate the deprecation process:

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

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

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

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

```
username example
Cisco Confidential
secret 10
$6$P53pb/FFxNIT4b/.$yVakako4fp9PZiIYYh1xS0.W6o/yPrSyC8j4gIs6xli57iC1OryPXyN9y8yojRD2nhAWo9pjr/WAIhbXqp8st.
!
```

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

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

Changes:

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

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

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

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

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

```
username example
Cisco Confidential
secret 10
$6$P53pb/FFxNIT4b/.$yVakako4fp9PZiIYYh1xS0.W6b/yPrSyC8j4gIs6xli57iC1OryPXyN9y8yojRD2nhAWb9pjr/WAIhbXqp8st.
!
```

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

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

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

Changes:

- All the options that were present until 24.4.1are removed except LINE (to accept cleartext).
- **During upgrade**: Any configuration using the Type 7 password configuration is automatically converted to Type 10 secret.
- **Post-upgrade**: You can still use the password-policy configurations option after new commits, but the it will be stored as Type 10 secret.
- New **syslog** has been added to indicate the deprecation process:

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

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

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

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

```
username example
secret 10
$6$P53pb/FFxNIT4b/.$yVakako4fp9PZiIYYh1xS0.W6b/yPrSyC8j4gLs6xli57iC1OryPXyN9y8yojRD2nhAWb9pjr/WAIhbXqp8st.
!
!
```

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

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

Changes:

- The options min-char-change, restrict-password-advanced, and restrict-password-reverse that were present until the Release 24.4.1 are deprecated.
- **During upgrade**: These deprecated configurations do not go through any change during upgrade.
- **Post-upgrade**: These deprecated keywords do not take effect when configured post-upgrade.
- New **syslog** have been added to indicate the deprecation process:

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

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

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

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

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

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

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

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

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

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

Changes:

- The options 5 and encrypted are removed.
- **During upgrade**: Configurations using Type 5 secret will remain unchanged.
- **Post-upgrade**: Though the keyword 5 has been deprecated, you can still apply the existing configurations using Type 5 secret.
- New **syslog** has been added to indicate the deprecation process:

```
%SECURITY-LOCALD-2-DEPRECATED_SECRET_TYPE : Type 5 secret is deprecated.

Please use the 'secret' keyword with option type 10 for user.
```

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

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

show running configuration command output post-upgrade:

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

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

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

Changes:

- The options 5 and encrypted are removed.
- **During upgrade**: Configurations using masked-secret with Type 5 will remain unchanged.
- **Post-upgrade**: Though the keyword 5 has been deprecated, you can still apply the existing configurations using Type 5 masked secret.
- New **syslog** has been added to indicate the deprecation process:

```
%SECURITY-LOCALD-2-DEPRECATED_SECRET_TYPE : Type 5 secret is deprecated.

Please use the 'secret' keyword with option type 10 for user.
```

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

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

show running configuration command output post-upgrade:

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

- **Special use cases**
  - **Use case 1: Configurations using both Type 7 password and secret with 8, 9, or 10 hashing, for the same user**
    - **During upgrade**:
      - For the first 3000 username configurations, the password configuration will be rejected, and the secret configuration will remain unchanged.
      - For the rest of the username configurations, the original secret configuration will be rejected, and the password will be converted to Type 10 secret.
    - **Post-upgrade**:
      - For a new username configured, or the username that is already present before the upgrade, the password configuration will be rejected.
      - New **syslog** has been added to indicate the deprecation process:

```
%SECURITY-PSLIB-4-SECRET_CONFIG_PRESENT : The password configuration
 is deprecated.
Once secret is configured, cannot use password config for user <user
 name> at index <x> now.
```

- where 'x' is a number representing the index.

- **Use case 2: Configurations using both Type 7 password and Type 5 secret, for the same user**
  - **During upgrade**:
    - For any username configuration, the original Type 5 secret configuration will be rejected, and the password will be converted to Type 10 secret.
  - **Post-upgrade**:
    - For a new username configured, or the username that is already present before the upgrade, the password configuration will be converted to Type 10 secret.
    - New **syslog** has been added to indicate the deprecation process:

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

TACACS+ protocol

This section explains TACACS+ for Cisco NCS 1010. Use it to understand the feature behavior, requirements, and setup context before you configure or verify AAA, RADIUS, and TACACS+ configuration.

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

Details

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

Configure TACACS+ server

Use this procedure to configure a TACACS+ server. It supports AAA configuration by defining the required TACACS+ server settings and related authentication details.

Use this task to configure TACACS+ server.

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

To configure TACACS+ server, perform these steps:

Before you begin

Follow these steps to configure TACACS+ server.

Procedure

1. Enter into the IOS XR configuration mode.

   Example

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

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

   Example

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

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

   Example

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

Configure and verify TACACS+ server groups

Use this procedure to configure and verify TACACS+ server groups. Server groups let AAA method lists use selected external TACACS+ hosts for authentication, authorization, or accounting.

Use this task to complete the configuration and verification workflow for configure TACACS+ server groups.

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

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

To configure TACACS+ server groups, perform these steps:

Before you begin

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

Follow these steps to configure TACACS+ server groups.

Follow these steps to configure TACACS+ server groups.

Procedure

1. Configure the required server group settings.

   For details, see Configure TACACS+ server groups.

2. Verify the server group configuration.

   For details, see Verify TACACS+ server group configuration.

The configure TACACS+ server groups workflow is complete after the configuration and verification subtasks are complete.

Configure TACACS+ server group commands

Use this procedure to configure TACACS+ server group commands. It defines the server group, TACACS+ host, key, and default authentication method list.

Configure the settings required by Configure TACACS+ server groups.

This subtask contains the configuration command sequence from Configure TACACS+ server groups.

Before you begin

Follow these steps to configure TACACS+ server groups.

Procedure

1. Enter into the IOS XR configuration mode.

   Example

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

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

   Example

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

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

   Example

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

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

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

   Example

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

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

   Example

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

6. Repeat steps 3 to 5 for every private server to be added to the server group.
7. Configure certificate-based authentication for users configured in the TACACS+ server or server groups.

   Example

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

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

   Example

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

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

   commit

   end

The configuration commands for Configure TACACS+ server groups are applied.

Verify TACACS+ server group configuration

Use this procedure to verify TACACS+ server group configuration. The command output displays configured TACACS+ server groups and helps confirm that the expected group details are available.

Verify the configuration created by Configure TACACS+ server groups.

This subtask contains the verification command from Configure TACACS+ server groups.

Before you begin

Follow these steps to verify TACACS+ server group configuration.

Procedure

Verify the TACACS+ server group configuration details.

Example

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

The command output displays the configured server group details.

RADIUS protocol

This section explains RADIUS for Cisco NCS 1010. Use it to understand the feature behavior, requirements, and setup context before you configure or verify AAA, RADIUS, and TACACS+ configuration.

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

Details

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

Configure and verify RADIUS server groups

Use this procedure to configure and verify RADIUS server groups. Server groups let AAA method lists use selected external RADIUS hosts for authentication, authorization, or accounting.

Use this task to complete the configuration and verification workflow for configure RADIUS server groups.

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

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

Before you begin

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

Follow these steps to configure RADIUS server groups.

Follow these steps to configure RADIUS server groups.

Procedure

1. Configure the required server group settings.

   For details, see Configure RADIUS server groups.

2. Verify the server group configuration.

   For details, see Verify RADIUS server group configuration.

The configure RADIUS server groups workflow is complete after the configuration and verification subtasks are complete.

Configure RADIUS server group commands

Use this procedure to configure RADIUS server group commands. It defines the server group, RADIUS host, authentication and accounting ports, key, and default authentication method list.

Configure the settings required by Configure RADIUS server groups.

This subtask contains the configuration command sequence from Configure RADIUS server groups.

Before you begin

Follow these steps to configure RADIUS server groups.

Procedure

1. Run the **configure** command to enter global configuration mode.

   Example

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

   Enters mode.

2. Run the **aaa group server radius group-name** command to group different server hosts into distinct lists and enter server group configuration mode.

   Example

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

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

3. Run the **radius-server {ip-address}** command to specify the hostname or IP address of the RADIUS server host.

   Example

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

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

4. Run the **auth-port port-number** command to specify the User Datagram Protocol (UDP) destination port for authentication requests; the host is not used for authentication if set to 0. If unspecified, the port number defaults to 1645.

Example

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

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

5. Run the **acct-port port-number** command to specify the UDP destination port for accounting requests; the host is not used for accounting if set to 0. If unspecified, the port number defaults to 1646.

Example

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

Specifies the UDP destep_3_1204918stination port for accounting requests; the host is not used for accounting if set to 0. If unspecified, the port number defaults to 1646.

6. Run the **key string** command to specify the authentication and encryption key used between NCS 1010 and the RADIUS server.

Example

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

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

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

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

–

8. Run the **aaa authentication { login } { default } group group-name local** command to specify the default method list for authentication and enable authentication for console in global configuration mode.

Example

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

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

9. Run the **commit or end** command to commit the changes or exit configuration mode.

The configuration commands for Configure RADIUS server groups are applied.

Verify RADIUS server group configuration

Use this procedure to verify RADIUS server group configuration. The command output displays configured RADIUS server groups and helps confirm that the expected group details are available.

Verify the configuration created by Configure RADIUS server groups.

This subtask contains the verification command from Configure RADIUS server groups.

Before you begin

Follow these steps to verify RADIUS server group configuration.

Procedure

Run the **show radius server-groups** command to display information about each configured RADIUS server group.

Example

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

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

The command output displays the configured server group details.

5 Setup and Upgrade Workflow

Topics:

- NCS 1010 upgrade
- Supported upgrade and downgrade releases
- Software upgrade plan
- Back up the current configuration
- Field programmable devices
- System stability checks
- Install file sources
- Standard ISO and RPMs
- Software upgrade methods
- Data model software upgrade method
- Software upgrade verification

This chapter provides details about Cisco NCS 1010 setup and upgrade workflow. Use it to review the requirements, behavior, examples, or limitations that support software setup and upgrade.

Use this reference to review how Cisco NCS 1010 setup and software upgrade topics are organized.

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

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

NCS 1010 upgrade

This section provides details about NCS 1010 software upgrades, including context that helps readers use software setup and upgrade concepts, references, and procedures.

Use this reference to review how Cisco NCS 1010 setup and software upgrade topics are organized.

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

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

Supported upgrade and downgrade releases

This table lists supported upgrade and downgrade release paths for Cisco NCS 1010. Use it to confirm valid source and destination releases before planning software maintenance.

Use this reference to review view supported upgrade and downgrade releases.

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

- Downgrading from software release 7.11.1 or later to any earlier version is traffic-affecting and may result in a CMA process traceback.
  - When downgrading the software image from release 24.4.x to an earlier version, we recommend to manually downgrade the line card firmware as well to prevent any impact on various functionalities.
  - It is recommended to perform the upgrade with FPD auto-upgrade enabled to ensure the FPD versions are up to date and to prevent potential upgrade-related issues.
- The following table lists the upgrade and downgrade paths supported for Cisco NCS 1010.

Source Release	Destination Release	Bridge SMU	Source Release	Destination Release	Target SMU
2531	2541	NA	2541	2531	NA
2511	2541	CSCwn69606	2541	2511	NA
2431	2541	CSCwm77418	2541	2431	CSCwm77418
7112	2541	CSCwm77418,CSCwk57066	2541	7112	CSCwm77418
2541	2611	NA	2611	2541	NA
2531	2611	NA	2611	2531	NA
2511	2611	CSCwn69606	2611	2511	NA
2431	2611	CSCwm77418	2611	2431	CSCwm77418

Source Release	Destination Release	Bridge SMU	Source Release	Destination Release	Target SMU
7112	2611	CSCwm77418, CSCwk75706	2611	7112	CSCwm77418

Software upgrade plan

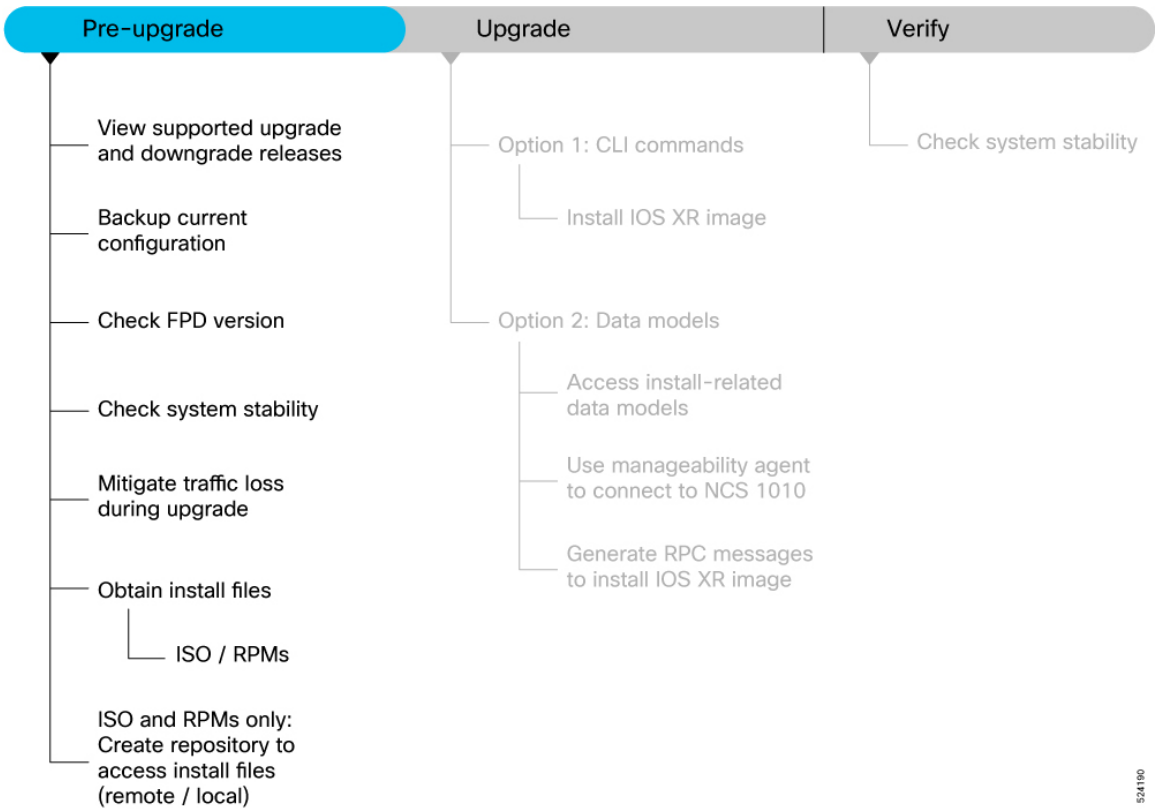
This section explains software upgrade planning for Cisco NCS 1010. Use it to understand preparation requirements before you upgrade the IOS XR software version.

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

Details

Pre-upgrade workflow image for NCS 1010

Figure 9: NCS 1010 Pre-upgrade workflow



Back up the current configuration

Use this procedure to back up the current configuration. It supports software setup and upgrade by preserving configuration data before upgrade or downgrade operations.

Use this task to backup current configuration.

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

Before you begin

Follow these steps to backup current configuration.

Procedure

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

   Example

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

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

   - Copy the configuration to the `harddisk:` location on NCS 1010.
   - Copy the configuration to a remote server. Ensure NCS 1010 has root access to the server.

2. Verify that the configuration is backed up.

Field programmable devices

This section explains Field Programmable Devices for Cisco NCS 1010. Use it to understand FPD behavior, hardware scope, and upgrade context before software maintenance.

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

Details
Table 7: 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.

Note

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

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

Note

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

Table 8: NCS 1010 FPDs

Location	FPDs
RP	• ADMConfig • CpuFpga • CpuFpgaGolden • BIOS • BIOS-Golden • SsdIntelS4510 • SsdMicron5300 • SsdSmartModular • TamFw/ TamFwGolden
PM0 and PM1	• AP-PrimMCU • AP-SecMCU
LC	• ILA • OLT • Raman-1 • Raman-2

Location	FPDs
Rack	• IoFpga • IoFpgaGolden • EITU-ADMConfig • SsdIntelS4510 • SsdMicron5300 • SsdSmartModular
Breakout module	• BRK-8 • BRK-24
Multiplexer and demultiplexer modules	• MD-32-ACC • MD-32-NEO

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

Check FPD Version

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

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

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

==============
Location   Card type             HWver FPD device       ATR Status   Running
Programd  Reload Loc
-----------------------------------------------------------------------------------
0/RP0/CPU0 NCS1010-CTLR-B-K9     1.0   ADMConfig            CURRENT    2.30
  2.30       NOT REQ
0/RP0/CPU0 NCS1010-CTLR-B-K9     1.0   BIOS             S   CURRENT    4.40
  4.40         0/RP0
0/RP0/CPU0 NCS1010-CTLR-B-K9     1.0   BIOS-Golden      BS  CURRENT
  4.40         0/RP0
0/RP0/CPU0 NCS1010-CTLR-B-K9     1.0   CpuFpga          S   CURRENT    1.11
  1.11         0/RP0
0/RP0/CPU0 NCS1010-CTLR-B-K9     1.0   CpuFpgaGolden    BS  CURRENT
  1.01         0/RP0
0/RP0/CPU0 NCS1010-CTLR-B-K9     1.0   SsdIntelS4510    S   CURRENT   11.32
 11.32         0/RP0
0/RP0/CPU0 NCS1010-CTLR-B-K9     1.0   TamFw            S   CURRENT    6.13
  6.13         0/RP0
0/RP0/CPU0 NCS1010-CTLR-B-K9     1.0   TamFwGolden      BS  CURRENT
  6.11         0/RP0
0/PM0      NCS1010-AC-PSU        0.0   AP-PriMCU            CURRENT    1.03
```

```
  1.03       NOT REQ
0/PM0      NCS1010-AC-PSU        0.0   AP-SecMCU            CURRENT    2.01
  2.01       NOT REQ
0/PM1      NCS1010-AC-PSU        0.0   AP-PriMCU            CURRENT    1.03
  1.03       NOT REQ
0/PM1      NCS1010-AC-PSU        0.0   AP-SecMCU            NEED UPGD  1.06
  1.06       NOT REQ
0/0/NXR0   NCS1K-E-OLT-R-C       1.0   OLT              S   CURRENT    1.16
  1.16       NOT REQ
0/0/NXR0   NCS1K-E-OLT-R-C       1.0   Raman-1          S   CURRENT    1.04
  1.04       NOT REQ
0/Rack     NCS1010-SA            0.1   EITU-ADMConfig       CURRENT    1.04
  1.04       NOT REQ
0/Rack     NCS1010-SA            0.1   IoFpga           S   CURRENT
  1.12       NOT REQ
0/Rack     NCS1010-SA            0.1   IoFpgaGolden     BS  NEED UPGD  1.12
  0.08       NOT REQ
0/Rack     NCS1010-SA            0.1   SsdIntelS4510    S   CURRENT   11.32
 11.32        0/Rack
0/1        NCS1K-MD-32E-C        0.1   MD-32-NEO        S   CURRENT    2.02
  2.02       NOT REQ
0/2        NCS1K-MD-32O-C        10.2  MD-32-ACC        S   CURRENT    2.18
  2.18       NOT REQ
0/3/0      NCS1K-BRK-8           1.0   BRK-8            S   CURRENT    2.08
  2.08       NOT REQ
0/3/3      NCS1K-BRK-24          1.0   BRK-24           S   CURRENT    2.08
  2.08       NOT REQ
```

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

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

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

Table 9: Description of Fields in show hw-module fpd Command

Field	Description
Location	Location of the FPD.
Card type	PID of the modules such as chassis, card, CPU, and PSU.
HWver	Hardware version where the FPD resides.
FPD device	Name of the FPD.
ATR	Attribute codes. The possible values are: • B - Golden Image • S - Secure Image • P - Protect Image The attribute code of the primary FPDs is S and the Golden FPDs is BS.
Status	Status of the FPD. See Table 10: Description of FPD Status Values in show hw-module fpd Command on page 106.

Field	Description
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 10: 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, • FPD version is up to date and compatible. • FPD image is protected.
UPGRADING	FPD upgrade started and the driver did not report the upgrade progress information yet.
%UPGD	Percentage of FPD upgrade completion.
RLOAD REQ	FPD upgrade is successfully completed and the FPD must be reloaded for the new version to take effect.
UPGD FAIL	FPD upgrade has failed. Check the syslog for failure reason. It could be a timeout or a failure that is reported by the driver.
UPGD DONE	FPD upgrade is successfully completed.

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

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

Table 11: Description of Fields in show fpd package Command

Field	Description
Card Type	PID of the modules such as chassis, card, CPU, and PSU.

Field	Description
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.

Upgrade FPDs automatically

Use this procedure to upgrade FPDs automatically. It helps update module FPD versions as part of the supported software maintenance workflow.

Use this task to 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.

Before you begin

Follow these steps to upgrade FPDs automatically.

Procedure

Run the following commands to disable automatic FPD upgrade.

Example

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

Upgrade FPDs manually

Use this procedure to upgrade FPDs manually. It helps update module FPD versions when the upgrade requires explicit manual control.

Use this task to upgrade FPDs manually.

Use the following procedure to upgrade the FPDs manually.

> **Note**
>
> The Golden FPDs cannot be upgraded using the CLI.

Before you begin

Follow these steps to upgrade FPDs manually.

Procedure

1. Run 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.

2. Run 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.

3. Run 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.

4. Run the **reload location location-id** command 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
   ```

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

FPD upgrades using YANG data models

This section provides details about upgrading FPDs by using YANG data models. Use it to review the programmatic method for FPD upgrade operations.

Use this reference to review upgrading FPDs using Yang data models.

The following information supports upgrading FPDs using Yang data models:

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

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

FPD downgrade scenarios

This section provides details about FPD downgrade scenarios for Cisco NCS 1010. Use it to review firmware behavior when moving from higher to lower releases.

Use this reference to review fPD downgrade scenarios.

When you downgrade from a higher release to lower release, the downgrade process of FPDs differs based on the higher and lower release firmware version.

- **Manual downgrade**
- When an FPD in the higher release runs a firmware that has an EVEN minor version, then the FPD does not downgrade automatically. The FPD requires manual downgrading using the force upgrade command[1] .
- For example, when you downgrade from R24.1.1 that has an FPD with an even firmware version 1.**10** to R7.11.1 that has the same FPD with firmware version 1.07, then the FPD does not downgrade automatically. The FPD must be manually downgraded using the force upgrade command .
- **Automatic downgrade**

[1] The force upgrade command is **upgrade hw-module location** *[location-id]* **fpd** *[fpd name]* **force**.

- When the change in the FPD firmware is in the major version, then the FPD automatically downgrades.
- For example. when you downgrade from R24.1.1 that has an FPD with firmware version **2**.10 to R7.11.1 that has the same FPD with a major difference in firmware version such as version **1**.07, then the FPD automatically downgrades.
- **Break release**
- When an FPD in the higher release runs a firmware that has an ODD minor version, then the FPD does not downgrade automatically. The FPD cannot be manually downgraded even with the force upgrade command .
- For example, when you downgrade from R24.1.1 that has an FPD with odd firmware version 1.**19** to R7.11.1 that has the same FPD with firmware version 1.07, then the FPD does not downgrade automatically. Also, the FPD cannot be forced to downgrade manually using the force upgrade command .
- **Minimum required firmware version**
- When an FPD in the lower release runs a firmware version that is less than the minimum version programmed in IDPROM of the FPD's respective cards, then the FPD does not downgrade automatically. The FPD cannot be manually downgraded even with the force upgrade command .
- For example, when you downgrade from R24.1.1 that has an FPD with firmware version 1.19 to R7.11.1 that has the same FPD with firmware version 1.07. If the minimum required firmware version that is programmed in IDPROM for that FPD is 1.09, then the FPD does not downgrade automatically. Also, the FPD cannot be forced to downgrade manually using the force upgrade command .

System stability checks

This section explains system stability checks for Cisco NCS 1010. Use it to understand software and hardware readiness before you continue with upgrade verification.

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

Details

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

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

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

`show install active summary`	Verify that the proper set of packages are active	NA
`show install committed summary`	Verify that the proper set of committed packages are same as active	Execute ‘install commit’ command

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

Install file sources

This section provides details about obtaining install files for Cisco NCS 1010. Use it to review base ISO, Golden ISO, and optional RPM download options.

Use this reference to review obtain install files.

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

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

Standard ISO and RPMs

Download install files from Cisco Software Center

Use this procedure to download install files from Cisco Software Center. It helps obtain the base ISO and RPM files required for the target software release.

Use this task to download install files from Cisco software center.

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

Before you begin

Follow these steps to download install files from Cisco software center.

Procedure

1. Access the Cisco Software Download page.

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

2. Click **Browse All** and navigate to NCS 1010 using **Optical Networking** > **Optical Data Center Interconnects** > **Network Convergence System 1000 Series** > **Network Convergence System 1010**.
3. Select the Software Type: IOS XR Software or IOS XR Software Maintenance Upgrades (SMU).
4. From the left pane, select the release.

   For the selected release, the Software Download page displays the downloadable files. For more information, see Install ISO and RPMs on page 113.

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

Software upgrade methods

This section provides details about upgrading the software for Cisco NCS 1010. Use it to review the processes involved in IOS XR software upgrade operations.

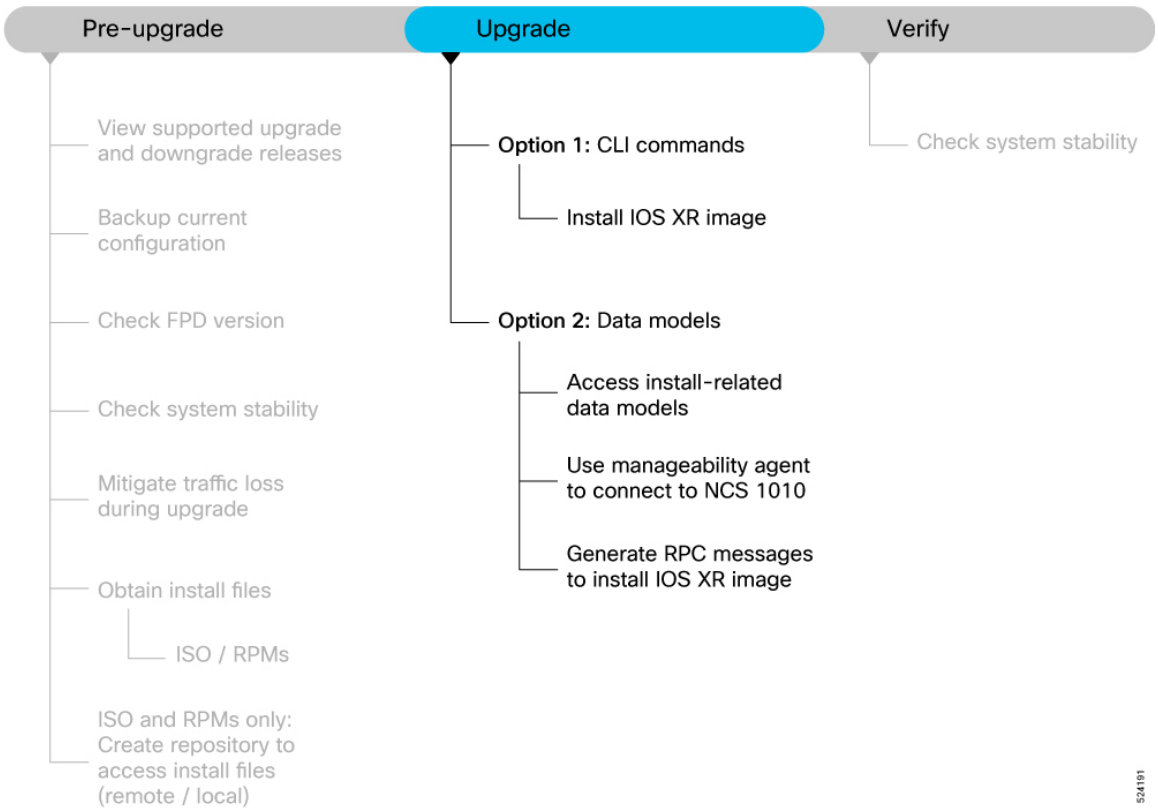
Use this reference to review upgrade the software.

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

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

> **Note**
>
> The NCS 1010 platform supports In-Service Software Upgrade (ISSU), and the software upgrade process is designed to be a non-impactful operation on network traffic.

Figure 10: NCS 1010 Upgrade workflow



CLI software upgrade method for Cisco NCS 1010

This section provides details about upgrading NCS 1010 using CLI commands. Use it to review the requirements, behavior, examples, or limitations that support software setup and upgrade.

Use this reference to review upgrade NCS 1010 using CLI commands.

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

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

Install ISO and RPMs

Use this procedure to install ISO and RPM files. It supports software upgrades by adding and activating the base image and feature packages on Cisco NCS 1010.

Use this task to install ISO and RPMs.

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

Before you begin

Follow these steps to install ISO and RPMs.

Procedure

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

Example

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

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

   Example

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

3. Install the base image to upgrade the system.

   Example

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

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

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

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

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

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

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

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

     Note

     Starting with upgrades to R2531 and later, this option is supported only if the ISO release is same as the running software release. This command is no longer supported for XR release upgrades. You may use the `install package replace` command to install optional RPMs or bug fixes.

     a. Install the image.

     b. Apply the changes.

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

c. Commit the operation.

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

> **Note**
>
> The **install commit** command must be executed immediately after software upgrades or SMU installations and before applying any new configuration changes or powering off the device to prevent loss of changes upon reboot.

4. (Optional) After the base image is upgraded, install the additional packages. For more information, see Install additional RPMs and bug fixes on page 150.

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

Install golden ISO

Use this procedure to install a Golden ISO. A Golden ISO applies a base image and selected bug fixes or optional packages through the supported install workflow.

Use this task to install golden ISO.

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

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

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

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

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

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

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

Note

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

Before you begin

Follow these steps to install golden ISO.

Procedure

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

   Example

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

2. Install the GISO.

   Example

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

   The

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

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

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

   ...
   ```

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

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

   ```
   RP/0/RP0/CPU0:ios#install apply
   Thu Feb 02 11:13:09.015
   Once the packaging dependencies have been determined, the install operation
   may have to reload the system.
   ```

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

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

- **Option 1:** Install GISO without control over reload timing.
  a. Install GISO to upgrade to a new release, add or remove bugfixes or optional packages.

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

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

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

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

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

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

  Note

  Starting with upgrades to R2531 and later, this option is supported only if the ISO release is same as the running software release. This command is no longer supported for XR release upgrades. You may use the `install package replace` command to install optional RPMs or bug fixes.

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

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

  b. Apply the changes.

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

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

Note

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

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

Apply the changes.

View the software version.

The

`GISO1`

custom label is replaced with the label

`24.3.1`

generated by the system.

Data model software upgrade method

This section provides details about upgrading by using data models for Cisco NCS 1010. Use it to review programmatic configuration and operational data workflows.

Use this reference to review upgrade using data models.

The following information supports upgrade using data models:

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

Access install-related data models

Use this procedure to access install-related data models. It supports software setup and upgrade by obtaining the YANG models used to manage install operations programmatically.

Use this task to access install-related data models.

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

`/pkg/yang`

directory.

Before you begin

Follow these steps to access install-related data models.

Procedure

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

   Example

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

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

   Example

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

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

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

Date Model	Description
Cisco-IOS-XR-shellutil-copy-act	Action model to copy files on the NCS 1010 from a source location.

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

Manageability agent connections to Cisco NCS 1010

This section explains using a manageability agent to connect to NCS 1010. Use it to understand NETCONF or gRPC communication before software maintenance operations.

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

Generate RPC messages to install an IOS XR image

Use this procedure to generate RPC messages to install an IOS XR image. It supports programmatic software installation through remote procedure call requests and responses.

Use this task to generate RPC messages to install IOS XR image.

This task supports Cisco NCS 1010 setup, deployment, upgrade, or maintenance workflows.

Before you begin

Not all software versions are supported as the target upgrade software version. You must review the supported upgrade and downgrade paths, hardware or software limitations, and bridging SMUs required for the version. For more information about checking the release support between the current and target versions, see Supported upgrade and downgrade releases on page 100.

Follow these steps to generate RPC messages to install IOS XR image.

Procedure

1. Invoke the **install-replace** RPC on the Cisco-IOS-XR-install-act.yang data model to upgrade NCS 1010.
2. Configure the values of the `source-type`, `source`, and `file` parameters.
3. Send `edit-config` NETCONF RPC request using the data model to configure the repository. Edit the values in the `repositories` parameters and send this request to NCS 1010 from the client.
4. Apply the changes to activate the ISO on NCS 1010 using RPCs by using the `install-apply` RPC on the`Cisco-IOS-XR-install-augmented-act.yang` datamodel and send the RPC from the client to NCS 1010.

Example

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

View the RPC response received from NCS 1010.

```
<?xml version="1.0"?>
  <rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">

   <op-id
xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-install-augmented-act">2.1</op-id>

  </rpc-reply>
```

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

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

Example

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

View the RPC response received from NCS 1010.

```
<?xml version="1.0"?>
  <rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">

   <data>
    <install xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-install-oper">
     <request>
      <request>install commit</request>
      <state>success</state>
     <timestamp>2022-06-27 T02:52:07Z</timestamp>
    <operation-id>26</operation-id>
   </request>
  </install>
```

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

What to do next

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

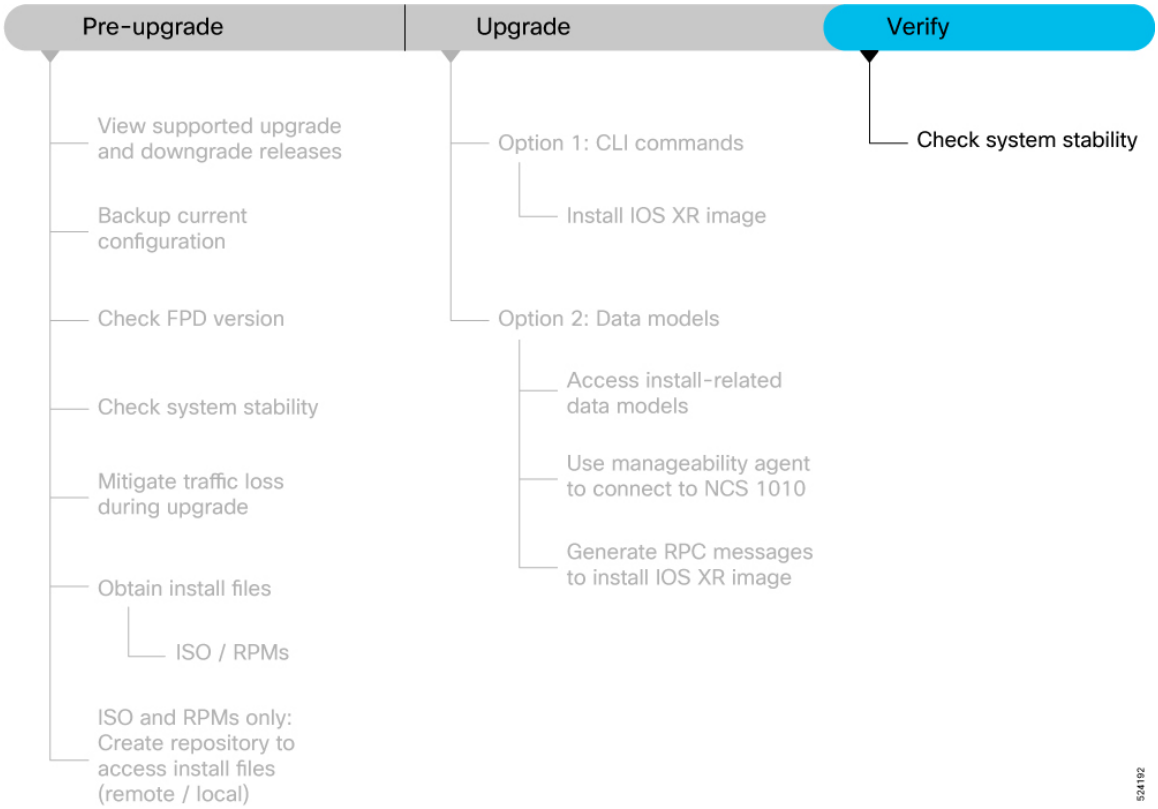
Software upgrade verification

This section provides details about verifying the software upgrade for Cisco NCS 1010. Use it to review checks for upgraded software and hardware component state.

Use this reference to review verify the software upgrade.

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

- **Figure 11: Workflow to Verify the Software Upgrade**



- This section contains the following topics:

System stability checks before upgrade

This section explains system stability checks for Cisco NCS 1010. Use it to understand software and hardware readiness before you continue with upgrade verification.

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

Details

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

Command	Reason	Workaround
show platform	Verify that all nodes are in `IOS XR RUN/OPERATIONAL` state	NA
show redundancy	Verify that a standby RP is available, and the system is in `NSR-ready` state	NA
show install active summary	Verify that the proper set of packages are active	NA

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

6 Classic ZTP Deployment

Topics:

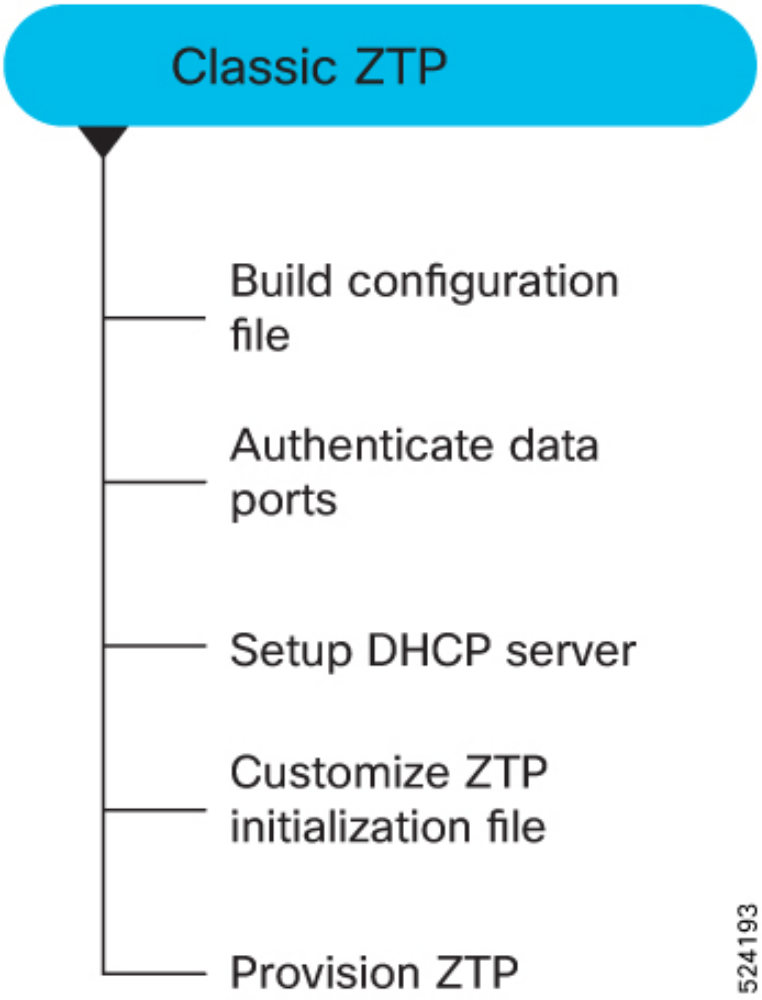
- DHCP configuration
- Provision Cisco NCS 1010 using DHCP-based ZTP fresh boot

This chapter provides details about classic ZTP deployment for Cisco NCS 1010. Use it to review DHCP setup, configuration files, ZTP options, logging, and manual invocation tasks.

Use this reference to review deploy NCS 1010 using classic ZTP.

Use this reference to review the classic ZTP workflow and the related provisioning topics for Cisco NCS 1010.

- **Figure 12: Classic ZTP Work Flow**



DHCP configuration

This section explains DHCP configuration for Cisco NCS 1010. Use it to understand the DHCP setup context required for manual configuration and classic ZTP deployment.

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.

DHCP relay

This section explains DHCP relay for Cisco NCS 1010. Use it to understand how relay agents forward DHCP packets between clients and servers on different subnets.

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.

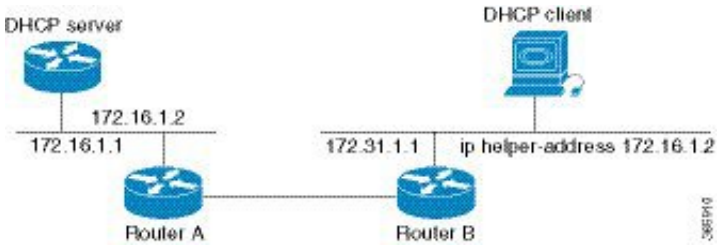
Details

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 13: Forwarding UDP Broadcasts to a DHCP Server Using a Helper Address



Prerequisites for configuring DHCP relay agent

This section provides details about prerequisites for configuring the DHCP relay agent. Use it to review user-group access, DHCP component roles, and server connectivity requirements.

Use this reference to review 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

This section provides details about DHCP relay feature limitations for Cisco NCS 1010. Use it to review supported helper-address behavior and option handling before configuration.

Use this reference to review 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.

DHCP relay agent configuration

This section provides details about configuring and enabling the DHCP relay agent for Cisco NCS 1010. Use it to review relay requirements, behavior, examples, and limitations.

Use this reference to review 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

This section explains the DHCP client for Cisco NCS 1010. Use it to understand dynamic address assignment, lease renewal behavior, and how DHCP responses reach the correct interface.

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.

Details

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
```

Provision Cisco NCS 1010 using DHCP-based ZTP fresh boot

Configure DHCP-based ZTP fresh boot for Cisco NCS 1010 when the device has no prior configuration. The workflow uses DHCP to locate provisioning data, downloads it from the web server, and applies the configuration or script.

Use this supertask to provision Cisco NCS 1010 automatically during a fresh boot using DHCP-based classic ZTP.

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

The high-level workflow uses DHCP to identify the device, obtain the URL for a configuration file or user script, download the file from a web server, and apply the provisioning data.

Before you begin

- Ensure that the Cisco NCS 1010 device has no prior configuration.
- Ensure that the DHCP server can identify the device and return the configuration file or script location.
- Ensure that the web server hosts the configuration file, script file, or binary referenced in the DHCP response.

Follow these steps to provision Cisco NCS 1010 using DHCP-based ZTP fresh boot.

Procedure

1. Create the configuration file or script file that the HTTP server provides to the device.

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

   > **Note**
   >
   > - If the downloaded file content starts with !! IOS XR, ZTP treats it as a configuration file.
   > - If the downloaded file content starts with #! /bin/bash, #! /bin/sh, or #!/usr/bin/python, ZTP treats it as a script file.

   For configuration file requirements, see Configuration file requirements.

   For ZTP script configuration, see Configure a ZTP bootscript.

2. Configure the DHCP server to identify the device and return the provisioning file location.

   The DHCP server identifies the device and responds with one of the following options:

   - DHCPv4 using BOOTP filename to supply the script or configuration location.
   - DHCPv4 using Option 67 (bootfile-name) to supply the script or configuration location.
   - DHCPv6 using Option 59 (OPT_BOOTFILE_URL) to supply the script or configuration location.

   The DHCP response provides the URL location that the network device uses to download the file from the web server.

   For details about DHCP server options, see DHCP server setup for ZTP.

3. Customize the ZTP DHCP request sequence if the default sequence does not meet the deployment requirements.

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

   - ZTP sends IPv4 DHCP requests first on all management ports. If the request fails, ZTP sends IPv6 DHCP requests on all management ports.
   - ZTP sends IPv4 DHCP requests first on all data ports. If the request fails, ZTP sends IPv6 DHCP requests on all data ports.

   For details about configurable ZTP options, see ZTP initialization file options.

4. Boot the unconfigured device with the IOS XR image.

   ZTP starts automatically only when the network device does not have a prior configuration.

The device starts the ZTP fresh boot workflow.

5. Verify that ZTP sends DHCP requests and receives the DHCP response.

   To help the bootstrap server uniquely identify the device, ZTP sends the following DHCP options:

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

   For details about the DHCP client, see DHCP client.

6. Verify that the device downloads the provisioning file from the web server.

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

7. Verify that the device applies the configuration file or executes the script or binary in the default bash shell.

   If you need to run ZTP manually after the initial boot, see Invoke ZTP manually through CLI or Invoke ZTP through reload.

8. Verify that the network device is up and running after ZTP completes.

   For ZTP log details, see ZTP logs.

   For troubleshooting data collection, see Generate tech support information for ZTP.

   The network device is up and running.

Cisco NCS 1010 completes DHCP-based ZTP fresh boot and applies the downloaded configuration file or script.

Configuration file requirements

This section provides details about ZTP configuration files for Cisco NCS 1010. Use it to review the required file marker, sample IOS XR configuration, and configuration-file guidance.

Use this reference to review build your configuration file.

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

- The configuration file content starts with !! IOS XR.
- The following is the sample configuration file. You can automate all the configurations. For more information on creating ZTP configuration file, refer ZTP Configuration Files Creation.

```
Tue May 4 18:08:59.544 UTC
Building configuration...
!! IOS XR Configuration 192.0.2.254I
!! Last configuration change at Tue May 4 17:12:47 2021 by cisco
!
line console
exec-timeout 0 0
!
line default
exec-timeout 0 0
session-timeout 0
!
vty-pool default 0 20
alias alarms show alarms brief system active
interface MgmtEth0/RP0/CPU0/0
ipv4 address dhcp
no shut
!
interface MgmtEth0/RP0/CPU0/1
description noshut-interface-ztp
ipv4 address 192.0.2.255 255.255.255.0
no shut
!
interface MgmtEth0/RP0/CPU0/2
description noshut-interface-ztp
no shut
!
interface PTP0/RP0/CPU0/0
description noshut-interface-ztp
no shut
Cisco NCS 1010 System Setup and Software Installation Guide, IOS XR Release
 7.7.x
19
Bring-up Cisco NCS 1010
Build your Configuration File
!
telnet vrf default ipv4 server max-servers 100a
ssh server v2
ssh server netconf vrf default
netconf-yang agent
ssh
!
netconf agent tty
grpc
ncs1010 static
address-family ipv4 unicast
0.0.0.0/0 192.0.2.255
end
```

Configure a ZTP bootscript

Use this procedure to configure a ZTP bootscript. The bootscript lets Cisco NCS 1010 run a bash script or configuration file during ZTP processing.

Use this task to 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.

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

Before you begin

Follow these steps to configure ZTP bootscript.

Procedure

1. Configure the ZTP bootscript to run on every boot.

   Example

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

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

2. Configure the ZTP bootscript to run before interface IP address assignment.

   Example

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

   The standard configuration waits for the first data-plane interface and then waits an extra minute for the management interface to receive an IP address. Use the preip option if the delay is not required.

3. Review the /disk0:/myscript content example.

   Example

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

   This example shows the content of /disk0:/myscript.

4. Review the ZTP bash script example.

Example

```
#! /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"`
```

This example shows sample content for the ZTP bash script.

5. Review the ZTP configuration file example.

Example

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

This example shows sample content for the ZTP configuration file.

The ZTP bootscript configuration and examples are available for use during ZTP.

Invoke ZTP manually through CLI

Use this procedure to invoke ZTP manually from the CLI. Manual invocation lets you clean ZTP state, start a ZTP DHCP session, or terminate an active ZTP process during testing.

Use this task to invoke ZTP manually through CLI.

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

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

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

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

Before you begin

Follow these steps to invoke ZTP manually through CLI.

Procedure

1. (Optional) Run the **ztp clean** command to remove all ZTP logs and saved settings.

   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.
   ```

   Example:

   Removes all the ZTP logs and saved settings.

2. Run the **ztp initiate** command to invoke a new ZTP DHCP session.

   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#
```

Example:

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

Reboots the Cisco NCS 1010 system.

3. (Optional) Run the **ztp terminate** command to terminate the ZTP process.

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
```

Example:

Terminates the ZTP process.

Invoke ZTP through reload

Use this procedure to invoke ZTP through reload. It restarts the system so Cisco NCS 1010 can run the ZTP workflow again during provisioning.

Use this task to invoke ZTP through reload.

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

Before you begin

Follow these steps to invoke ZTP through reload.

Procedure

1. Run the **configure** command to enter configuration mode.

Example

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

Example:

Enters the configuration mode.

2. Run the **commit replace** command to remove the entire running configuration.

Example

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

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

Warning

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

Example:

Removes the entire running configuration.

3. Run the **ztp clean** command to remove all ZTP logs and saved settings.

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.
```

Example:

Removes all the ZTP logs and saved settings.

4. Run the **reload** command to reboot Cisco NCS 1010 and invoke ZTP.

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]
```

```
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
```

Example:

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

Reboots the Cisco NCS 1010 system.

Data port authentication

This section provides details about data port authentication for Cisco NCS 1010. Use it to review how ZTP validates data ports before switching from management ports.

Use this reference to review authenticate data ports.

The following information supports authenticate data ports:

- On fresh boot, ZTP process is initiated from management ports and may switch to data ports. To validate the connection with DHCP server, authentication is performed on data ports through DHCP option 43 for IPv4 and option 17 for IPv6. These DHCP options are defined in option space and are included within **dhcpd.conf** and **dhcpd6.conf** configuration files. You must provide following parameters for authentication while defining option space:
  - Authentication code–The authentication code is either 0 or 1; where 0 indicates that authentication is not required, and 1 indicates that MD5 checksum is required.

    Note

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

  - Client identifier–The client identifier must be 'exr-config'.

- MD5 checksum–This is chassis serial number. It can be obtained using **echo -n $SERIALNUMBER | md5sum | awk '{print $1}'** .

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

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

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

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

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

 }
 else {
   #option dhcp6.bootfile-url
"http://[2001:1851:c632:1::1]/cisco-2/cisco-mini-x.iso.sh";
   option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/cisco-2/ztp.cfg";
```

```
   }
}
```

DHCP server setup for ZTP

This section provides details about DHCP server setup for Cisco NCS 1010. Use it to review DHCP pointers to software images, configuration files, and scripts for ZTP.

Use this reference to review setup DHCP server.

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

- The DHCP request from the NCS 1010 has the following DHCP options to identify itself:
  - **Option 60**: “vendor-class-identifier” : Used to Identify the following four elements:
    - The type of client: For example, PXEClient
    - The architecture of The system (Arch): For example: 00009 Identify an EFI system using a x86-64 CPU
    - The Universal Network Driver Interface (UNDI):
    - For example 003010 (first 3 octets identify the major version and last 3 octets identify the minor version)
    - The Product Identifier (PID):
  - **Option 61**: “dhcp-client-identifier” : Used to identify the Serial Number of the device.
  - **Option 66** : Used to request the TFTP server name.
  - **Option 67**: Used request the TFTP filename.
  - **Option 97**: “uuid” : Used to identify the Universally Unique Identifier a 128-bit value (not usable at this time)
- **Example**
- The following DHCP request sample provides a fixed IP address and a configuration file with the mac address of the management interface.

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

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

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

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

The DHCP server responds with the following DHCP options:

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

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

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

######### Network 11.11.11.0/24 ################
shared-network 11-11-11-0 {

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

    }

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

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


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

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


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


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

ZTP initialization file options

This section provides details about customizing the ZTP initialization file for Cisco NCS 1010. Use it to review configurable options and related classic ZTP behavior.

Use this reference to review customize ZTP initialization file.

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

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

  Note

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

  By default, the USB port has the higher priority.

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

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

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

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

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

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

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

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

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

2023-03-14 12:51:29,251 53612 [Configuration] INF: Provisioning via config
replace
2023-03-14 12:51:43,131 53612 [Configuration] INF: Configuration has been
applied
2023-03-14 12:51:43,131 53612 [Env ] DEB: cfg::createRefOnConfigCommit: called
2023-03-14 12:51:44,218 53612 [Env ] DEB: cfg:: Generating hash for File
name: /disk0:/ztp/customer/config.inithash_tmp
2023-03-14 12:51:44,218 53612 [Env ] DEB: cfg::_generateCfgAndSaveHash:: HASH
 : c7980cfc23a401bbbf296e3d49c76bf9, type : 1
2023-03-14 12:51:59,715 53612 [Env ] DEB: cfg:: Generating hash for File
name: /disk0:/ztp/customer/config.successhash_tmp
```

```
..........
2023-03-14 12:51:59,715 53612 [Env ] DEB: cfg::_generateCfgAndSaveHash:: HASH
 : c7980cfc23a401bbbf296e3d49c76bf9, type : 2
2023-03-14 12:52:04,901 53612 [Env ] DEB: cfg::getRefOnSuccess :: called
..........
2023-03-14 12:52:05,403 53612 [Engine ] INF: ZAdmin, current state:active,
exit code:success
2023-03-14 12:52:05,403 53612 [Engine ] INF: ZAdmin, current state:final,
exit code:success: state changed to final
```

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

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

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

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

[Fetcher Priority]
USB: 0

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

- **Enable ZTP Using CLI**
- If you want to enable ZTP using CLI, use the **ztp enable** command.
- **Configuration example**

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

- **Disable ZTP Using CLI**
- If you want to disable ZTP using CLI, use the **ztp disable** command.
- **Configuration example**

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

How classic ZTP provisioning works

This section explains how classic ZTP provisioning works for Cisco NCS 1010. Use it to understand the components, sequence, and result before you configure or troubleshoot classic ZTP deployment.

This process supports Cisco NCS 1010 setup, deployment, or maintenance activities.

Summary

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

Workflow

The classic ZTP provisioning ZTP involves the following stages:

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

   The following is the default sequential flow of the ZTP process:
   - ZTP sends IPv4 DHCP request first on all the management port. In case there is a failure, then ZTP sends IPv6 DHCP request on all the management port.
   - ZTP sends IPv4 DHCP request first on all the data port. In case there is a failure, then ZTP sends IPv6 DHCP request on all the data port.

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

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

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

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

ZTP logs

This section explains ZTP logging for Cisco NCS 1010. Use it to understand the feature behavior, requirements, and setup context before you configure or verify classic ZTP deployment.

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.

Details

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

```
2022-06-17 11:52:34,682  19292 [Xr           ] INF: Downloading the file to
/tmp/ztp.script
2022-06-17 11:52:35,329  19292 [Report       ] INF: User script downloaded
successfully. Provisioning in progress.
2022-06-17 11:52:35,330  19292 [Engine       ] DEB: ZAdmin, current
state:active. Processing work: Config device work for ZAdmin. done = False
2022-06-17 11:52:35,330  19292 [ZAdmin       ] DEB: Proceeding to provision
the NCS 1010
2022-06-17 11:52:35,331  19292 [Engine       ] DEB: ZAdmin, current
state:active. Processing work: ZAdmin: Apply configuration. done = False
2022-06-17 11:52:35,331  19292 [Engine       ] INF: ZAdmin, current
state:active: state tag changed to provision
RP/0/RP0/CPU0:Jun 17 11:52:35.341 UTC: pyztp2[140]:
%INFRA-ZTP-4-CONFIG_INITIATED : ZTP has initiated config load and commit
operations
2022-06-17 11:52:35,339  19292 [Env          ] DEB: No MTU configs detected
2022-06-17 11:52:35,340  19292 [Engine       ] DEB: ZAdmin, current
state:active. Processing work: ZAdmin: Apply configuration. done = False
2022-06-17 11:52:35,354  19292 [Xr           ] DEB: Will apply the following
config: /disk0:/ztp/customer/config.candidate
2022-06-17 11:52:35,354  19292 [Xr          ] INF: Applying user configurations
2022-06-17 11:52:35,355  19292 [Configuration] INF: Provisioning via config
replace
2022-06-17 11:52:54,656  19292 [Configuration] INF: Configuration has been
applied
2022-06-17 11:52:54,656  19292 [Engine       ] DEB: ZAdmin, current
state:active. Processing work: Sending standby sync message. done = False
2022-06-17 11:52:54,663  19292 [Engine       ] DEB: ZAdmin, current
state:active. Processing work: [privileged] getting engine status. done = False
2022-06-17 11:52:54,664  19292 [Engine       ] DEB: ZAdmin, current
state:active. Processing work: ZAdmin: Execute post-configuration script. done
```

```
 = False
2022-06-17 11:52:55,212  19292 [Env         ] INF: Env::cleanup, success:True,
 exiting:False
2022-06-17 11:52:55,213  19292 [ZtpHelpers   ] DEB: Executing: source
/pkg/bin/ztp_helper.sh && echo -ne  | xrcmd "show running-config"
2022-06-17 11:52:55,825  19292 [Env          ] INF: Executing command ip netns
 exec vrf-default /sbin/dhclient -4 -cf /etc/dhcp/dhclient.conf.ztp -lf
/var/lib/dhcp/dhclient.leases.ztp -sf /etc/dhcp/dhclient-script.ztp2 -r
Mg0_RP0_CPU0_0 to release IP
2022-06-17 11:52:56,968  19292 [Xr           ] INF: Removing linux route with
 ip 203.0.113.254
2022-06-17 11:52:57,023  19292 [Engine       ] INF: ZAdmin, current
state:active, exit code:success
2022-06-17 11:52:57,023  19292 [Engine      ] INF: ZAdmin, current state:final,
 exit code:success: state changed to final
2022-06-17 11:52:59,737  19292 [Engine      ] DEB: ZAdmin, current state:final,
 exit code:success. Processing work: Sending standby sync message. done = False
2022-06-17 11:52:59,738  19292 [Engine      ] WAR: ZAdmin, current state:final,
 exit code:success: work is ignored: work=<desc='Sending standby sync message'
 done=False priv=False>
2022-06-17 11:52:59,738  19292 [Engine      ] DEB: ZAdmin, current state:final,
 exit code:success. Processing work: [privileged] getting engine status. done
 = False
2022-06-17 11:53:04,744  19292 [main     ] DEB: Moved to final state
2022-06-17 11:53:04,745  19292 [main     ] DEB: ZTP completed successfully
2022-06-17 11:53:04,745  19292 [main     ] INF: Exiting SUCCESSFULLY
2022-06-17 11:53:04,746  19292 [main     ] DEB: Exiting. Will not retry now.
2022-06-17 11:53:04,746  19292 [main    ] DEB: Shutting down adaptor. Cleanup
 False. Exiting False
2022-06-17 11:53:04,748  19292 [Engine      ] DEB: ZAdmin, current state:final,
 exit code:success. Processing work: [privileged] prepare engine shutdown.
done = False
2022-06-17 11:53:04,849  19292 [Engine      ] DEB: ZAdmin, current state:final,
 exit code:success. Processing work: [privileged] shutting down ZAdmin engine.
 done = False
2022-06-17 11:53:04,849  19292 [Engine      ] INF: ZAdmin, current state:final,
 exit code:shutdown
2022-06-17 11:53:04,849  19292 [Engine      ] INF: ZAdmin, exit code:shutdown:
 state changed to None
2022-06-17 11:53:04,849  19292 [Engine      ] DEB: ZAdmin, exit code:shutdown:
 breaking engine loop after shutdown
2022-06-17 11:53:04,850  19292 [Engine      ] DEB: ZAdmin, exit code:shutdown:
 end of event loop
2022-06-17 11:53:04,850  19292 [Adaptor     ] DEB: Adaptor : Cleanup for admin
 context on Terminate
2022-06-17 11:53:06,119  19292 [main     ] INF: Exiting SUCCESSFULLY
2022-06-17 11:53:06,119  19292 [main     ] INF: ZTP Exited
RP/0/RP0/CPU0:Jun 17 11:53:06.119 UTC: pyztp2[140]: %INFRA-ZTP-4-EXITED : ZTP
 exited
```

Generate tech support information for ZTP

Use this procedure to generate technical support information for ZTP. It collects troubleshooting data that Cisco Technical Support can use to analyze unresolved provisioning issues.

Use this task to 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.

Before you begin

Follow these steps to generate tech support information for ZTP.

Procedure

Run the **show tech-support ztp** command to collect ZTP debugging information.

Example

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

Use the show tech-support ztp command to collect all debugging information for the ZTP process.

The tech support information is saved as a .tgz file in the specified location and can be shared with Cisco Technical Support for ZTP troubleshooting.

7 Software Maintenance

Topics:

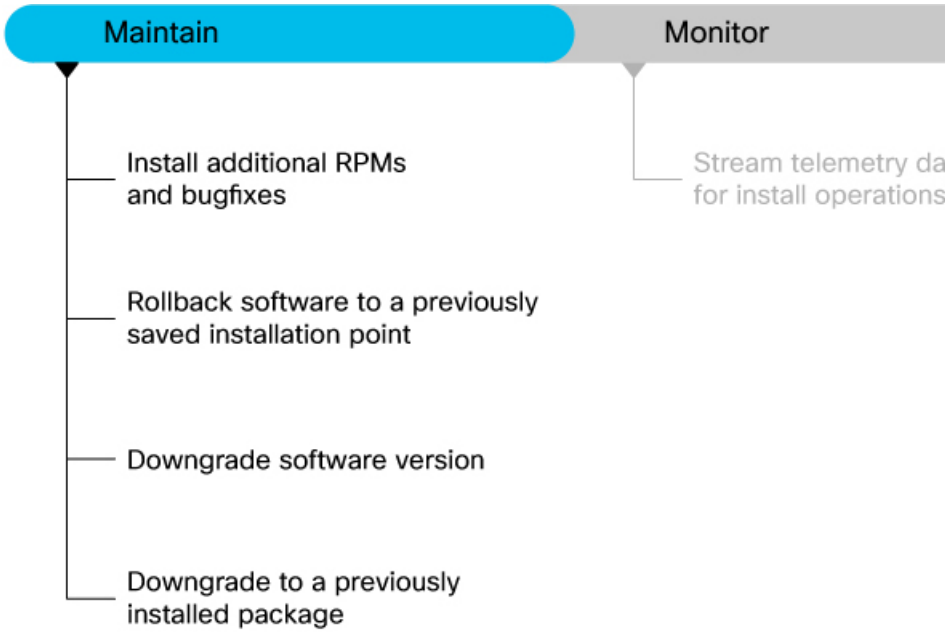
- Install additional RPMs and bug fixes
- Downgrade software version
- Telemetry sensor paths for install operations

This chapter provides details about managing the router for Cisco NCS 1010. Use it to review the requirements, behavior, examples, or limitations that support Cisco NCS 1010 setup and maintenance.

Use this reference to review manage the router.

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

- The following workflow shows the tasks involved in managing the software:
- **Figure 14: Workflow to Manintain and Monitor the Software Installation**



- This section contains the following topics:

Install additional RPMs and bug fixes

Use this procedure to install additional RPMs and bug fixes. It supports IOS XR software maintenance by outlining the required actions, inputs, or verification points for the activity.

Use this task to install additional RPMs and bug fixes.

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

Before you begin

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

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

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

Follow these steps to install additional RPMs and bug fixes.

Procedure

1. Complete the install RPMs using command line interface task.

   For details, see install RPMs using command line interface.

2. Complete the install RPMs using YANG data model task.

   For details, see install RPMs using YANG data model.

Install RPMs using command line interface

Use this procedure to install RPMs by using the command line interface. It supports IOS XR software maintenance when optional RPMs or bug fixes are available as TAR files.

Use this task to install RPMs using command line interface.

Optional RPMs and bug fixes are available as TAR files on the

Software Download

page. Starting with Cisco IOS XR Release 24.3.1, you are no longer required to manually extract the RPMs from the TAR file; you can install the bug fix RPM directly from the TAR file.

Before you begin

Follow these steps to install RPMs using command line interface.

Procedure

1. Check the available packages in the repository.

Example

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

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

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

Example

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

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

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

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

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

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

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

Least impactful apply method: process restart
```

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

> **Note**
>
> This option is not applicable when you downgrade or remove RPMs.

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

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

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

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

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

  > **Note**
  >
  > If the remote repository is reachable through a named VRF, you must mention the named VRF in the above commands. For example,
  >
  > where **vrf1** is the named VRF through which the remote repository is accessible.

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

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

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

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

  Or

  b. Apply the changes.

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

3. Check the status of the install operation.

Example

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

The following actions are available:

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

Note

Include the keyword

`noprompt`

in the commands to enable the system to bypass your permission to reload the NCS 1010.

4. Verify the image and packages are activated successfully.

Example

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

5. Commit the transaction.

Example

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

Note

The **install commit** command must be executed immediately after software upgrades or SMU installations and before applying any new configuration changes or powering off the device to prevent loss of changes upon reboot.

Install RPMs using YANG data model

Use this procedure to install RPMs by using the YANG data model. It supports IOS XR software maintenance through programmatic package installation and verification.

Use this task to install RPMs using YANG data model.

Use

`Cisco-IOS-XR-install-augmented-act.yang`

data model to install the RPMs or bug fixes.

Before you begin

Follow these steps to install RPMs using YANG data model.

Procedure

Invoke the **install-package-replace** RPC on the data model.

Example

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

```
<install-package-upgrade
xmlns=http://cisco.com/ns/yang/Cisco-IOS-XR-install-augmented-act>
  <source-type>ftp</source-type>
  <source>203.0.113.1;vrf1/repoinfra/install_RPMs.tar</source>
</install-package-upgrade>
```

If the install operation lists the repository reachable through a VRF, you must add the VRF name for the operation to be successful.

Downgrade software version

Use this procedure to downgrade software version. It supports IOS XR software maintenance by outlining the required actions, inputs, or verification points for the activity.

Use this task to downgrade software version.

Downgrade the current software version to a previous software release in case of an upgrade failure or based on requirement.

Note

When downgrading the software image from release 24.4.x to an earlier version, we recommend to manually downgrade the line card firmware as well to prevent any impact on various functionalities.

Before you begin

Check the FPD status and ensure that all the FPDs are in `CURRENT` state.

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

If the FPDs are not in `CURRENT` state, upgrade the FPDs.

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

After all the FPDs are upgraded, reload the NCS 1010.

```
RP/0/RP0/CPU0:ios#reload location all
Proceed with reload? [confirm]
```

After the NCS 1010 reloads, check that all the FPDs are in

`CURRENT`

state.

Follow these steps to downgrade software version.

Procedure

1. Determine the supported target versions to downgrade from the current version.

   Example

   ```
   RP/0/RP0/CPU0:ios#show install upgrade-matrix
   ```

   View the hardware or software limitations, and bridging SMUs required for the version downgrade. For more information about checking compatibility between the current and target versions, see

   Downgrading Packages:

   Customers can also downgrade user-specified packages (for example, xr-telnet). This is separate from downgrading the entire XR version, but an ISO for an earlier version of XR is used instead of a newer ISO.

2. Back up the file system of the current version for recovery purposes.

   Example
   Copy the running configuration to the harddisk: directory on the NCS 1010:

   ```
   RP/0/RP0/CPU0:ios#copy running-config harddisk:/running_config-<mmddyyyy>
   ```

Copy the running configuration to a remote server:

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

3. Download the target version from the Software Download Center.
4. You can either install from the remote repository or copy the ISO image file to the /harddisk: of the NCS 1010.

   Example

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

5. Verify that the MD5 checksum of the copied target file matches with the MD5 value of the source on the Software Download Center.

   Example

   ```
   RP/0/RP0/CPU0:ios#show md5 file /harddisk:/1010-x64-<target-version>.iso
   ```

6. Install the base image to downgrade the system.

   Example

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

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

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

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

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

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

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

     a. Install the image.

     b. Apply the changes.

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

7. After the base image is downgraded, install the additional packages. For more information, see Install additional RPMs and bug fixes on page 150.

   During an install operation, if the system reboots unexpectedly or an apply by reload results in the system failing to boot, it automatically recovers to its software state before the current transaction.

Downgrade to a previously installed package

Use this procedure to downgrade to a previously installed package. It supports IOS XR software maintenance by outlining the required actions, inputs, or verification points for the activity.

Use this task to downgrade to a previously installed package.

You can downgrade a package to a previously installed version. By default, the subsequent previous version (version previous to the current version) is installed. Also, you can downgrade the software to a specific version of interest. To remove a bug fix RPM from the installed packages, downgrade the package to a version where the fix was not applied.

> **Note**
>
> While downgrading, you can choose any previous version, including the base version of the RPM. However, when downgrading a bug fix RPMs, ensure that you also consider all dependencies of the current version.
>
> Bug fix RPM is an upgrade to the existing package. The action of removing a bug fix RPM either removes the entire feature, or fails if the package is mandatory.

You can use the **show install fixes deactivate** command to view information related to removing a bug fix. This command provides information such as the package changes, other bug fixes that get deactivate, instructions for adding packages missing for the bug fix removal to be successful, command for removing the bug fix, and any recommendations, if applicable. See the following example:

> **Note**
>
> You can specify any number of DDTS seperated by a space in the **show install fixes deactivate** command. For example, to know the recommendations for removing bug fix for ABC123, DEF456, and GHI789, you can use **show install fixes deactivate ABC123 DEF456 GHI789** command.

The following example shows the package `xr-telnet-24.3.1v1.0.1` is downgraded to `xr-telnet-24.3.1v1.0.0`. The path to source can be a local location or a configured repository.

Before you begin

Ensure you have access to the previously installed package and its source.

Follow these steps to downgrade to a previously installed package.

Procedure

1. Downgrade the package using one of the following options:

Example

```
RP/0/RP0/CPU0:ios#install package downgrade xr-telnet
```

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

```
RP/0/RP0/CPU0:ios#install source <path-to-source> xr-telnet-24.3.1v1.0.0
```

```
<install>
 <packages>
   <packagename>xr-telnet-24.3.1v1.0.0
            xr-telnet-24.0.11v1.0.0
            xr-telnet-24.4.1v1.0.0
</packagename>
 </packages>
   <source>file://<path-to-source>/</source>
</install>
```

- Downgrade the package where the fix was applied. When multiple older versions of the package are present in the configured repositories, the immediate previous version of the package is installed. Use caution when using this command as the current version of the package is removed completely.

  Apply the changes.

  > **Note**
  >
  > To identify whether to reload the NCS 1010 or restart the affected processes as part of the apply operation, use either **show install history last transaction verbose** command or **show install request** command.

- Install a specific earlier version of the optional package. The changes are applied automatically.

  > **Note**
  >
  > An automatic change may trigger a reload of the NCS 1010 depending on the package being downgraded.

- Use `install` RPC on the `Cisco-IOS-XR-install-act.yang` data model. Here is an example usage with a local repository:

  The package version `xr-telnet-24.3.1v1.0.1` `xr-telnet-24.0.11v1.0.1` `xr-telnet-24.3.1v1.0.1` is downgraded to `xr-telnet-24.3.1v1.0.0` `xr-telnet-24.0.11v1.0.0` `xr-telnet-24.4.1v1.0.0`.

2. Commit the operation.

Example

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

Note

The **install commit** command must be executed immediately after software upgrades or SMU installations and before applying any new configuration changes or powering off the device to prevent loss of changes upon reboot.

Telemetry sensor paths for install operations

This section provides details about streaming telemetry data for install operations for Cisco NCS 1010. Use it to review the requirements, behavior, examples, or limitations that support IOS XR software maintenance.

Use this reference to review stream telemetry data for install operations.

The following information supports stream telemetry data for install operations:

- To stream telemetry data that is related to software installation, you must create subscriptions to the sensor paths in the YANG data models. See *Obtain Data Models for Install Operation* for the list of supported data models. For information about establishing a telemetry session and creating subscriptions, see the

Stream Telemetry Data About	Description	YANG Path
Summary of active packages	Data is streamed after a successful **apply** operation. An active package is the software currently running on the system.	`Cisco-IOS-XR-install-oper: install/packages/active/summary`
Summary of committed packages	Data is streamed after a successful **commit** operation. A package that is committed remains active following a system reload.	`Cisco-IOS-XR-install-oper: install/packages/committed/summary`
Status of the last request operation	Data is streamed when starting a new request and also when entering an `idle` state. If the operation has failed, this includes error messages along with recovery state.	`Cisco-IOS-XR-install-oper: install/request`
Image version and GISO label	Data is streamed after a successful **apply** operation.	`Cisco-IOS-XR-install-oper: install/version`
Packaging information	Data is streamed at the start and end of a packaging operation.	`Cisco-IOS-XR-install-augmented-oper: install/history/latest-packaging-operation`
Atomic information	Data is streamed at the start and end of **apply** operation.	`Cisco-IOS-XR-install-augmented-oper: install/history/latest-atomic-change`

Stream Telemetry Data About	Description	YANG Path
Transaction information	Data is streamed at the start, in progress, and end of a **commit** operation. **Note** After a transactional rollback, some of the data such as summary of active packages, image version can change. However, telemetry events are not sent after the reload operation.	`Cisco-IOS-XR-install-augmented-oper:install/history/latest-transaction`

8 Setup and Upgrade Troubleshooting

Topics:

- NCS 1010 boot failure recovery
- Recover password
- Resolve insufficient disk space during software installation
- Recover frozen console prompt

This chapter provides details about troubleshooting NCS 1010 setup and upgrade. Use it to review the requirements, behavior, examples, or limitations that support setup and upgrade troubleshooting.

Use this reference to review troubleshoot NCS 1010 setup and upgrade.

Use the procedures in this section to troubleshoot NCS 1010 bring-up, software upgrade or downgrade by understanding the problem, probable cause, and the solution.

- The following image shows the tasks involved in finding solutions to NCS 1010 setup and upgrade issues:
- This section contains the following topics:

NCS 1010 boot failure recovery

This section provides details about recovering Cisco NCS 1010 from boot failure. Use it to review recovery methods for nonresponsive systems during setup or upgrade troubleshooting.

Use this reference to review recover NCS 1010 from boot failure.

The following information supports recover NCS 1010 from boot failure:

- If the command line interface is not accessible, you can recover the NCS 1010 from a boot failure using one of these recovery methods.

Boot the NCS 1010 using USB drive

Use this procedure to boot Cisco NCS 1010 from a USB drive. Use it when an externally bootable USB flash drive is required for recovery.

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

Problem:

After installing the hardware, you boot the NCS 1010 after connecting to the console port and powering ON the NCS 1010. The NCS 1010 initiates the boot process using the pre-installed operating system (OS) image. But the NCS 1010 fails to boot, times out or stops responding after the boot process initializes.

Cause:

The NCS 1010 does not boot if an install image is not present on the NCS 1010 or the image is corrupt.

Solution:

Boot the NCS 1010 using a bootable USB flash drive.

The bootable USB flash drive is used to reimage the NCS 1010 during system upgrade or boot the NCS 1010 in case of boot failure. During the USB boot process, the NCS 1010 is re-imaged with the version available on the USB flash drive.

To boot the NCS 1010 using a USB flash drive, you need the following devices:

- A local machine (Windows, Linux, or MAC) with USB Type-A.
- USB flash drive with a storage capacity that is between 8GB (min) and 32 GB (max). USB 2.0 and USB 3.0 are supported.

> **Note**
>
> USB Type-C is not supported.

Before you begin

Follow these steps to boot the NCS 1010 using USB drive.

Procedure

1. Create a bootable USB flash drive from your local machine (Windows or MAC):
   a) Connect the USB flash drive to your local machine and format it with File Allocation Table (FAT) 32 file system using the Windows Operating System or Apple MAC Disk Utility. Formatting the USB drive to FAT creates addressable sectors that ensures that each piece of information in the file can be found by the computer.

After formatting the USB flash drive, right-click on the USB disk and view the properties.

b) On the Software Download page, navigate to the required Cisco IOS XR product and release. The USB boot image is available in the format `<platform>-usb-<version>.zip` compressed file. For example, the USB boot image for Cisco NCS 1010s for release 24.3.1 is `1010-x64-usb-24.3.1.zip` `ASR9K-x64-usb-24.3.1.zip` file.

c) Download the compressed USB boot image from the Software Download page to your host computer.

d) Verify that the copy operation is successful. To verify, compare the file size on the Software Download page and the copied file on your computer. You can also verify the MD5 checksum value. This value ensures that the copied file is valid and untampered.

e) Unzip the file to extract the content of the compressed boot file inside the USB flash drive. This converts the USB flash drive to a bootable drive.

> **Note**
>
> The content of the zipped file (
>
> `EFI`
>
> and
>
> `boot`
>
> directories) should be extracted directly into the root of the USB flash drive. If the unzipping application places the extracted files in a new folder, move the
>
> `EFI`
>
> and
>
> `boot`
>
> directories to the root folder of the USB flash drive.

f) Remove the USB flash drive from your computer.

The USB flash drive is ready to be used as a bootable disk to install and boot the Cisco IOS XR image.

2. Boot the NCS 1010 using the bootable USB flash drive.

Example

```
RP/0/RP0/CPU0:ios#show media location all
Fri Jan 27 08:29:00.808 UTC

Media Info for Location: node0_RP0_CPU0
Partition         Size           Used          Percent        Avail
-------------------------------------------------------------------
rootfs:           54.4G          16.5G          30%           38G
data:             77.3G          20.5G          27%           56.8G
disk0:            3.9G           12M            1%            3.6G
/var/lib/docker   6.6G           17M            1%            6.2G
disk2:            15G            6.1G           42%           8.6G
log:              5.3G           572M           12%           4.4G
harddisk:         61G            19G            32%           39G
```

a) Use this procedure only on active RP; the standby RP must either be powered OFF or removed from the chassis. After the active RP is installed with images from USB, insert or power ON the standby RP as appropriate.

b) Connect to the console.
c) Insert the USB flash drive in the USB Port Type-A on the NCS 1010.

Ensure that the NCS 1010 is powered ON. When the USB bootable drive is plugged into an operational NCS 1010, the device is detected as disk2:. Verify using **show media location all** command.

d) View the contents of the USB drive.

Example

```
RP/0/RP0/CPU0:ios#dir disk2:
```

e) Initiate the reimage from the USB bootable drive.

Example

```
RP/0/RP0/CPU0:ios#reload bootmedia usb noprompt
```

```
RP/0/RP0/CPU0:ios#hw-module location all bootmedia usb
```

Note

If the NCS 1010 was powered OFF, power ON the NCS 1010. Press the

`Esc USB Flash Memory`

option in the

Boot Manager

menu, and press the

`Enter`

key. The BIOS GRUB automatically detects the image from the USB flash drive, starts the installation, and displays the progress of the installation operation.

The NCS 1010 reboots after the reimage with new version available in the USB drive. After the installation is complete, the NCS 1010 reboots and enters the prompt to configure the root username and password.

Boot the NCS 1010 using iPXE

Use this procedure to boot Cisco NCS 1010 with iPXE. Use it when DHCP boot parameters and network access are required to recover or install the system image.

Use this task to boot the NCS 1010 using ipxe.

Problem:

You connect to the console port and power ON the NCS 1010. The NCS 1010 initiates the boot process using the pre-installed operating system (OS) image. But the NCS 1010 fails to boot, times out or stops responding after the boot process initializes.

Cause:

The NCS 1010 does not boot if an install image is not present on the NCS 1010 or the image is corrupt.

Solution:

Boot the NCS 1010 using the image from an iPXE server.

iPXE is a pre-boot execution environment that is included in the network card of the management interfaces. It works at the system firmware (UEFI) level of the NCS 1010. iPXE enables network boot for a NCS 1010 that is offline. The bootloader downloads and installs the ISO image located on an HTTP, FTP, or TFTP server. iPXE boot re-images the NCS 1010. iPXE acts as a boot loader and provides the flexibility to choose the image that the system will boot based on the Platform Identifier (PID), the serial number, or the management MAC address. iPXE must be defined in the DHCP server configuration file.

Before you begin

Follow these steps to boot the NCS 1010 using ipxe.

Procedure

1. Configure the DHCP server for IPv4, IPv6, or both communication protocols before you use the iPXE boot.

   Example

   ```
   host <platform>
   {
   hardware ethernet <ncs1010-mac-address>;
   if exists user-class and option user-class = "iPXE" {
    filename = "http://<httpserver-address>/<path-to-image>/<image>";
   }
   ```

   Ensure that the above configuration is successful.

   ```
   host <platform>
   {
   option dhcp-client-identifier "<ncs1010-serial-number>";
     filename "http://<IP-address>/<path-to-image>/<image>";
     fixed-address <IP-address>;
   }
   ```

   The serial number of the NCS 1010 is derived from the BIOS and is used as an identifier.

   a) Create `dhcpd.conf` file in `/etc/` or `/etc/dhcp` directory. This configuration file stores the network information such as the path to the script, location of the ISO install file, location of the provisioning configuration file, serial number, MAC address of the NCS 1010. The following example shows a sample `dhcpd.conf` file.

      Example

      ```
      allow bootp;
      allow booting;
      ddns-update-style interim;
      option domain-name "cisco.com";
      option time-offset -8;
      ignore client-updates;
      default-lease-time 21600;
      max-lease-time 43200;
      ```

```
option domain-name-servers <ip-address-server1>, <ip-address-server2>;
log-facility local0;
 :
subnet <subnet> netmask <netmask> {
  option ncs1010 <ip-address>;
  option subnet-mask <subnet-mask>;
  next-server <server-addr>;
}
  :
host <hostname> {
  hardware ethernet e4:c7:22:be:10:ba;
  fixed-address <address>;
  filename "http://<address>/<path>/<image.bin>";
```

b) Test the server once the DHCP server is running. For example, for IPv4 protocol:

- Use the MAC address of the NCS 1010:

> **Note**
>
> Using the
>
> `host`
>
> statement provides a fixed address that is used for DNS, however, verify that option
>
> `77`
>
> is set to iPXE in the request. This option is used to provide the boot file to the system when required.

- Use the serial number of the NCS 1010:

2. Recover the NCS 1010 using iPXE boot.

Example

```
BIOS Ver: 09.19 Date: xx/xx/xxxx 17:02:33

Press <DEL> or <ESC> to enter boot manager.
    iPXE initialising devices...ok

iPXE 1.0.0+ (5fbe7) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP TFTP VLAN EFI ISO9660 NBI Menu
BootMode : 1
Trying net0...
net0: 00:00:01:1c:00:00 using i350-b on PCI01:00.0 (open)
  [Link:up, TX:0 TXE:0 RX:0 RXE:0]
Configuring (net0 00:00:01:1c:00:00)................... ok
net0: 203.0.113.1/255.255.255.0
net0: fe80::2a0:c9ff:fe00:0/64
net1: fe80::2a0:c9ff:fe00:1/64 (inaccessible)
net2: fe80::2a0:c9ff:fe00:2/64 (inaccessible)
net3: fe80::2a0:c9ff:fe00:3/64 (inaccessible)
net4: fe80::200:ff:fe00:4/64 (inaccessible)
net5: fe80::200:ff:fe00:5/64 (inaccessible)
net6: fe80::662:73ff:fe08:1dba/64 (inaccessible)
Next server: 203.0.113.17
```

```
Filename: http://203.0.113.15/system_image.iso
http://203.0.113.15/<image>... ok
```

```
RP/0/RP0/CPU0:ios# hw-module location all bootmedia network reload
Wed Dec 23 15:29:57.376 UTC
Reload hardware module ? [no,yes]
```

a) Connect to the console.
b) Power ON the NCS 1010.
c) Press `Esc`
d) Use the arrow key and navigate to the `Built-in EFI iPXE` option in the **Boot Manager** menu, and press the `Enter` key.

Example

```
iPXE> ifstat
net0: 00:a0:c9:00:00:00 using i350-b on PCI01:00.0 (closed)
  [Link:up, TX:0 TXE:0 RX:0 RXE:0]
net1: 00:a0:c9:00:00:01 using i350-b on PCI01:00.1 (closed)
  [Link:up, TX:0 TXE:0 RX:0 RXE:0]
net2: 00:a0:c9:00:00:02 using i350-b on PCI01:00.2 (closed)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Down (http://ipxe.org/38086193)]
net3: 00:a0:c9:00:00:03 using i350-b on PCI01:00.3 (closed)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Down (http://ipxe.org/38086193)]
net4: 00:00:00:00:00:04 using dh8900cc on PCI02:00.1 (closed)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Down (http://ipxe.org/38086193)]
net5: 00:00:00:00:00:05 using dh8900cc on PCI02:00.2 (closed)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Down (http://ipxe.org/38086193)]
net6: 04:62:73:08:57:86 using dh8900cc on PCI02:00.3 (closed)
  [Link:up, TX:0 TXE:0 RX:0 RXE:0]

iPXE> set net6/ip 192.0.2.255
iPXE> set net6/netmask 255.255.255.0
iPXE> set net6/gateway 10.48.42.1
iPXE>
iPXE> ifopen net6

iPXE> ping 10.48.42.1
64 bytes from 10.48.42.1: seq=1
64 bytes from 10.48.42.1: seq=2
Finished: Operation canceled (http://ipxe.org/0b072095)
```

e) Boot the image using one of the following options:

- Option 1: Boot with ISO image. After the reimage is successful, add optional RPMs, bug fixes and update running configuration file.
- Option 2: [Preferred option] Boot with Golden ISO (GISO) image that contains the ISO image, optional RPMs, bug fixes and configuration file. Booting with GISO saves time by eleminating the need to update the files individually.

You must keep the standby RP in the BIOS while installing the image on the active RP.

The BIOS GRUB automatically detects the image from the iPXE server, starts the installation, and displays the progress of the installation operation. After the installation is complete, the NCS 1010 reboots and enters the prompt to configure the root username and password.

You can also boot the NCS 1010 from the iPXE server by using the **hw-module location all bootmedia network reload** command.

This command configures the NCS 1010 to perform a network-based boot across all modules in the NCS 1010 before a restart. Upon reload, the NCS 1010 attempts to load the operating system image from the specified iPXE server.

Recover password

Use this procedure to recover the password on Cisco NCS 1010. It helps restore access when incorrect or lost login credentials prevent device access.

Use this task to recover password.

Problem:

Unable to access the NCS 1010 due to incorrect login credentials.

Cause:

A root password is used to login to the NCS 1010. If you forget this root password, you cannot access the NCS 1010.

Solution:

If you lose your admin and root user credentials, the NCS 1010 becomes inaccessible. The system can be recovered using a NCS 1010 reimage using iPXE or USB boot. However, this approach is not scalable.

You can use the **system recovery** feature to recover the lost password.

With this feature, the system is recovered without the need to reimage the NCS 1010. The system is recovered to its initial state with the current running software. The installed software and SMUs are retained after the system is recovered. The process complies with the Cisco Product Security Baseline (PSB) where user data is securely erased before recovering the NCS 1010. The following data that are generated at run-time are erased:

- XR and admin configuration including the password data
- Cryptographic keys on the disk
- Data on encrypted partition
- Generated core files
- SNMP interface index files
- Third-party application (TPA) software and data
- Files created by the user

Use the following procedure to recover the password on NCS 1010.

Note

This procedure is applicable only when you have already enabled the password recovery feature on your NCS 1010.

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

Before you begin

Follow these steps to recover password.

Procedure

1. Power ON the NCS 1010, and press the `ESC` on the RP console to enter the BIOS GRUB menu.

   This procedure must be executed on each RP individually on a modular system.

2. Boot on the standby RP. Press `ESC` key to enter the GRUB (bootstrap program) menu.
3. On the RP0 card console select the **IOS-XR-recovery** option from the GRUB menu and press **Enter**.
4. Select the **IOS-XR-recovery** option from the GRUB menu and press **Enter** on the card console when the `Initiating IOS-XR System Recovery...` message is displayed on the card console.

   Note

   Do not wait until the card reaches the

   `Enter root-system username:`

   prompt. If you reach this prompt, the card will reload automatically and exit the BIOS GRUB menu. The card will boot up as active post the recovery process.

5. On the RP card, create a new root user and password. Log in to the NCS 1010 using the new root username and password.

   Example

   ```
   RP/0/RP1/CPU0:June 10 06:13:24.551 CEST: sys_rec[1188]:
   %SECURITY-SYSTEM_RECOVERY-1-REPORT :
   System Recovery at 06:10:19 CEST Fri June 10 2022 was successful

   RP/0/RP1/CPU0:June 10 06:15:13.967 CEST: sys_rec[1188]:
   %SECURITY-SYSTEM_RECOVERY-1-REPORT :
   System Recovery
   ```

   The NCS 1010 boots with the default configuration. Proceed with configuring the NCS 1010 or load a configuration from a backup file if you had already taken a backup. It is recommended to backup data and save the configuration on an external server.

   Ensure that you see this message in the RP console. If this message is not displayed, then repeat the process from step 1 to step 5 until you see the message:

   The password recovery procedure is complete.

The option to recover the system using console port is disabled on bootup because all the previous configurations are erased. With this configuration disabled, if you select **IOS-XR-recovery** option from GRUB menu to recover the system, the recovery is skipped. Enable the password recovery feature again using the **system recovery** command.

Resolve insufficient disk space during software installation

Use this procedure to resolve insufficient disk space during software installation. It helps recover from package installation failures caused by limited available storage.

Use this task to rectify insufficient disk space when installing software.

Problem:

The software installation terminates with the error `Error on 0/1/CPU0: Insufficient disk space to install packages`.

Cause:

To install the Cisco IOS XR software, an unused disk space of so-and-so must be available on the NCS 1010. If this space is not available before installing the software, the installation process terminates with the error.

Solution:

Identify the required disk space using the **show install log** or **install add** command.

View the space consumed by the harddisk: location using the **show media location all** command.

```
RP/0/RP0/CPU0:ios#show media location all
Wed Jan 8 08:29:00.808 UTC

Media Info for Location: node0_RP0_CPU0
Partition                          Size     Used  Percent    Avail
-------------------------------------------------------------------
rootfs:                          54.4G    16.5G      30%      38G
data:                            77.3G    20.5G      27%    56.8G
disk0:                            3.9G      12M       1%     3.6G
/var/lib/docker                   6.6G      17M       1%     6.2G
disk2:                             15G     6.1G      42%     8.6G
log:                              5.3G     572M      12%     4.4G
harddisk:                          61G      19G      32%      39G

Media Info for Location: node0_RP1_CPU0
Partition                          Size     Used  Percent    Avail
-------------------------------------------------------------------
rootfs:                          54.3G    16.5G      30%    37.9G
data:                            77.4G    46.1G      60%    31.4G
disk0:                            3.9G     8.5M       1%     3.6G
/var/lib/docker                   6.6G      19M       1%     6.2G
log:                              5.3G     492M      10%     4.5G
harddisk:                          61G      44G      78%      14G

Media Info for Location: node0_0_CPU0
Partition                          Size     Used  Percent    Avail
-------------------------------------------------------------------
rootfs:                          54.4G    10.1G      18%    44.4G
data:                            77.3G     1.9G       2%    75.5G
/var/lib/docker                   6.6G      16M       1%     6.2G
disk0:                            3.9G     8.2M       1%     3.6G
harddisk:                          61G     109M       1%      57G
log:                              5.3G     372M       8%     4.6G
```

```
Media Info for Location: node0_6_CPU0
Partition                              Size     Used  Percent    Avail
-------------------------------------------------------------------------
rootfs:                              54.4G    10.1G      18%    44.4G
data:                                77.3G     1.9G       2%    75.4G
disk0:                                3.9G     8.3M       1%     3.6G
/var/lib/docker                       6.6G      16M       1%     6.2G
harddisk:                              61G     154M       1%      57G
log:                                  5.3G     374M       8%     4.6G
```

Use the following procedure to free up the disk space to make room for the software installation.

Before you begin

Follow these steps to rectify insufficient disk space when installing software.

Procedure

1. Remove inactive packages from the system.

   Example
   View the inactive packages:

```
RP/0/RP0/CPU0:ios(admin)#show install inactive
6 inactive package(s) found:
    ncs5500-xr-6.6.1
    ncs5500-k9sec-3.1.0.0-r661
    ncs5500-mpls-2.1.0.0-r661
    ncs5500-isis-2.1.0.0-r661
    ncs5500-mcast-2.1.0.0-r661
    ncs5500-mgbl-3.0.0.0-r661
```

   Remove the inactive packages:

```
RP/0/RP0/CPU0:ios(admin)#install remove inactive all synchronous
   instdir[198]: %INSTALL-INSTMGR-6-INSTALL_OPERATION_STARTED :
Install operation 8 '(admin) install remove inactive all' started by user
'user_b'
Install operation 8 '(admin) install remove inactive all' started by user
'user_b' at
   09:25:41 UTC Fri June 10
Info:     This operation will remove the following package:
ncs5500-xr-6.6.1
    ncs5500-k9sec-3.1.0.0-r661
    ncs5500-mpls-2.1.0.0-r661
    ncs5500-isis-2.1.0.0-r661
    ncs5500-mcast-2.1.0.0-r661
    ncs5500-mgbl-3.0.0.0-r661
Proceed with removing these packages? [confirm]
The install operation will continue synchronously.
```

2. Remove stale or unnecessary files from the harddisk: location such as cores, debug logs, kdump and showtech data. We recommended that you do not remove files from other partitions because these locations may contain files that are relevant to collecting debug information. Carefully inspect the files to be deleted.

Example

```
RP/0/RP0/CPU0:ios#rmdir harddisk:
Remove directory filename []?newdir
Delete harddisk:/newdir[confirm]y
```

```
RP/0/RP0/CPU0:ios#delete harddisk:/file
```

```
RP/0/RP0/CPU0:ios#dir harddisk:
 Directory of harddisk:
 37146       drwx  4096        Sun Dec 14 15:30:48 2008  malloc_dump
 43030       drwx  4096        Wed Dec 24 11:20:52 2008  tracebacks
 43035       drwx  4096        Thu Jan  8 18:59:18 2009  sau
 51026       drwx  4096        Sat Dec 27 02:52:46 2008  tempA
 51027       drwx  4096        Sat Dec 27 02:04:10 2008  dir.not.del
 -430307552  -rwx  342         Fri Jan 16 10:47:38 2009  running-config
 -430305504  -rwx  39790       Mon Jan 26 23:45:56 2009  cf.dat
39929724928 bytes total (39883235328 bytes free)
```

Use the **delete** command to remove specific directory or files. When a directory contains files such as images, bug fixes or configuration files, you must remove the files before deleting the directory.

Verify that the unwanted directory is removed from the harddisk.

Recover frozen console prompt

Use this procedure to recover a frozen console prompt. It helps restore console access when the prompt stops responding during setup or troubleshooting.

Use this task to recover frozen console prompt.

Problem:

The console access is frozen and does not respond. In this state, no output or input characters are displayed on the console.

Cause:

The Priority Flow Control (PFC) functionality is enabed on the console by default. The PFC is also referred to as Class-based Flow Control (CBFC) or Per Priority Pause (PPP) is a mechanism that prevents frame loss due to congestion. Pressing the `Ctrl + S` keys enables the flow control and no output will be seen on the XR console until resumed.

Solution:

Reset the console prompt.

Before you begin

Follow these steps to recover frozen console prompt.

Procedure

Press the `Ctrl + Q` keys to resume the console output.

9 Disaster Recovery

Topics:

- Disaster recovery partitions
- CPU replacement considerations
- Backup ISO image health check
- Automated file management behavior

This chapter describes disaster recovery for Cisco NCS 1010, including partition behavior, CPU replacement considerations, and backup ISO image health checks.

Use this reference to understand disaster recovery behavior for Cisco NCS 1010.

The disaster recovery behavior information includes:

- How backup partitions support disaster recovery operations.
- How the chassis SSD stores software and configuration information used during recovery.
- How recovery information helps restore Cisco NCS 1010 after CPU replacement or image corruption.

Disaster recovery partitions

This section explains disaster recovery partitions and how the RP SSD and chassis SSD support software and configuration restoration workflows.

Disaster recovery partitions is a Cisco NCS 1010 feature or capability that

- supports the related system setup or management workflow,
- defines behavior that affects Cisco NCS 1010 operation, and
- provides context before you configure or verify the feature.

Disaster recovery partitions details

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

This section explains CPU replacement considerations and how the chassis SSD helps preserve software and configuration during disaster recovery operations.

CPU replacement considerations is a Cisco NCS 1010 feature or capability that

- supports the related system setup or management workflow,
- defines behavior that affects Cisco NCS 1010 operation, and
- provides context before you configure or verify the feature.

CPU replacement considerations details

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.

Backup ISO image health check

This section provides details about backup ISO image health checks, including how background audits detect corruption and trigger recovery actions.

Use this reference to understand backup ISO image health check for Cisco NCS 1010.

The backup ISO image health check information includes:

- The Health Check feature ensures error-free booting of NCS 1010 chassis during disaster recovery operations. NCS 1010 has a partition for disaster recovery where the backup ISO image is stored. The backup ISO image is stored in the chassis SSD.
- The chassis SSD content is audited against the running software by the install process in the background every 12 hours to detect corruption. If the ISO image is corrupted, the software will recover it by copying from the backup location. If the software fails to synchronize with the chassis SSD, then the Disaster Recovery ISO Image Corruption alarm is raised. See the Troubleshooting Guide for Cisco NCS 1010 to clear the alarm.

Automated file management behavior

Provides details about automated file management on Cisco NCS 1010, including feature history, SSD file types, log file rotation, showtech residual file cleanup, and OTDR file cleanup.

Use this reference to understand how the automated file management system archives older files and removes files from local nodes to free SSD space.

Table 12: Feature History

Feature Name	Release Information	Feature Description
Automated File Management System	Cisco IOS XR Release 24.4.1	The Automated File Management System handles files efficiently on each node. The system automatically archives older files and removes them from local nodes to free SSD space. It manages system-generated log files, showtech-related residual files, and OTDR files.

The SSD stores these file types.

- **User**: The user creates and owns files for requirements and deletes the files when the files are no longer needed.
- **System**: The system organizes files automatically based on the file content and the application that created the content, such as .log files and showtech-related residual files. Automated file management applies to system-generated files.

Automated file management handles log files in these stages.

- The NCS 1014 system uses log rotation configurations to manage log files as required.
- The system checks for *.log* files that exceed 10 MB. This threshold applies only to files in the `/tmp` folder. For files in other folders, the system uses a different threshold and applies the same process.
- After locating the file, the system archives that file with the .gz extension and creates a new .log file with the same name.
- The system maintains one active log file and two archived files.
- When new files are archived, the system deletes the oldest archived file so that only the two most recently archived files and the current log file are retained.

Automated file management handles showtech-related residual log files in these stages.

- The system generates residual files during log collection when **Showtech logs** is in operation.
- After collecting the logs, the system automatically removes the residual files to conserve space.

10 Border Gateway Protocol Configuration

Topics:

- BGP routing

This chapter describes Border Gateway Protocol (BGP) configuration for Cisco NCS 1010, including routing concepts, router identifiers, prerequisites, and procedures for configuring BGP peers.

BGP routing

This section explains how BGP routing exchanges reachability information between peers and supports management connectivity on Cisco NCS 1010 devices.

BGP routing is a Cisco NCS 1010 feature or capability that

- supports the related system setup or management workflow,
- defines behavior that affects Cisco NCS 1010 operation, and
- provides context before you configure or verify the feature.

BGP routing details
Table 13: Feature History

Feature Name	Release Information	Feature Description
Configuring BGP	Cisco IOS XR Release 7.11.1	BGP routers from a TCP connection between peer routers to exchange network reachability information to open and confirm the connection parameters. BGP on NCS 1010 is enabled only on management port enabling the customer to manage the Dynamic Circuit Network(DCN) connectivity.

BGP uses TCP as its transport protocol. Two BGP routers form a TCP connection between one another (peer routers) and exchange messages to open and confirm the connection parameters.

BGP routers exchange network reachability information. This information is mainly an indication of the full paths (BGP autonomous system numbers) that a route should take to reach the destination network. This information helps construct a graph that shows which autonomous systems are loop free and where routing policies can be applied to enforce restrictions on routing behavior.

Any two routers forming a TCP connection to exchange BGP routing information are called peers or neighbors. BGP peers initially exchange their full BGP routing tables. After this exchange, incremental updates are sent as the routing table changes. BGP keeps a version number of the BGP table, which is the same for all its BGP peers. The version number changes whenever BGP updates the table due to routing information changes. Keepalive packets are sent to ensure that the connection is alive between the BGP peers and notification packets are sent in response to error or special conditions.

Note

ASN change for the BGP process is not currently supported via **commit replace**.

BGP implementation prerequisites

This section provides details about the prerequisites for implementing BGP, including user group permissions and task IDs required to run BGP commands.

Use this reference to understand BGP implementation prerequisites for Cisco NCS 1010.

The BGP implementation prerequisites information includes:

- 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

This section provides details about how BGP selects a router identifier and why a stable router ID helps avoid unnecessary session resets.

Use this reference to understand BGP router identifier for Cisco NCS 1010.

The BGP router identifier information includes:

- 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.
- 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).

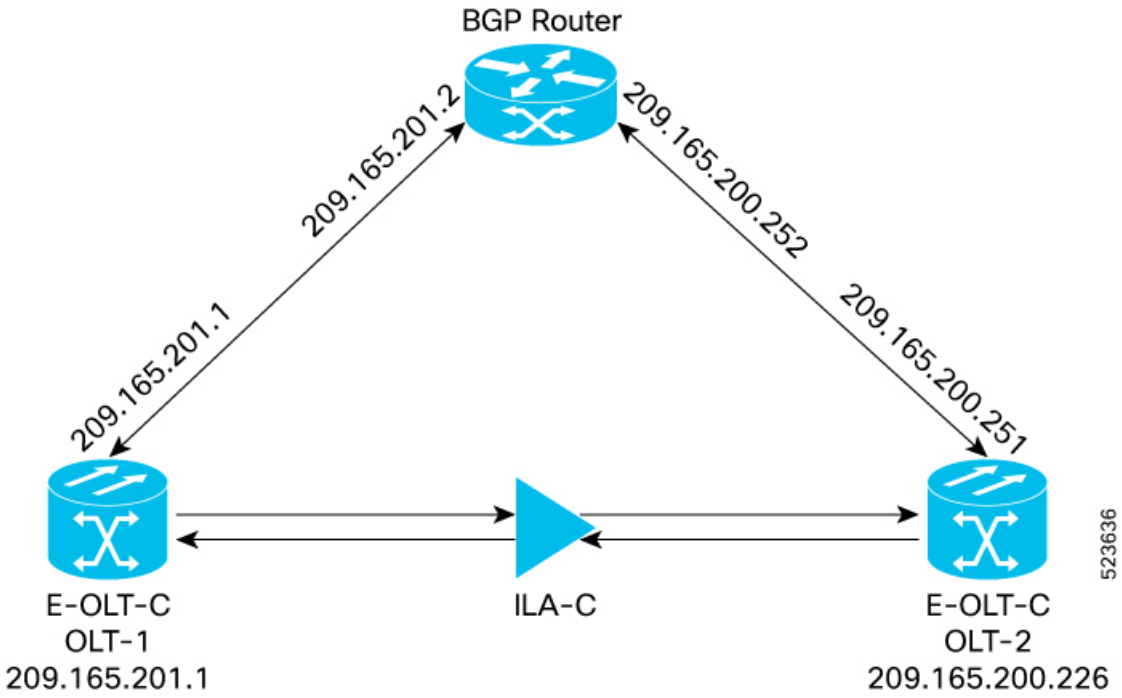
> **Note**
>
> We strongly recommend that the **bgp router-id** command is configured to prevent unnecessary changes to the router ID (and consequent flapping of BGP sessions).

Configure BGP

Use this procedure to configure BGP on Cisco NCS 1010 so that peer routers can exchange route information for management connectivity.

Configure BGP so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

Figure 15: BGP Topology



Note

Autonomous system numbers 2.0 standard supports 0 to 65535, and autonomous system numbers standard 4.0 supports 65536 onwards.

Before you begin

Follow these steps to configure BGP:

Procedure

1. Run the **config** command to verify the configuration.

Example

```
config
route-policy pass-all
end-policy
commit

RP/0/RP0/CPU0:OLT-1#conf
Fri Feb 23 09:32:29.216 IST
router bgp 1
bgp router-id 209.165.201.1
 address-family ipv4 unicast
  redistribute connected
  redistribute ospf 1 route-policy pass-all
 !
 address-family vpnv4 unicast
 !
 neighbor 209.165.201.2
  remote-as 100
```

```
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!
commit

show running-config router bgp
Thu Jan 11 15:54:43.439 IST
router bgp 1
 bgp router-id 209.165.201.1
 address-family ipv4 unicast
  redistribute connected
  redistribute ospf 1 route-policy pass-all
 !
 address-family vpnv4 unicast
 !
 neighbor 209.165.201.2
  remote-as 100
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!
```

2. Run the **conf** command to verify the configuration.

Example

```
RP/0/RP0/CPU0:OLT-2#conf
Fri Feb 23 09:32:29.216 IST
router bgp 1
 bgp router-id 209.165.200.226
 address-family ipv4 unicast
  redistribute connected
  redistribute ospf 1 route-policy pass-all
 !
 address-family vpnv4 unicast
 !
 neighbor 209.165.200.252
  remote-as 100
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!
commit

show running-config router bgp
Thu Jan 11 15:59:12.210 IST
router bgp 1
 bgp router-id 209.165.200.226
 address-family ipv4 unicast
  redistribute connected
  redistribute ospf 1 route-policy pass-all
 !
 address-family vpnv4 unicast
```

```
 !
 neighbor 209.165.200.252
  remote-as 100
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!

show route
route  router-id
```

3. Run the **config** command to verify the configuration.

Example

```
RP/0/RP0/CPU0:BGP-ROUTER#config
Fri Feb 23 09:32:29.216 IST
router bgp 100
 bgp router-id 209.165.201.19
 address-family ipv4 unicast
  redistribute connected
 !
 neighbor 209.165.201.1
  remote-as 1
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
 neighbor 209.165.200.251
  remote-as 1
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!
commit

show running-config router bgp
Thu Jan 11 15:59:58.059 IST
router bgp 100
 bgp router-id 209.165.201.19
 address-family ipv4 unicast
  redistribute connected
 !
 neighbor 209.165.201.1
  remote-as 1
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
 neighbor 209.165.200.251
  remote-as 1
  address-family ipv4 unicast
   route-policy pass-all in
```

```
   route-policy pass-all out
  !
 !
!
```

The system completes the configure BGP task.

11 Cisco Discovery Protocol Configurations

Topics:

- CDP support on Cisco NCS 1010

Explains Cisco Discovery Protocol (CDP) configuration settings for Cisco NCS 1010, including global behavior, interface behavior, default timers, and monitoring outputs used to verify neighbor discovery.

CDP support on Cisco NCS 1010

Provides details about CDP support on Cisco NCS 1010, including neighbor discovery behavior, default advertisement timers, and required RPM packages.

Use this reference to review CDP support, discovery behavior, feature history, and RPM package requirements for Cisco NCS 1010.

CDP configuration supports Cisco NCS 1010 setup and management with the following capabilities.

- Supports the related system setup or management workflow.
- Defines behavior that affects Cisco NCS 1010 operation.
- Provides context before you configure or verify the feature.

CDP provides the following discovery behavior on Cisco NCS 1010.

- CDP is a Cisco proprietary Layer 2 protocol used to obtain information about peer Cisco devices.
- CDP exchanges packets with neighboring devices to discover their platform type and capabilities.
- Each CDP-enabled device sends periodic advertisements to a multicast address and advertises at least one address at which it can receive messages.
- The hold-time information indicates how long a receiving device keeps CDP information before discarding it. The default hold time is 180 seconds.
- Each device listens to periodic CDP messages every 60 seconds by default to learn about neighboring devices and determine when their interfaces go up or down.

Table 14: 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 support 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

Use this procedure to enable CDP globally on Cisco NCS 1010 so that the device can participate in neighbor discovery.

Configure enable CDP globally so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

To enable CDP globally, use the following commands:

Before you begin

Follow these steps to enable CDP globally:

Procedure

1. Run the **configure** command to enter configuration mode.
2. Run the **cdp** command to enable CDP.
3. Run the **commit** command to save the configuration changes.

The system completes the enable CDP globally task.

Disable CDP globally

Use this procedure to disable CDP globally on Cisco NCS 1010 when neighbor discovery must be turned off for the system.

Configure disable CDP globally so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

To disable CDP globally, use the following commands:

Before you begin

Follow these steps to disable CDP globally:

Procedure

1. Run the **configure** command to enter configuration mode.
2. Run the **no cdp** command to disable CDP.
3. Run the **commit** command to save the configuration changes.

The system completes the disable CDP globally task.

Enable CDP on interfaces

Use this procedure to enable CDP on interfaces so that Cisco NCS 1010 can discover neighboring devices on selected links.

Configure enable CDP on interfaces so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

To enable CDP on the management interface, use the following commands:

To enable CDP on the Gigabit Ethernet (GE) interface, use the following commands:

interface mgmtEth *rack/slot/instance/port*

The following example enables CDP on the management interface.

interface gigabitEthernet *rack/slot/instance/port*

The following example enables CDP on the Gigabit Ethernet (GE) interface.

Before you begin

Follow these steps to enable CDP on interfaces:

Procedure

1. Run the **configure** command to enter configuration mode.
2. Run the **cdp** command to enable CDP.
3. Run the **commit** command to save the configuration changes.
4. Run the **configure** command to complete this action.

Example

```
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
```

5. Run the **configure** command to enter configuration mode.
6. Run the **cdp** command to enable CDP.
7. Run the **commit** command to save the configuration changes.
8. Run the **configure** command to complete this action.

Example

```
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
```

The system completes the enable CDP on interfaces task.

Modify CDP default settings

Use this procedure to modify CDP default settings such as packet version, hold time, and update timer on Cisco NCS 1010.

Configure modify CDP default settings so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

Use this task to modify CDP parameters such as the default version, holdtime, and timer.

Before you begin

Follow these steps to modify CDP default settings:

Procedure

1. Run the **configure** command to enters global configuration mode.

Example

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

Enters global configuration mode.

2. Run the **cdp advertise v1** command to 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.

   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.

3. Run the **cdp holdtime seconds** command to 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. 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.

   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.

4. Run the **cdp timer seconds** command to 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. A lower timer setting causes CDP update packets to be sent more frequently.

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.

5. Run the **commit** command to saves the configuration changes and remains within the configuration session.

Example

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

Saves the configuration changes and remains within the configuration session.

The system completes the modify CDP default settings task.

Monitor CDP

Use this procedure to monitor CDP on Cisco NCS 1010 so that you can confirm neighbor discovery and review operational details.

Configure monitor CDP so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

Use the **show cdp entry** *entry-name* command to display information about a specific neighboring device or all the neighboring devices discovered using CDP.

Use the **show cdp interface** [*interface-name*] command to display information about the interfaces on which CDP is enabled.

Before you begin

Follow these steps to monitor CDP:

Procedure

---

1. Run the **Use the show cdp command to display global CDP information.** command to complete this action.
2. Run the **required** command to complete this action.

Example

```
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
```

3. Run the **Use the show cdp neighbors command to display detailed information about neighboring devices discovered using CDP.** command to complete this action.
4. Run the **required** command to complete this action.

Example

```
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
```

5. Run the **required** command to complete this action.

Example

```
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
```

6. Run the **required** command to complete this action.

Example

```
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
```

7. Run the **required** command to complete this action.

Example

```
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
```

8. Run the **required** command to complete this action.

Example

```
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
```

9. Run the **Use the show cdp traffic command to display information about the traffic gathered between devices using CDP.** command to complete this action.
10. Run the **required** command to complete this action.

Example

```
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
```

The system completes the monitor CDP task.

12 Daisy Chain Topology

Topics:

- Daisy chain topology
- Configure daisy chain on management ports
- Verify daisy chain
- Enable storm control on TOR switch
- Disable DAD on the management port

This chapter describes daisy chain topology for Cisco NCS 1010, including management-port configuration, verification, storm control, and DAD settings for nodes.

Daisy chain topology

This section explains daisy chain topology and how connected Cisco NCS 1010 nodes maintain management connectivity through the ring arrangement.

Daisy chain is a Cisco NCS 1010 feature or capability that

- supports the related system setup or management workflow,
- defines behavior that affects Cisco NCS 1010 operation, and
- provides context before you configure or verify the feature.

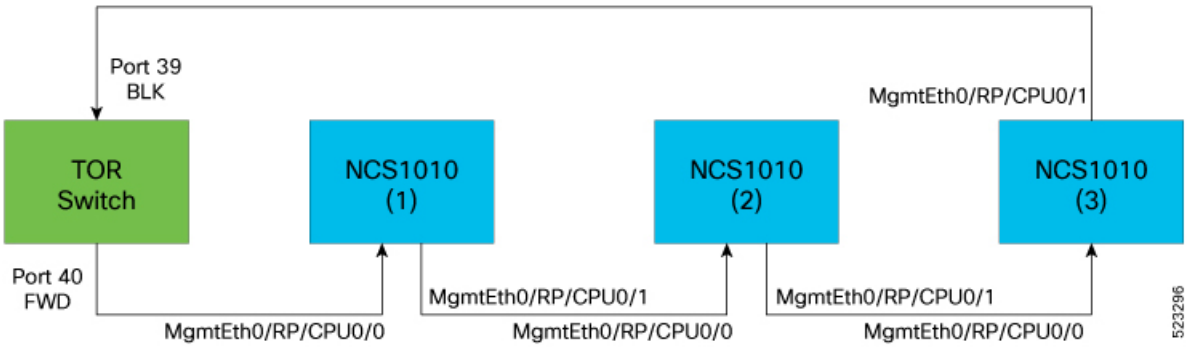
Daisy chain topology details
Table 15: Feature History

Feature Name	Release Information	Description
Daisy Chain on NCS 1010 Management Ports	Cisco IOS XR Release 7.10.1	You can now connect NCS 1010 devices in a Daisy Chain topology. Here multiple NCS 1010 devices are connected to form a ring-like topology, and only the first and last nodes are connected to a Top-of-Rack (TOR) switch, thereby reducing the number of connections. The Daisy Chain topology also provides more redundancy as data is transmitted in both directions. The first connection acts as a primary path and carries the traffic whereas the last connection acts as a secondary path. In case the primary path fails, the secondary path serves as its backup for data transmission and allows traffic to continue to transmit in the network.

The daisy chain arrangement allows multiple NCS 1010 nodes to be connected to each other in a ring, where only the first and the last nodes are connected to a TOR switch. The switch allows management of all the NCS 1010 devices in the network and also prevents traffic storm. The data transmitted over the network passes through each node in the ring until it reaches the destination node. This arrangement allows the switch to send data in both directions and prevents one node failure from cutting off certain network parts.

The following diagram shows the Daisy Chain topology where three NCS 1010 nodes are connected to each other over the management ports *0* and *1*.

Figure 16: NCS 1010 in a Daisy Chain Network



Configure daisy chain on management ports

Use this procedure to configure daisy chain on management ports so that Cisco NCS 1010 nodes can support management connectivity.

Configure daisy chain on management ports so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

To configure IP Address on management port 0, use these commands:

Before you begin

The following prerequisites must be met before configuring Daisy Chain on NCS1010:

- Enable Storm Control on Switch.
- STP must be running on the TOR switch.
- Daisy chain must be enabled on all the NCS1010 devices in the topology.

Configuring Daisy Chain on managements ports of NCS 1010 devices involves the following tasks:

- Configure IP Address on Management Port
- Configure Daisy Chain

Configure IP Address on Management Port

interface *type* **Rack/Slot/Instance/Port**

ipv4 address *ip-address*

Configure Daisy Chain

interface *type* **Rack/Slot/Instance/Port**

no ipv4 address

no ipv6 address

bridge-port routed-interface *type***Rack/Slot/Instance/Port**

Before you begin

Follow these steps to configure daisy chain on management ports:

Procedure

Run the **configure** command to enter configuration mode.

The system completes the configure daisy chain on management ports task.

Verify daisy chain

Use this procedure to verify daisy chain status so that you can confirm management connectivity and topology behavior after configuration.

Verify daisy chain so that you can confirm the Cisco NCS 1010 configuration, operational state, or feature behavior before continuing with related setup tasks.

To verify daisy chain configuration on management ports of NCS1010 device, use these commands:

Before you begin

Follow these steps to verify daisy chain:

Procedure

Run the **Verify daisy chain** command to complete this task.

show running-config interface*type*

The system completes the verify daisy chain task.

Enable storm control on TOR switch

Use this procedure to enable storm control on the TOR switch so that the daisy chain topology limits excessive traffic.

Configure enable storm control on TOR switch so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

To enable storm control on a TOR switch, use the following commands:

Before you begin

Follow these steps to enable storm control on TOR switch:

Procedure

1. Run the **Enable storm control on TOR switch** command to complete this task.

   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.

2. Run the **Enable storm control on TOR switch** command to complete this task.

   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.

3. Run the **Enable storm control on TOR switch** command to complete this task.

   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.

4. Run the **Enable storm control on TOR switch** command to complete this task.

   errdisable recovery interval *value*

5. Run the **Enable storm control on TOR switch** command to complete this task.

   errdisable recovery cause storm-control

The system completes the enable storm control on TOR switch task.

Disable DAD on the management port

Use this procedure to disable DAD on the management port when the daisy chain topology requires duplicate address detection to be turned off.

Configure disable DAD on the management port so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

To disable DAD on the management port, use the following commands:

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.

interface *type* **Rack/Slot/Instance/Port**

ipv6 nd dad attempts *value*

Before you begin

Follow these steps to disable DAD on the management port:

Procedure

Run the **configure** command to enter configuration mode.

The system completes the disable DAD on the management port task.

13 Access Control Lists

Topics:

- Access control lists
- How access control lists work
- Access control list application behavior
- ACL configuration procedures

This chapter describes Access Control Lists (ACL) configuration for Cisco NCS 1010, including ACL behavior, rule application, IPv4 and IPv6 tasks, and verification guidance.

Access control lists

This section explains access control lists and the key attributes you need to understand before configuring IPv4 or IPv6 packet filtering.

Access control lists is a Cisco NCS 1010 feature or capability that

- supports the related system setup or management workflow,
- defines behavior that affects Cisco NCS 1010 operation, and
- provides context before you configure or verify the feature.

Access control lists details
Table 16: Feature History

Feature Name	Release Information	Feature Description
ACL on Management Port	Cisco IOS XR Release 7.11.1	Access Control List (ACL) feature enables you to permit or deny specific devices to connect to the management port and access NCS 1010 devices. This control enhances network security. Both IPv4 and IPv6 ACLs are supported on the management port. Commands added: • **ipv4-access-list** • **ipv4-access-group** • **show access-lists-ipv4** • **ipv6-access-list** • **ipv6-access-group** • **show access-lists-ipv6**

Access Control Lists (ACLs) perform packet filtering to control the packets that move through the network. These controls allow to limit the network traffic and restrict the access of users and devices to the network. ACLs have many uses, and therefore many commands accept a reference to an access list in their command syntax. An ACL consists of one or more access control entries (ACE) that collectively define the network traffic profile. Access control entries (ACE) are entries in an ACL that describe the access rights related to a particular security identifier or user.

There are 2 types of ACLs:

- Standard ACLs-Verifies only the source IP address of the packets. Traffic is controlled by the comparison of the address or prefix configured in the ACL, with the source address found in the packet.
- Extended ACLs-Verifies more than just the source address of the packets. Attributes such as destination address, specific IP protocols, User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) port numbers, Differentiated Services Code Point (DSCP), and so on are validated. Traffic is controlled by a comparison of the attributes stated in the ACL with those in the incoming or outgoing packets.

Purpose of ACLs

ACLs allow you to perform the following:

- 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.

How access control lists work

This section describes how access control lists evaluate packet fields and apply matching permit or deny statements to control network traffic.

An ACL is a sequential list consisting of permit and deny statements that apply to IP addresses and upper-layer IP protocols. The ACL has a name by which it is referenced. Many software commands accept an ACL as part of their syntax.

An ACL can be configured and named; however, it does not take effect until the ACL is referenced by a command that accepts an ACL. Multiple commands can reference the same ACL. An ACL can control traffic arriving at the router or leaving the router, but not traffic originating at the router.

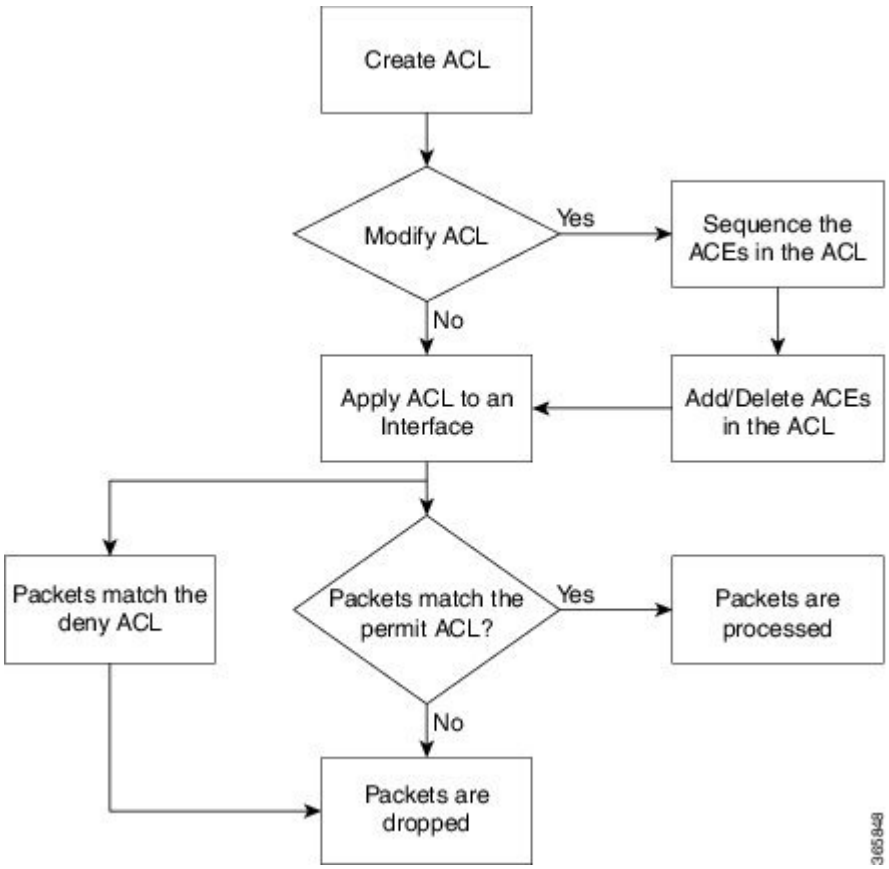
Source address and destination address are two of the most typical fields in an IP packet on which to base an ACL. Specify source addresses to control packets from certain networking devices or hosts. Specify destination addresses to control packets that are sent to certain networking devices or hosts.

You can also filter packets on the basis of transport layer information, such as whether the packet is a TCP, UDP, Internet Control Message Protocol (ICMP), or Internet Group Management Protocol (IGMP) packet.

ACL Workflow

The following image illustrates the workflow of an ACL.

Figure 17: ACL Workflow



Helpful Hints for Creating ACLs

Consider the following when creating ACLs:

Guidelines and Restrictions for Configuring ACLs

You must be aware of the following restrictions for configuring ACLs.

Summary

The key components involved in how access control lists work are:

- ACL entries: Define permit and deny statements for network traffic.
- Interfaces: Apply access lists to inbound or outbound traffic.
- Traffic flows: Match configured rules before the device permits or denies packets.

Workflow

The how access control lists work process involves the following stages:

1. An ACL is a sequential list consisting of permit and deny statements that apply to IP addresses and upper-layer IP protocols. The ACL has a name by which it is referenced. Many software commands accept an ACL as part of their syntax.
2. An ACL can be configured and named; however, it does not take effect until the ACL is referenced by a command that accepts an ACL. Multiple commands can reference the same ACL. An ACL can control traffic arriving at the router or leaving the router, but not traffic originating at the router.
3. Source address and destination address are two of the most typical fields in an IP packet on which to base an ACL. Specify source addresses to control packets from certain networking devices or hosts. Specify destination addresses to control packets that are sent to certain networking devices or hosts.

4. You can also filter packets on the basis of transport layer information, such as whether the packet is a TCP, UDP, Internet Control Message Protocol (ICMP), or Internet Group Management Protocol (IGMP) packet.
5. ACL Workflow
6. The following image illustrates the workflow of an ACL.
7. ACL Workflow
8. Helpful Hints for Creating ACLs

Result

The process determines how matching traffic is permitted, denied, or forwarded.

Access control list application behavior

This section provides details about access control list application behavior, including ACL references, inbound and outbound packet checks, and undefined ACL handling on terminal lines and network interfaces.

Review how Cisco IOS XR software applies access control lists to terminal lines and network interfaces after you create an ACL.

Access control list application behavior depends on these conditions.

- ACL creation and reference requirements.
- Inbound and outbound packet processing.
- Undefined ACL handling.

Table 17: ACL application behavior

If	Then
You create an ACL.	You must reference the ACL to make it work. You can apply an ACL to outbound or inbound interfaces. This behavior applies to terminal lines and network interfaces.
An inbound ACL receives a packet.	Cisco IOS XR software checks the source address of the packet against the ACL. If the ACL permits the address, the software continues to process the packet. If the ACL rejects the address, the software discards the packet and returns an ICMP host unreachable message. The ICMP message is configurable.
An outbound ACL receives a packet that is routed to a controlled interface.	Cisco IOS XR software checks the source address of the packet against the ACL. If the ACL permits the address, the software sends the packet. If the ACL rejects the address, the software discards the packet and returns an ICMP host unreachable message.
You apply an ACL that has not yet been defined to an interface.	Cisco IOS XR software acts as if the ACL has not been applied to the interface and accepts all packets. Note this behavior if you use undefined ACLs as a means of security in your network.

ACL configuration procedures

Provides links to ACL configuration procedures for Cisco NCS 1010, including IPv4, IPv6, extended ACL, and ACL modification tasks.

Use this reference to choose the correct ACL configuration task for the required traffic direction, IP version, or access-list operation.

The ACL configuration procedures include the following tasks.

- For inbound IPv4 filtering on the management Ethernet interface, see Configure an ingress IPv4 ACL on management Ethernet interface.
- For outbound IPv4 filtering on the management Ethernet interface, see Configure an egress IPv4 ACL on the management Ethernet interface.
- For inbound IPv6 filtering on the management Ethernet interface, see Configure an ingress IPv6 ACL on the management Ethernet interface.
- For outbound IPv6 filtering on the management Ethernet interface, see Configure an egress IPv6 ACL on the management Ethernet interface.
- For ACL rules that match destination address, protocol, port, or DSCP attributes, see Configure extended access lists.
- For updates to an existing ACL, see Modify ACLs.

Configure an ingress IPv4 ACL on management Ethernet interface

Use this procedure to configure an ingress IPv4 ACL on the management Ethernet interface so that inbound IPv4 traffic follows defined rules.

Configure an ingress IPv4 ACL on management Ethernet interface so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

Use the following configuration to configure an ingress IPv4 ACL on mgmtEth interface.

Before you begin

Follow these steps to configure an ingress IPv4 ACL on management Ethernet interface:

Procedure

1. Run the **configure** command to configure mgmtEth interface with an IPv4 address.

   Example

```
/* Configure mgmtEth interface with an IPv4 address */
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)#interface mgmtEth 0/RP0/CPU0/2
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 198.51.100.247 255.255.255.0
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
```

2. Run the **show ipv4 interface brief** command to verify if the interface is up.

Example

```
/* Verify if the interface is up */
RP/0/RP0/CPU0:ios(config)#do show ipv4 interface brief
Thu Oct 19 17:32:10.998 UTC

Interface                      IP-Address      Status          Protocol
Vrf-Name
Loopback0                   192.0.2.89      Up              Up       default

GigabitEthernet0/0/0/0       198.51.100.1    Up              Up       default

GigabitEthernet0/0/0/2       198.51.100.10   Up              Up       default

MgmtEth0/RP0/CPU0/0          198.51.100.247  Up              Up       default

PTP0/RP0/CPU0/0              unassigned      Shutdown        Down     default

MgmtEth0/RP0/CPU0/1          192.0.2.121     Up              Up       default

PTP0/RP0/CPU0/1              unassigned      Shutdown        Down     default

MgmtEth0/RP0/CPU0/2          192.0.2.1       Down            Down     default
```

3. Run the **ipv4 access-list V4-ACL-INGRESS** command to configure an IPv4 ingress 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
```

4. Run the **show access-lists ipv4** command to verify the ingress ACL creation.

Example

```
/* Verify the ingress ACL creation */
RP/0/RP0/CPU0:ios(config)# do 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
```

5. Run the **interface mgmtEth 0/RP0/CPU0/0** command to apply the ingress ACL to the mgmtEth 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
```

6. Run the **show ipv4 interface mgmtEth 0/RP0/CPU0/0** command to verify if the ingress ACL has been successfully applied to the interface.

Example

```
/* Verify if the ingress ACL has been successfully applied to the 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 198.51.100.247/16
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  common access list is not set, access list is V4-ACL-INGRESS
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000
```

You have successfully configured an IPv4 ingress ACL on the mgmtEth interface.

Configure an egress IPv4 ACL on the management Ethernet interface

Use this procedure to configure an egress IPv4 ACL on the management Ethernet interface so that outbound IPv4 traffic follows defined rules.

Configure an egress IPv4 ACL on the management Ethernet interface so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

Use the following configuration to configure an egress IPv4 ACL on the mgmtEth interface.

Before you begin

Follow these steps to configure an egress IPv4 ACL on the management Ethernet interface:

Procedure

1. Run the **configure** command to configure mgmtEth interface with an IPv4 address.

Example

```
/* Configure mgmtEth interface with an IPv4 address */
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 198.51.100.247 255.255.255.0
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
```

2. Run the **show ipv4 interface brief** command to verify if the interface is up.

Example

```
/* Verify if the interface is up */
RP/0/RP0/CPU0:ios(config)#do show ipv4 interface brief
Thu Oct 19 17:32:10.998 UTC

Interface                      IP-Address      Status          Protocol
Vrf-Name
Loopback0                     192.0.2.89      Up              Up       default

GigabitEthernet0/0/0/0         198.51.100.1    Up              Up       default

GigabitEthernet0/0/0/2         198.51.100.10   Up              Up       default

MgmtEth0/RP0/CPU0/0            198.51.100.247  Up              Up       default

PTP0/RP0/CPU0/0                unassigned      Shutdown        Down     default

MgmtEth0/RP0/CPU0/1            192.0.2.121     Up              Up       default

PTP0/RP0/CPU0/1                unassigned      Shutdown        Down     default

MgmtEth0/RP0/CPU0/2            192.0.2.1       Down            Down     default
```

3. Run the **ipv4 access-list V4-ACL-EGRESS** command to configure an IPv4 egress ACL.

Example

```
/* 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
```

4. Run the **show access-lists ipv4** command to verify the egress ACL creation.

Example

```
/* Verify the egress ACL creation */
RP/0/RP0/CPU0:ios(config)# do 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
...
```

5. Run the **interface mgmtEth 0/RP0/CPU0/0** command to apply the egress ACL to the mgmtEth interface.

Example

```
/* 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
```

6. Run the **show ipv4 interface mgmtEth 0/RP0/CPU0/0** command to verify if the egress ACL has been successfully applied to the interface.

Example

```
/* Verify if the egress ACL has been successfully applied to the 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 198.51.100.247/16
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is V4-ACL-EGRESS
  Inbound  common access list is not set, access list is not set
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000
```

You have successfully configured an IPv4 egress ACL on the mgmtEth interface.

Configure an ingress IPv6 ACL on the management Ethernet interface

Use this procedure to configure an ingress IPv6 ACL on the management Ethernet interface so that inbound IPv6 traffic follows defined rules.

Configure an ingress IPv6 ACL on the management Ethernet interface so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

Use the following configuration to configure an ingress IPv6 ACL on the mgmtEth interface.

Before you begin

Follow these steps to configure an ingress IPv6 ACL on the management Ethernet interface:

Procedure

1. Run the **required** command to configure a mgmtEth interface with an IPv6 address.

   Example

```
/* Configure a mgmtEth interface with an IPv6 address */
RP/0/RP0/CPU0:ios#
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
```

2. Run the **show ipv6 interface mgmtEth 0/RP0/CPU0/1 brief** command to verify if the interface is up.

   Example

```
/* Verify if the interface is up */
RP/0/RP0/CPU0:ios(config)#do 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
```

3. Run the **ipv6 access-list V6-INGRESS-ACL** command to configure an IPv6 ingress 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
RP/0/RP0/CPU0:ios(config-ipv6-acl)#exit
```

4. Run the **show access-lists ipv6** command to verify the ingress ACL creation.

   Example

```
/* Verify the ingress ACL creation */
RP/0/RP0/CPU0:ios(config)#do 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
```

5. Run the **interface mgmtEth 0/RP0/CPU0/1** command to apply the ingress ACL to the HundredGigE interface.

   Example

```
/* Apply the ingress ACL to the HundredGigE 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
```

6. Run the **show ipv6 interface mgmtEth 0/RP0/CPU0/1** command to verify if the ingress ACL has been successfully applied to the interface.

   Example

```
/* Verify if the ingress ACL has been successfully applied to the interface
*/
RP/0/RP0/CPU0:ios(config)#do show ipv6 interface mgmtEth 0/RP0/CPU0/1
Fri Oct 20 05:38:00.753 UTC
MgmtEth0/RP0/CPU0/1 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)

  IPv6 is enabled, link-local address is fe80::3afd:f8ff:fe66:872
  Global unicast address(es):
    2001::1, subnet is 2001::/64
  Joined group address(es): ff02::1:ff00:1 ff02::1:ff66:872 ff02::2
      ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND cache entry limit is 1000000000
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 160 to 240 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
  Outgoing access list is not set
  Inbound  common access list is not set, access list is V6-INGRESS-ACL
  Table Id is 0xe0800000
  Complete protocol adjacency: 0
  Complete glean adjacency: 0
  Incomplete protocol adjacency: 0
  Incomplete glean adjacency: 0
  Dropped protocol request: 0
  Dropped glean request: 0
  RA DNS Server Address Count: 0
  RA DNS Search list Count: 0
```

You have successfully configured an IPv6 ingress ACL on the mgmtEth interface.

Configure an egress IPv6 ACL on the management Ethernet interface

Use this procedure to configure an egress IPv6 ACL on the management Ethernet interface so that outbound IPv6 traffic follows defined rules.

Configure an egress IPv6 ACL on the management Ethernet interface so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

Use the following configuration steps to configure an egress IPv6 ACL on the mgmtEth interface.

Before you begin

Follow these steps to configure an egress IPv6 ACL on the management Ethernet interface:

Procedure

1. Run the **interface mgmtEth 0/RP0/CPU0/1** command to configure a mgmtEth interface with an IPv6 address.

   Example

```
/* Configure a mgmtEth interface with an IPv6 address */
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
Thu Jan 25 11:41:25.778 IST
Router(config-if)# exit
```

2. Run the **show ipv6 interface brief** command to verify if the interface is up.

   Example

```
/* Verify if the interface is up */
Router(config)# do show ipv6 interface brief
Thu Jul 11 09:47:50.812 UTC
HundredGigE 0/0/0/0 [Up/Up]
    fe80::bd:b9ff:fea9:5606
    1001::1
HundredGigE 0/0/0/1 [Up/Up]
    fe80::23:e9ff:fea8:a44e
    2001::1
```

3. Run the **ipv6 access-list V6-EGRESS-ACL** command to configure an IPv6 egress ACL.

   Example

```
/* 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
Router(config-ipv6-acl)# exit
```

4. Run the **show ipv6 interface mgmtEth 0/RP0/CPU0/1 brief** command to verify the egress ACL creation.

Example

```
/* Verify the egress ACL creation */
RP/0/RP0/CPU0:ios(config)#do 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
…
```

5. Run the **interface mgmtEth 0/RP0/CPU0/1** command to apply the egress ACL to the mgmtEth interface.

Example

```
/* 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
```

6. Run the **show ipv6 interface mgmtEth 0/RP0/CPU0/1** command to verify if the egress ACL has been successfully applied to the interface.

Example

```
/* Verify if the egress ACL has been successfully applied to the interface
*/
RP/0/RP0/CPU0:ios(config)#do show ipv6 interface mgmtEth 0/RP0/CPU0/1
Fri Oct 20 05:38:00.753 UTC
MgmtEth0/RP0/CPU0/1 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)

  IPv6 is enabled, link-local address is fe80::3afd:f8ff:fe66:872
  Global unicast address(es):
    2001::1, subnet is 2001::/64
  Joined group address(es): ff02::1:ff00:1 ff02::1:ff66:872 ff02::2
      ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND cache entry limit is 1000000000
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 160 to 240 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
  Outgoing access list is V6-EGRESS-ACL
  Inbound  common access list is not set, access list is not set
  Table Id is 0xe0800000
  Complete protocol adjacency: 0
  Complete glean adjacency: 0
  Incomplete protocol adjacency: 0
  Incomplete glean adjacency: 0
  Dropped protocol request: 0
  Dropped glean request: 0
  RA DNS Server Address Count: 0
```

```
  RA DNS Search list Count: 0
...
```

You have successfully configured an IPv6 egress ACL on the mgmtEth interface.

Configure extended access lists

Use this procedure to configure extended access lists on Cisco NCS 1010 so that packet filtering can match source, destination, and protocol fields.

Configure extended access lists so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

To configure Extended Access Lists, you must create an access list and specify the condition to allow or deny the network traffic.

Use Extended Access Lists to verify more than just the source address of the packets. Attributes such as destination address, specific IP protocols, UDP or TCP port numbers, DSCP, and so on are validated. Traffic is controlled by a comparison of the attributes stated in the ACL with those in the incoming or outgoing packets.

Running Configuration

Verification

Before you begin

Follow these steps to configure extended access lists:

Procedure

1. Run the **configure** command to enter the global configuration mode and create the access list.

   Example

```
/* Enter the global configuration mode and create the access list*/
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ipv4 access-list acl_1
RP/0/RP0/CPU0:ios(config-ipv4-acl)#10 remark Do not allow user1 to telnet out
/*Specify the condition to allow or deny the network traffic.*/
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
```

2. Run the **show running-config** command to verify the configuration.

   Example

```
RP/0/RP0/CPU0:ios#show running-config
Fri Oct 20 06:21:11.024 UTC
!! Building configuration...
!! IOS XR Configuration 24.1.1.23I
!! Last configuration change at Fri Oct 20 06:19:08 2023 by cisco

!
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
!
```

3. Run the **show access-lists ipv4 acl_1** command to verify the configuration.

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
```

The system completes the configure extended access lists task.

Modify ACLs

Use this procedure to modify ACLs on Cisco NCS 1010 so that existing packet-filtering rules match current traffic-control requirements accurately.

Configure modify ACLs so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

This section describes a sample configuration to modify ACLs.

/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/

Before you begin

Follow these steps to modify ACLs:

Procedure

---

1. Run the **configure** command to complete this action.

Example

```
*/ Create an Access List*/
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ipv4 access-list acl_1

*/Add entries (ACEs) to the ACL*/
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
```

2. Run the ***/Verify the entries of the ACL*/:** command to complete this action.

Example

```
*/Verify the entries of the ACL*/:
Router#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
```

3. Run the **ipv4 access-list acl_1** command to complete this action.

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
```

4. Run the **show access-lists ipv4 acl_1** command to verify the configuration.

Example

```
*/Verify the entries of the ACL:*/
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 ACLs in operation.

14 Remote Node Management

Topics:

- Remote node management details for NCS 1010
- Remote node management with OSC
- Remote node management prerequisites
- DHCP relay configuration for OLT node
- Loopback IP address for OSC interface
- OSPF neighbor discovery
- ILA node configuration
- OLT node configuration

This chapter describes remote node management in NCS 1010. It helps readers use related concepts, references, and procedures for managing remote nodes.

Remote node management details for NCS 1010

This section provides details about remote node management in NCS 1010, including context that helps readers use related concepts, references, and procedures.

Use this reference to review remote node management in NCS 1010.

The following information supports remote node management in NCS 1010:

- This chapter describes how to manage an ILA node remotely in NCS 1010 by the remote node management feature.

Remote node management with OSC

This section explains remote node management using OSC for Cisco NCS 1010. Use it to understand the feature behavior, requirements, and setup context before you configure or verify remote node management.

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.

Remote node management prerequisites

This section provides details about remote node management prerequisites for Cisco NCS 1010. Use it to review the required conditions before you configure remote node management.

Use this reference to review 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 220
- The loopback address and IP address must be present for OSC interface. See Loopback IP address for OSC interface on page 222
- The OSPF neighbor discovery must be successful between OLT and ILA nodes. See OSPF neighbor discovery on page 222

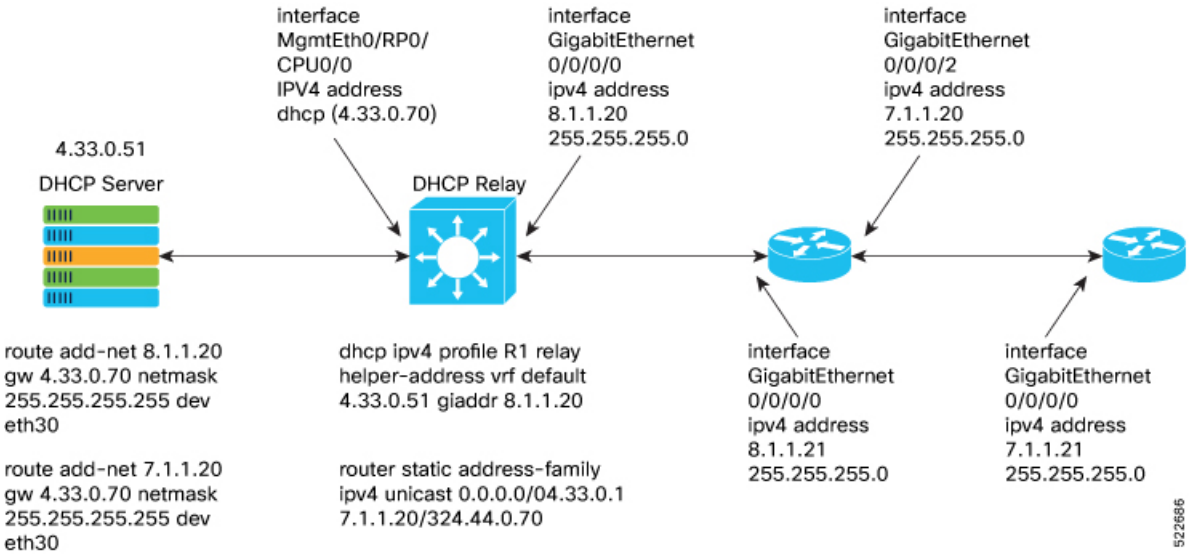
DHCP relay configuration for OLT node

This section explains the DHCP relay configuration for an OLT node. Use it to understand the feature behavior, requirements, and setup context before you configure or verify remote node management.

The OLT node must be configured with the DHCP management connection to manage the ILA node remotely over OSC interface.

Details

Figure 18: 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

This section explains the loopback IP address for the OSC interface. Use it to understand the feature behavior, requirements, and setup context before you configure or verify remote node management.

The loopback IP address must be mapped for the OSC interface.

Details

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

This section explains OSPF neighbor discovery for Cisco NCS 1010. Use it to understand the feature behavior, requirements, and setup context before you configure or verify remote node management.

The OSPF neighbor discovery indicates the successful connection between OLT and ILA node.

Details

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

ILA node configuration

This section provides details about configuring an ILA node for Cisco NCS 1010. Use it to review the command example and settings that support remote node management.

Use this reference to review configure ILA node.

The following information supports 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
```

OLT node configuration

This section provides details about configuring an OLT node for Cisco NCS 1010. Use it to review the command example and settings that support remote node management.

Use this reference to review 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
```

-

15 Remote Console Connection Workflow

Topics:

- Remote console connection details
- Remote console connection support for Cisco NCS 1010
- Find the MAC address of all the nodes
- Connect to a remote node

This chapter describes remote console connection for Cisco NCS 1010. It helps readers use related concepts, references, and procedures for remote access.

Remote console connection details

This section provides details about remote console connection for Cisco NCS 1010, including context that helps readers use related concepts, references, and procedures.

Use this reference to review remote console connection.

The following information supports remote console connection:

- Use this reference to review remote console connection behavior and related topics for Cisco NCS 1010.

Remote console connection support for Cisco NCS 1010

This section provides details about remote console connection for Cisco NCS 1010. Use it to review the requirements, behavior, examples, or limitations that support remote console connection.

Use this reference to review remote console connection.

You can access the remote nodes only when you enable the RCOM interface on all the nodes. The RCOM interface remains shut by default. Remote console enables you to access the remote nodes and configure them.

- Benefits of remote console connections

  Remote console connections help

  - to access the remote nodes information even if intermediate nodes are in headless mode, and
  - to troubleshoot the remote nodes that don't have physical console connection.

- Limitations of remote console connections

  Remote console connection cannot be established if

  - the peer MAC addresses are missing,
  - hostname has changed, or
  - RCOM interface is in Shutdown state on each node

Table 18: Feature History

Feature Name	Release Information	Feature Description
Remote Console Connection	Cisco IOS XR Release 24.4.15	The Remote Console Connection enables you to connect to remote OLT and ILA nodes within the network using the OSC GigabitEthernet interface. This functionality allows you to access detailed information about the remote nodes through the near-end nodes by enabling the RCOM interface that is disabled by default. It provides essential access for gathering information and troubleshooting the remote nodes, even when intermediate nodes are in headless mode or lack a physical console connection. Establish connetion to the remote nodes through their hostname or MAC address. The commands that enable remote connection are: • **remote-connect hostname <hostname>** • **remote-connect mac <mac-address>** • **show remote-connect neighbours**

Find the MAC address of all the nodes

Use this procedure to find the MAC address of all nodes. It supports remote console setup in an OLT-ILA-OLT topology before connection configuration.

Use this task to find the MAC address of all the nodes.

> **Note**
>
> This procedure considers that you are establishing console connection to a remote ILA node in a 3-node OLT-ILA-OLT topology. The output changes for different topologies.

Before you begin

Enable LLDP in all nodes in the network. See Link Layer Discovery Protocol Support on Management Interface.

Follow these steps to find the MAC address of all the nodes.

Procedure

1. Unshut the RCOM interface in all the nodes to prepare for remote console connection.

Example
For security reasons, console access to the remote nodes are disabled by default. To access the remote node's console, enable the RCOM management interface.

```
RP/0/RP0/CPU0:ios(config)#interface MgmtEth 0/RP0/RCOM0/0
RP/0/RP0/CPU0:ios(config-if)#no shutdown
RP/0/RP0/CPU0:ios(config-if)#commit
Wed Sep  4 11:40:14.864 IST
```

> **Note**
>
> Unshutting the RCOM interface enables only the LLDP receive operations to receive remote node details.

2. Run **end** to exit the RCOM interface.

Example

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

3. Run **show ipv4 interface brief** to verify the RCOM interface status.

Example
The output, displaying the node's IPv4 interface details, highlights the *UP* status of the node's RCOM interface.

```
RP/0/RP0/CPU0:ios#show ipv4 interface brief
Wed Sep  4 11:40:23.460 IST

Interface                      IP-Address      Status          Protocol
Vrf-Name
Loopback0                      10.1.1.2        Up              Up       default

GigabitEthernet0/0/0/0         10.1.1.2        Up              Up       default

GigabitEthernet0/0/0/2         10.1.1.2        Up              Up       default

MgmtEth0/RP0/CPU0/0            4.34.2.200      Up              Up       default

PTP0/RP0/CPU0/0                unassigned      Shutdown        Down     default

MgmtEth0/RP0/CPU0/1            unassigned      Shutdown        Down     default

PTP0/RP0/CPU0/1                unassigned      Shutdown        Down     default

MgmtEth0/RP0/CPU0/2            10.127.126.174  Shutdown        Down     default

MgmtEth0/RP0/RCOM0/0           unassigned      Up              Up       default
```

4. Run **show lldp neighbors** or **show lldp neighbors details** to find the MAC addresses of all the nodes in the topology.

> **Note**
>
> You can find the details of only the nodes that are connected to the OLT node through OSC gigabit ethernet interface.

Example
The **show lldp neighbors** output, displaying the neighbors details of the near-end OLT node, highlights the hostname of its neighbor nodes.

```
RP/0/RP0/CPU0:ios#show lldp neighbors
Thu Mar 20 16:19:15.151 IST
Capability codes:
        (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
        (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID       Local Intf                     Hold-time  Capability
Port ID
R4              GigabitEthernet0/0/0/0         120        R
GigabitEthernet0/0/0/0
R5              GigabitEthernet0/0/0/2         120        R
GigabitEthernet0/0/0/0

Total entries displayed: 2
```

Example
The **show lldp neighbors details** output displays the system name and its peer MAC address in detail.

```
RP/0/RP0/CPU0:ios#show lldp neighbors details
Thu Mar 20 16:19:17.010 IST
Capability codes:
        (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
        (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

------------------------------------------------
Local Interface: GigabitEthernet0/0/0/0
Chassis id: cced.4dda.9ebe
Port id: GigabitEthernet0/0/0/0
Port Description - not advertised
System Name: R4

System Description:
24.4.15.18I, NCS1010

Time remaining: 108 seconds
Hold Time: 120 seconds
Age: 264651 seconds
System Capabilities: R
Enabled Capabilities: R
Management Addresses:
  IPv4 address: 10.1.1.1

Peer MAC Address: cc:ed:4d:da:9e:c2


------------------------------------------------
Local Interface: GigabitEthernet0/0/0/2
```

```
Chassis id: 689e.0bb8.7122
Port id: GigabitEthernet0/0/0/0
Port Description - not advertised
System Name: R5

System Description:
24.4.15.18I, NCS1010

Time remaining: 92 seconds
Hold Time: 120 seconds
Age: 290600 seconds
System Capabilities: R
Enabled Capabilities: R
Management Addresses:
  IPv4 address: 10.1.1.3

Peer MAC Address: 68:9e:0b:b8:71:26


Total entries displayed: 2
```

Connect to a remote node

Use this procedure to connect to a remote node. It supports remote console access through the near-end OLT node in supported topologies.

Use this task to connect to a remote node.

Follow the steps to establish a connection to a remote node through the near-end OLT node.

Before you begin

LLDP must be enabled in all nodes in the network. See Link Layer Discovery Protocol Support on Management Interface.

RCOM interface must be in *UP* status.

Follow these steps to connect to a remote node.

Procedure

1. Run **remote-connect** with **mac** or **hostname** of the remote node to establish connection to it.

   Example
   If you have the hostname of the destination node, use `remote-connect hostname <hostname>`.

   ```
   RP/0/RP0/CPU0:ios#remote-connect hostname R5
   ```

   In this example, *R5* is the hostname of the remote node.

   Example
   If you have the MAC address of the destination node, use `remote-connect mac <mac-address>`.

   ```
   RP/0/RP0/CPU0:ios#remote-connect mac 98:a2:c0:34:3f:f3
   ```

In this example, *98:a2:c0:34:3f:f3* is the MAC address of the remote node.

2. Run **show remote-connect neighbours** to verify the connections between the remote nodes and the OLT node.

   Example
   The output highlights RCOM interface, hostname, and mac-address of the remote nodes that are connected to the OLT node.

```
RP/0/RP0/CPU0:ios#show remote-connect neighbours
Fri Oct 25 12:08:08.905 IST
Remote console neighbour details
Records : 2

Interface               ChassisId       Mac                 Hostname
MgmtEth0/RP0/RCOM0/0    FCxxxxxxx58     f8:39:18:3e:79:1b   R5
MgmtEth0/RP0/RCOM0/0    FCxxxxxxxWY     68:9e:0b:b8:71:1e   R4
```

3. (Optional) Run the **remote-connect timeout** command to set the timeout value for a remote connection session.

   Example
   If you want to set a timeout value, use `remote-connect hostname <hostname> timeout <1-60>`.

```
RP/0/RP0/CPU0:ios#remote-connect hostname PROD2 timeout 3
```

   In this example, the timeout value is set to *3 minutes*.

> **Note**
>
> The default timeout value is 15 minutes. The timeout range is 1 to 60 minutes.

4. (Optional) Run the **remote-connect verbose** command to collect logs for the remote connection between the near-end and far-end node.

   Example

```
RP/0/RP0/CPU0:ios#remote-connect mac 98:a2:c0:34:3f:f3 verbose
Wed Mar 19 15:15:59.436 IST
Verbose log enabled, log stored at : /var/log/remote_console/rcon.log ...
Welcome to Remote Console! Press Ctrl-Z to end session
Timeout set as : 15 min
```

   In this example, **verbose** is enabled for the *98:a2:c0:34:3f:f3* mac-aaddress.

16 Automated File Management

Topics:

- Automated file management behavior

This section provides details about automated file management for Cisco NCS 1010. Use it to review the requirements, behavior, examples, or limitations that support automated file management.

Use this reference to review automated file management.

The following information supports automated file management:

- Use this reference to review automated file management behavior and related topics for Cisco NCS 1010.

Automated file management behavior

Provides details about automated file management on Cisco NCS 1010, including feature history, SSD file types, log file rotation, showtech residual file cleanup, and OTDR file cleanup.

Use this reference to understand how the automated file management system archives older files and removes files from local nodes to free SSD space.

Table 19: Feature History

Feature Name	Release Information	Feature Description
Automated File Management System	Cisco IOS XR Release 24.4.1	The Automated File Management System handles files efficiently on each node. The system automatically archives older files and removes them from local nodes to free SSD space. It manages system-generated log files, showtech-related residual files, and OTDR files.

The SSD stores these file types.

- **User**: The user creates and owns files for requirements and deletes the files when the files are no longer needed.
- **System**: The system organizes files automatically based on the file content and the application that created the content, such as .log files and showtech-related residual files. Automated file management applies to system-generated files.

Automated file management handles log files in these stages.

- The NCS 1014 system uses log rotation configurations to manage log files as required.
- The system checks for *.log* files that exceed 10 MB. This threshold applies only to files in the `/tmp` folder. For files in other folders, the system uses a different threshold and applies the same process.
- After locating the file, the system archives that file with the .gz extension and creates a new .log file with the same name.
- The system maintains one active log file and two archived files.
- When new files are archived, the system deletes the oldest archived file so that only the two most recently archived files and the current log file are retained.

Automated file management handles showtech-related residual log files in these stages.

- The system generates residual files during log collection when **Showtech logs** is in operation.
- After collecting the logs, the system automatically removes the residual files to conserve space.

17 Audit Log Storage and Monitoring

Topics:

- Audit logging and monitoring details
- Audit logs
- How audit logging works
- Guideline: Use audit logging
- Note: Review audit log storage behavior
- Configure audit logging
- AuditD process keepalive

This chapter describes audit logging and monitoring for Cisco NCS 1010. It helps readers use audit concepts, storage notes, monitoring guidance, and configuration procedures.

Audit logging and monitoring details

This section provides details about audit logging and monitoring for Cisco NCS 1010, including context for audit concepts, storage notes, monitoring guidance, and configuration procedures.

Use this reference to review implementing audit monitoring.

The following information supports implementing audit monitoring:

- TThis chapter explains the audit monitoring and logging capabilities available on NCS 1010 and how to configure audit monitoring.

Feature name	Release information	Description
Audit logging and monitoring	Cisco IOS XR 25.3.1	You can enable audit logging and monitoring on the NCS 1010. You can also configure predefined rule groups that allow NCS 1010 to monitor activities, log events, and, when necessary, forward audit logs to a remote syslog server for centralized analysis and incident response. This feature helps enhance security and compliance on your network. CLI: These new commands are introduced: • **linux security audit monitor** *group-keyword* • **show linux security audit monitor status** • **linux security audit logging syslog** • • **logging** *remote-server-ip* **vrf** *remote-server-ip* • **show linux security audit logging syslog**

Audit logs

This section explains audit logging for Cisco NCS 1010. Use it to understand audit rules, security-event records, local log storage, and optional forwarding to a remote syslog server.

Audit logging is a security and compliance feature that

- operates according to defined audit rules automatically creating audit logs whenever specified actions or changes occur on the router
- integrates with the Linux Audit Daemon to monitor and log relevant security events across the router, and

- allows forwarding of audit logs to a remote syslog server.

Linux audit daemon is a user-space component of the Linux auditing system that

- tracks and logs system calls, file accesses, user actions, and other events as specified by audit rules, and
- provides administrators with insights to detect suspicious behavior and maintain system integrity.

An audit rule is a configuration that

- specifies which files, directories, or system events should be monitored
- determines the conditions for monitoring, and
- forms the foundation of an audit logging system.

An Audit log is a chronological record that

- is automatically generated when a monitored event, as defined by an audit rule, occurs, and
- typically includes details such as the event type, timestamp, user or process involved, and affected resources.

Audit rules and audit logs for security monitoring

Administrators define audit rules to track changes to sensitive files, monitor system calls, and observe other critical activities. By customizing audit rules, organizations can align monitoring with their unique security and compliance requirements.

Audit rules establish what to watch, while audit logs capture and document every relevant occurrence, ensuring a complete and actionable history of system activity.

For example, an audit rule that monitors changes to /etc/passwd file creates an audit log entry each time this file is modified.

Audit logging is not to be confused with system logging. While audit logging records security-relevant events, such as user actions and changes to sensitive files, system logging (syslog) captures general system events like service status updates, routine errors, or informational messages.

How audit logging works

This section describes how audit logging works on Cisco NCS 1010, including key stages, system behavior, and outcomes that support setup or management workflows.

Summary

These are the key components involved in this feature:

- Network Administrator: The user who initiates configurations via CLI.
- Linux audit daemon : The process that monitors system activity according to the installed rules and writes audit event logs.
- Local rsyslog daemon: The process that forwards logs to a remote syslog server.
- Remote syslog server: The external server that maintains the logs generated by the router.

The Linux audit daemon is the core service that actually performs event monitoring and logging, based on the audit rules configured by the administrator. It operates at the operating system level on each node, such as line cards and processors.

Workflow

These stages describe how audit monitoring and logging works.

1. The network administrator enables audit monitoring via CLI.
2. The router software receives the configurations, applies the relevant audit rules, and ensures these rules are distributed to all appropriate nodes.
3. On each node, the Linux audit daemon actively monitors system events as defined by the audit rules and writes the logs to a local log file at **/var/log/audit/audit.log**.
4. If the network administrator has enabled log forwarding, the audit logs are sent to the local rsyslog daemon, which then forwards the logs to a remote syslog server.

Guideline: Use audit logging

This section outlines audit logging guidance for Cisco NCS 1010. Use it to apply the required recommendation when you configure, monitor, or maintain audit logging and monitoring.

Granularity of audit rules

- You can enable or disable audit rules only at the group level, not individually within a group.
- Regularly review the status of audit rules and audit log forwarding to ensure monitoring remains effective.

Resource usage on NCS 1010

Use caution when enabling all rule groups, especially those that monitor frequent events, as this may increase CPU, memory, or disk usage. Enable only the groups required for compliance or security needs.

Security of audit logs and syslog servers

- Allow only users with appropriate administrative privileges to configure or view Linux security audit settings.
- Protect access to audit logs and syslog servers to prevent unauthorized access or tampering.

Log forwarding to remote syslog servers

- Confirm that the remote syslog server is reachable and properly configured before enabling log forwarding.
- NCS 1010 forwards audit logs to remote syslog servers in unencrypted plain text. Use only trusted network segments for remote syslog servers.

Note: Review audit log storage behavior

This section outlines audit log storage guidance for Cisco NCS 1010. Use it to understand local audit log storage behavior and when to forward audit logs to a remote syslog server.

- NCS 1010 stores audit logs locally at **/var/log/audit/audit.log** , unless you enable log forwarding.
- By default, the system rotates up to five audit log files, each up to 8 MB in size.

Configure audit logging

Use this procedure to configure audit logging. It supports audit logging and monitoring by outlining the required actions, inputs, or verification points for the activity.

Follow this task to configure and monitor audit logs for specific system events by enabling the relevant audit rule groups.

This task supports Cisco NCS 1010 setup, deployment, upgrade, or maintenance workflows.

Before you begin

Follow these steps to configure audit logging.

Procedure

1. Run the **linux security audit monitor <group-keyword>** command to enable a group of audit rules.

   Example

```
RP/0/RP0/CPU0:ios# configure
RP/0/RP0/CPU0:ios(config)# linux security audit monitor xr-software
RP/0/RP0/CPU0:ios(config)# linux security audit monitor user-group-config-files
RP/0/RP0/CPU0:ios(config)# commit
```

2. Run the **show linux security audit monitor status** command, to verify the general status of all active audit rule groups.

   Example

```
RP/0/RP0/CPU0:ios# show linux security audit monitor status
Wed Aug 20 16:16:23.518 IST
key name: xr-software                  status: enabled
rules:
-a always,exit -F arch=b64 -F dir=/pkg/bin -F perm=wa -k xr_bin_changes
-a always,exit -F arch=b64 -F dir=/pkg/sbin -F perm=wa -k xr_sbin_changes
-a always,exit -F arch=b64 -F dir=/pkg/lib -F perm=wa -k xr_lib_changes
------------------------------------------------------------
key name: user-group-config-files      status: enabled
rules:
-a always,exit -F arch=b64 -F path=/etc/passwd -F perm=wa -k passwd_changes
-a always,exit -F arch=b64 -F path=/etc/shadow -F perm=wa -k shadow_changes
-a always,exit -F arch=b64 -F path=/etc/group -F perm=wa -k group_changes
-a always,exit -F arch=b64 -F path=/etc/sudoers -F perm=wa -k sudoers_changes
------------------------------------------------------------
```

3. (Optional) Run the **linux security audit logging syslog** command to enable forwarding of audit rules.

   Example

```
RP/0/RP0/CPU0:ios# configure
RP/0/RP0/CPU0:ios(config)# linux security audit logging syslog
RP/0/RP0/CPU0:ios(config)# commit
```

4. (Optional) Run the **logging remote-server-ip vrf vrf-name** command to configure the remote syslog server.

   Example

```
RP/0/RP0/CPU0:ios# configure
RP/0/RP0/CPU0:ios(config)# logging 10.0.1.2 vrf default severity info port
default facility local6
RP/0/RP0/CPU0:ios(config)# commit
```

5. (Optional) Run the **show linux security audit logging syslog** command, to verify whether audit log forwarding is enabled and to view the configured remote syslog server.

   Example

```
RP/0/RP0/CPU0:ios# show linux security audit logging syslog
Wed Aug 20 16:16:44.553 IST
status: enabled
syslog-server(s):
ipaddr: 10.0.1.2 vrf: vrf-default port: 514
ipaddr: 10.0.1.9 vrf: vrf-default port: 514
```

AuditD process keepalive

Explains how AuditD process keepalive events verify the liveness of the AuditD service and audit pipeline on NCS 1010, improving compliance, enabling early detection of silent failures, and supporting large-scale monitoring across distributed fleets.

AuditD process keepalive is a NCS 1010 capability that

- verifies the health of the Linux AuditD service by periodically generating keepalive events
- confirms that the audit logging path remains operational, and
- provides a positive audit signal for compliance and early detection of silent failures.

Table 20: Feature History Table

Feature name	Release information	Feature description
AuditD process keepalive	Cisco IOS XR 26.3.1	AuditD keepalive periodically generates an identifiable Linux audit event at a configurable interval between 10 and 43,200 minutes. Monitoring systems can use these events to verify AuditD health and detect interruptions in audit-event generation or delivery. Additionally, the feature provides automated syslog notifications for service state changes across all nodes. This proactive monitoring eliminates silent failures, ensuring continuous compliance and enhanced operational visibility for large-scale deployments. CLI keywords introduced: The keywords **keepalive** and **interval** are added to the **linux security audit** command.

AuditD process keepalive

Starting with Release 26.3.1, NCS 1010 supports an AuditD keepalive capability. When you enable keepalive, the system periodically generates an identifiable Linux audit event at a user-configurable interval using the **linux security audit keepalive interval** *interval* command. This keepalive event verifies that AuditD process is running and the audit log forwarding remains operational, improving detection of silent failures and strengthening auditability across distributed system fleets.

AuditD process keepalive produces periodic, identifiable Linux audit events that act as heartbeat signals for monitoring systems, helping detect failures in audit-event generation or delivery.

AuditD process keepalive works independently of remote syslog forwarding. If remote syslog forwarding is enabled, NCS 1010 forwards keepalive events to the configured remote syslog server. If remote syslog forwarding is not enabled, NCS 1010 stores the keepalive events in the local audit log.

Each keepalive event is written to the AuditD log as an identifiable record. You can use this record to confirm that:

- AuditD is running
- the audit event path is working
- audit events continue to reach the remote syslog server, if forwarding is enabled

Benefits of AuditD process keepalive

These are key benefits of the AuditD process keepalive:

- Confirms AuditD liveness automatically without manual device checks.
- Helps detect silent AuditD or log-pipeline failures earlier.
- Supports large-scale monitoring across distributed system fleets.
- Improves compliance verification by providing a positive audit signal.
- Allows operators to configure hearbeat log frequency.

Guidelines for AuditD process keepalive

Outlines guidelines for configuring and monitoring the AuditD process keepalive feature on NCS 1010.

Follow these guidelines when deploying AuditD process keepalive:

- Configure the keepalive interval in minutes. The supported range is 10 to 43200 minutes.
- Filter on the `audit_keepalive` key in your monitoring tools so that you can identify keepalive events separately from operational audit events.
- Use the `location` field in the audit log, when present, to identify the source node for a keepalive event. This field helps you distinguish events from different route processor, line card, sysadmin, or host locations.
- Enable AuditD process keepalive explicitly. The feature is not active until you enable it.

Restrictions for AuditD process keepalive

Outlines key restrictions for deploying AuditD process keepalive on NCS 1010, including feature scope, server configuration, and device compatibility.

Always remember these restrictions when deploying AuditD process keepalive:

- AuditD process keepalive verifies AuditD liveness. It does not replace general device health monitoring.
- Keepalive event delivery to a remote server depends on remote syslog forwarding being configured and operational.
- If remote forwarding is not configured, the feature still generates keepalive events, but NCS 1010 stores them locally.
- Verify platform and release availability before deployment.

Configure the AuditD process keepalive

Use this procedure to configures AuditD process keepalive on NCS 1010 and verify that keepalive audit events are generated as expected.

Ensure NCS 1010 AuditD generates keepalive audit events at a configured interval and optionally forwards them to a remote syslog server for monitoring.

Use this task to maintain the integrity of audit logging by confirming periodic keepalive events. You can forward these events to a remote syslog server or store them locally for review.

Before you begin

- Decide how often you want NCS 1010 to generate keepalive events.
- If you want to forward keepalive events to a remote syslog server, configure the remote syslog destination on the system.

Follow these steps to configure AuditD process keepalive:

Procedure

1. Enable AuditD process keepalive and set the interval.

   Example

   ```
   RP/0/RP0/CPU0:ios# linux security audit keepalive interval 1010
   ```

   To disable AuditD process keepalive, use the **no linux security audit keepalive interval 1010** command.

2. Enable audit log forwarding.

   Example

   ```
   RP/0/RP0/CPU0:ios# linux security audit logging syslog
   ```

3. Configure the remote syslog destination.

   Example

   ```
   RP/0/RP0/CPU0:ios#  logging 10.0.1.2 vrf default port 514
   RP/0/RP0/CPU0:ios(config)# commit
   ```

4. If you enabled remote forwarding, verify the forwarding status.

   Example

   ```
   RP/0/RP0/CPU0:ios#  show linux security audit logging syslog
   Mon Aug 10 10:18:01.703 IST
   status: disabled
   syslog-server(s):
   ipaddr: 10.0.1.2 vrf: vrf-default port: 514
   ```

5. Confirm that keepalive events are generated:

- Check on the remote syslog server if remote forwarding is enabled and the following syslogs are generated.

```
#Audit configuration syslog
<182>1 2026-03-29T12:41:19.653359+05:30 NCS1020_B0_DT_01
SECURITY-AUDIT-6-EVENT - - - type=CONFIG_CHANGE
msg=audit(1774768279.648:2200): auid=4294967295 ses=4294967295
subj=system_u:system_r:auditctl_t:s0 op=add_rule key="audit_keepalive"
list=4 res=1#035AUID="unset”

#AuditD keepalive interval syslog
<182>1 2026-03-29T12:41:29.105188+05:30 NCS1020_B0_DT_01
SECURITY-AUDIT-6-EVENT - - - type=SYSCALL msg=audit(1774768289.101:2235):
 arch=c000003e syscall=87 success=yes exit=0 a0=7fe4a994a60c a1=7fe4a994a47b
 a2=1 a3=6 items=2 ppid=8348 pid=13338 auid=4294967295 uid=0 gid=1000
euid=0 suid=0 fsuid=0 egid=1000 sgid=1000 fsgid=1000 tty=(none)
ses=4294967295 comm="xr_auditd"
exe="/opt/cisco/install-iosxr/base/bin/xr_auditd"
subj=system_u:system_r:iosxr_t:s0 key="audit_keepalive"#035ARCH=x86_64
SYSCALL=unlink AUID="unset" UID="root" GID="iosxr" EUID="root" SUID="root"
 FSUID="root" EGID="iosxr" SGID="iosxr" FSGID="iosxr”
```

> **Note**
>
> By default, the syslogs are generated in `/var/log/syslog/`. Otherwise, the syslogs are generated in the path configured on the remote syslog server.

- Check in the local audit log at `/var/log/audit/audit.log` if remote forwarding is not enabled.

NCS 1010 generates periodic keepalive audit events at the configured interval.

What to do next

Configure your monitoring system to alert when expected `audit_keepalive` events stop arriving for a device or location.

18 Process Memory Management

Topics:

- Process memory management details
- Core dump folders

This chapter describes process memory management for Cisco NCS 1010. It helps readers use core dump storage concepts, references, and procedures.

Process memory management details

This section provides details about process memory management for Cisco NCS 1010, including context that helps readers use core dump storage concepts, references, and procedures.

Use this reference to review process memory management.

Process memory management refers to the efficient allocation and isolation of memory for individual processes within the modular operating system. Each process runs in its own protected memory space to ensure stability and prevent interference.

- NCS 1014 has dynamic memory allocation, leveraging segments like code, data, stack, heap, and shared memory for process execution.

Table 21: Feature History Table

Feature Name	Release	Description
Core dump folder limit	Cisco IOS-XR Release 26.1.1	You can now set disk storage limits to 20% to prevent excessive core dumps, and a new folder limit retains up to 20 coreinfo files. The system automatically cleans up every 15 minutes, ensuring efficient use of disk space and preserving resources for critical operations. This process provides better disk management and enhanced system reliability with configurable storage limits and automated cleanup. **CLI**: • The command **exception disk-usage-limit** is introduced. **YANG data model:** • `Cisco-IOS-XR-sysadmin-dumper.yang`

Core dump folders

This section explains core dump folders in Cisco IOS XR. Use it to understand storage attributes, system limits, and troubleshooting features for diagnostic files.

A core dump folder is a directory in the Cisco IOS XR file system that

- stores diagnostic files, known as **core dumps**, which capture the state of the system's memory during a crash or failure,
- is typically located at `/misc/disk1/coredumps`, and
- provides critical information for troubleshooting and debugging system issues.

To manage storage and maintain system reliability, Cisco IOS XR introduces several mechanisms related to core dump folder limits:

- Configurable storage limits: Administrators can set disk storage limits to 20% of total disk space for core dumps, helping prevent the folder from consuming excessive space during repeated crash scenarios.

- Maximum retained files: The storage capacity of /misc/disk1 is 140GB, and 20% of the disk is configured for core dumps, which comes to roughly 28GB. This indicates that the disk usage limit (20%) is being enforced. `.core.txt` are automatically deleted to free space, every 15 minutes.
- Syslog alerts: The system generates syslog messages to notify administrators when storage limits are reached or when automatic cleanup actions occur. This optimizes disk usage and helps prevent data loss during continuous crash scenarios.

Before Release 26.2.1, the core dump folder lacked a storage limit for core dump files. This could result in the allocated hard disk space being filled during continuous crash scenarios, potentially preventing Cisco IOS XR from saving the latest core files and leading to data loss.

From Release 26.2.1 onward, these improvements allow you to effectively control core dump folder storage and preserve disk space for critical operations, by configuring

- **Throttling mechanism bypass**: Use the **exception filepath** to redirect core files to another directory, utilizing new storage space.
- **Disk usage limit**: Use the **exception disk-usage-limit <usage limit percent>** command to set the storage limit between 20% and 50% of the total hard disk space.

Example

If the number of coreinfo files exceeds 50, Cisco IOS XR deletes older files during its periodic cleanup. These syslog messages indicate this action:

```
RP/0/RP0/CPU0:Jan 20 20:15:02.014 UTC: dumper[69243]: %OS-SYSLOG-6-LOG_INFO :
 Coreinfo count is currently 52, deleting
/misc/disk1/coredumps/sleep_11899.by.3.20250120-191744.xr-vm_node0_RP0_CPU0.34684.core.txt
RP/0/RP0/CPU0:Jan 20 20:15:02.056 UTC: dumper[69255]: %OS-SYSLOG-6-LOG_INFO :
 Coreinfo count is currently 51, deleting
/misc/disk1/coredumps/sleep_13995.by.3.20250120-193249.xr-vm_node0_RP0_CPU0.34684.core.txt
```

Requirement: Maintain core dump folder disk space and usage thresholds

This section outlines core dump folder storage requirements. Use it to maintain disk space and usage thresholds that preserve diagnostic data and system reliability.

To ensure reliable system operation and preserve critical diagnostic data, follow these requirements for managing core dump folder storage:

- Maintain at least 10% free disk space before saving host core dumps. If disk space drops below this threshold, the system stops saving core files and deletes any temporary files.
- Prevent the core dump folder from exceeding the configured disk usage limit. f the folder is over this limit and a recent core file exists within 10 minutes of a new crash, the system stops saving the new core dump and deletes any incoming files.
- Regularly monitor disk space and core dump usage to avoid unexpected data loss and ensure proper incident investigation.

Failure to observe these requirements may result in the loss of core dumps necessary for troubleshooting and analysis.

How core dump folder limit works

This section describes how core dump folder limit works on Cisco NCS 1010, including key stages, system behavior, and outcomes that support setup or management workflows.

Summary

Cisco IOS XR Software controls the core dump folder storage by monitoring disk usage and enforcing configured limits to ensure the system does not run out of disk space.

The key components that are involved in core dump folder limit are:

- **Cisco IOS XR Software**: Monitors disk usage and performs actions.
- **Core dump folder**: Stores the core files created during system crashes.

Workflow

After you enable the disk usage limit with the **exception disk-usage-limit <usage limit percent>** command, IOS XR software controls the core dump folder storage under various disk storage scenarios:

1. Cisco IOS XR software continuously monitors the disk usage of the core dump folder against the configured limit.
2. If the core dump folder storage size exceeds the configured disk usage limit, Cisco IOS XR software performs a set of actions.

When...	And...	Then Cisco IOS XR software...	And displays these syslog message...

the core dump folder storage size exceeds the configured disk usage limit value and	if the latest core file in the hard disk is more than 10 minutes older than the current crash time	deletes the older core files starting from oldest to newest until the core dump folder storage value is below the the default or configured disk usage limit value.	`RP/0/RP0/CPU0:Jan 21 10:19:36.119 UTC: dumper[67458]: %OS-SYSLOG-3-LOG_ERR : sleep_14157 signature 4f24a379e430a4a8b0967b02548`
			
			`RP/0/RSP0/CPU0:Jan 21 10:19:36.251 UTC: dumper[399]: %OS-CRHELPER-6-DELETE_STALE_CORE : Deleting /misc/disk1/coredumps/sleep_1042_by_3_20250121-100007-xr-vm_node0_RSP0_CPU0_3484.core.gz on active RP for reason: Freeing up old core in /misc/disk1/coredumps, attempting to copy new core file`

the core dump folder storage size exceeds the configured disk usage limit value and	if the latest core file in the hard disk is within 10 minutes than the current crash time	will stop saving the core files and delete any incoming core files.	

When...	And...	Then Cisco IOS XR software...	And displays these syslog message...
			RP/0/RP0/CPU0:/misc/disk1/coredumps] $RP/0/RSP0/CPU0: Jan 24 10:23:53.921 UTC: dumper[175]: %OS -COREHELPER-2-ABORT_COPY : Copy of sleep_19617.by.3. 20250124-102353.xr-vm_node 0_RSP0_CPU0.34684.core.lz4 to 0/RSP0/CPU0:/misc/disk1/ **coredumps stoped for reason: Dumper disk usage exceeds 20% threshold of /misc/disk1. Potential /misc/disk1/coredumps usage with new core: 100.0%, seconds since last core: 36.0s. Deleting core from /misc/scratch/core.**
the hard disk is filled more than 90 percent of the total hard disk space	-	will stop saving the core files and delete any incoming core files as it can't delete any old core files.	

When...	And...	Then Cisco IOS XR software...	And displays these syslog message...
			RP/0/RP0/CPU0:Mar 25 04:25:06.252 UTC: dumper[329]: %OS -COREHELPER-2-ABORT_COPY : Copy of sleep_3071.by.3.208025-04505.x-vm_node0_RP0_CPU0.3484.core.lz4 to 0/RSP0/CPU0:/misc/disk1/coredumps **stoped for reason: Not enough available space in /misc/disk1 (less than 10%). Current /misc/disk1 bytes in use: 23590813696B, size of core file: 1471276B. Deleting core from /misc/scratch/core.**

Result

Cisco IOS XR ensures that core dump storage remains within configured limits, prevents disk space exhaustion, and logs actions so operators can respond if necessary.

Configure core dump folder limit

Use this procedure to configure the core dump folder limit. It helps manage diagnostic file retention and prevent excessive core dump storage usage.

Use this procedure to limit the storage usage of the core dump folder. This prevents excessive accumulation of core dumps and ensures optimal disk space management and system reliability.

This task supports Cisco NCS 1010 setup, deployment, upgrade, or maintenance workflows.

Before you begin

Follow these steps to configure core dump folder limit.

Procedure

1. Set the core dump folder storage limit. This prevents excessive core dumps on the hard disk.

Example

```
RP/0/RP0/CPU0# configuration
RP/0/RP0/CPU0(config)# exception disk-usage-limit 29
RP/0/RP0/CPU0(config)# commit
```

You can configure the core dump folder storage limit to range to 20% of the total hard disk space.

2. Run **show exception** command to view the set core dump folder storage limit.

Example
This example displays the core dump folder storage limit set to 20 percent of the total hard disk space.

```
RP/0/RP0/CPU0# show exception
 Exception path for choice 1 is not configured or removed
 Exception path for choice 2 is not configured or removed
 Exception path for choice 3 is not configured or removed
 Default fallback/copy path = /misc/disk1/coredumps
 Core dump usage on disk limited to 20%
```

The system enforces the specified storage limit for the core dump folder, ensuring disk space is optimally managed and preventing excess core dump accumulation.