System Setup and Software Installation Guide for Cisco NCS 5500 Series Routers, IOS XR Release 7.6.x

First Published: 2022-03-09

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

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

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

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

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

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

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

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

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

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)

© 2022 Cisco Systems, Inc. All rights reserved.

CONTENTS

PREFACE **Preface** ix

PART I **Setup System and Install IOS XR Software** 11

CHAPTER 1 **New and Changed Feature Information** 1

New and Changed System Setup Features 1

CHAPTER 2 **Cisco NCS 5500 Product Overview** 3

Command Modes 3

CHAPTER 3 **Bring-up the Router** 5

Boot the Router 5

Boot the router using USB 6

Boot the Router Using iPXE 9

Setup Root User Credentials 12

Access the System Admin Console 13

Configure the Management Port 14

Perform Clock Synchronization with NTP Server 16

CHAPTER 4 **Perform Preliminary Checks** 17

Verify Software Version 17

Verify Status of Hardware Modules 18

Verify Firmware Version 19

Verify SDR Information 21

Verify Interface Status 23

CHAPTER 5 **Create User Profiles and Assign Privileges** 25

Create User Groups 26

Configure User Groups in XR VM 27

Create a User Group in System Admin VM 28

Create Users 29

Create a User Profile in XR VM 30

Create a User Profile in System Admin VM 33

Create Command Rules 34

Create Data Rules 37

Change Disaster-recovery Username and Password 39

Recover Password using PXE Boot 41

CHAPTER 6 **Perform System Upgrade and Install Feature Packages** 43

Upgrading the System 43

View supported software upgrade or downgrade versions 44

Compatibility checks for Cisco IOS XR software upgrades and downgrades 45

Show commands for software upgrade and downgrade 45

Supported software upgrade from running version 46

Supported releases to upgrade software from current version to target version 47

Supported releases from current version to an ISO version 47

Supported releases from running version to an ISO version 48

Upgrading Features 48

Install Prepared Packages 50

Install Packages 53

Uninstall Packages 58

CHAPTER 7 **Manage Automatic Dependency** 61

Update RPMs and SMUs 62

Upgrade Base Software Version 63

Downgrade an RPM 64

CHAPTER 8 **Customize Installation using Golden ISO** 67

Limitations 67

Customize Installation using Golden ISO 68
Limitations 68
Golden ISO Workflow 68
Build Golden ISO 69
Build Golden ISO Using Script 70
Install Golden ISO 77

CHAPTER 9 **Deploy Router Using Classic ZTP** 81
Build your configuration file 83
Create User Script 83
ZTP Shell Utilities 84
ZTP Helper Python Library 85
Authentication on Data Ports 90
Set Up DHCP Server 91
Customize ZTP Initialization File 94
Zero Touch Provisioning on a Fresh Boot of a Router 95
Fresh Boot Using Removable Storage Device 96
Fresh Boot Using DHCP 97
Invoke ZTP Manually 98

CHAPTER 10 **Deploy Router Using Secure ZTP** 101
On board Devices Using Three-Step Validation 102
Secure ZTP Components 102
Secure Zero Touch Provisioning 109
Secure ZTP with Removable Storage Device 110
Prepare Removable Storage Device to Provision Secure ZTP 110
How Does Secure ZTP Work with Removable Storage Device? 111
Secure ZTP with DHCP 113
Initial Set Up for Secure ZTP 113
How Does Secure ZTP Work? 114
Upgrade Image Using Secure ZTP 118
Disable Secure ZTP 121

CHAPTER 11 **Disaster Recovery** 123

Boot using USB Drive 123
Create a Bootable USB Drive Using Compressed Boot File 123
Boot the Router Using USB 124
Boot the Router Using iPXE 125
Zero Touch Provisioning 126
Setup DHCP Server 126
Invoke ZTP 128
Boot the Router Using iPXE 130
Disaster Recovery Using Manual iPXE Boot 131

PART II **Setup System and Install IOS XR7 Software** 133

CHAPTER 12 **Setup Cisco NCS 5700 Series Routers with XR7 OS** 135
Bring-up the Router 135
Boot the Router Using Manual iPXE 136
Boot the Router Using USB Drive 137
Configure the Management Port on the Router 139
Synchronize Router Clock with NTP Server 140
Perform Preliminary Checks with Router 141
Verify Software Version on Router 141
Verify Status of Hardware Modules on Router 142
Verify Interface Status on the Router 145
Verify Node Status on Router 146
Create Users and Assign Privileges on the Router 147
Create a User Profile 147
Create a User Group 148

CHAPTER 13 **Install Cisco IOS XR7 on NCS 5700 Series Routers** 151
Software Deliverables and Terminologies 151
Supported Packages 153

CHAPTER 14 **Workflow for Installing Cisco IOS XR7 Software** 155
Obtain Data Models for Install Operation 155
Create a repository to install IOS XR SMUs or optional packages 157

Create and Configure a Local Repository 157
Create and Configure an External Repository 158
Upgrade the System 160
Upgrade the current active version of Cisco IOS XR Software 160
Install optional packages for additional functionality 163

CHAPTER 15 **Additional Install Operations** 165
View the Version of Installed Packages 165
Golden ISO images 167
Build a Golden ISO image 168
Upgrade the System to Obtain Bug Fixes 169
Remove a bug fix from Cisco IOS XR software 174
Roll back software to a previously saved installation point 174
Roll back Cisco IOS XR software using a rollback ID 175
Caveats 177

PART III **System Hardware Profiles** 181

CHAPTER 16 **NCS5700 MDB Profile** 183
Hardware MDB Profiles for Layer 2 and 3 Services 183
Configure Hardware MDB Profiles for Layer 2 Services 186

Preface

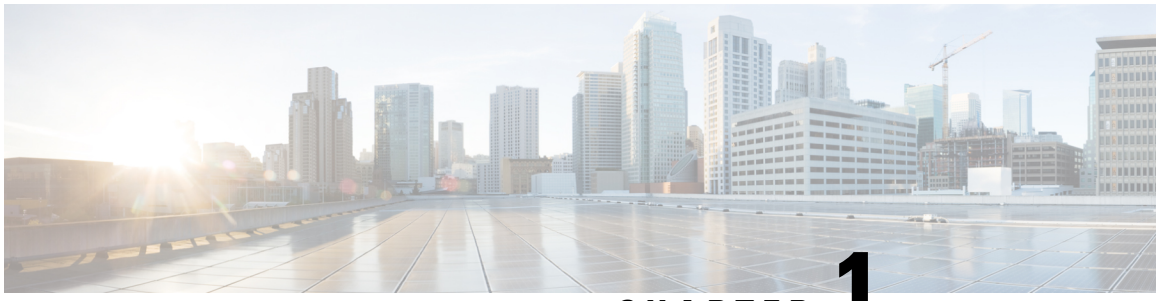
Note This release has reached end-of-life status. For more information, see the End-of-Life and End-of-Sale Notices.

Preface

PART I

Setup System and Install IOS XR Software

- New and Changed Feature Information, on page 1
- Cisco NCS 5500 Product Overview, on page 3
- Bring-up the Router, on page 5
- Perform Preliminary Checks, on page 17
- Create User Profiles and Assign Privileges, on page 25
- Perform System Upgrade and Install Feature Packages, on page 43
- Manage Automatic Dependency, on page 61
- Customize Installation using Golden ISO, on page 67
- Deploy Router Using Classic ZTP, on page 81
- Deploy Router Using Secure ZTP, on page 101
- Disaster Recovery, on page 123



CHAPTER 1

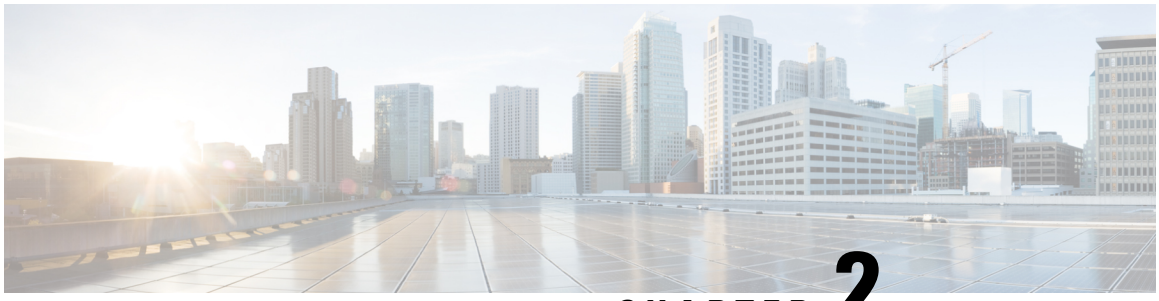
New and Changed Feature Information

This table summarizes the new and changed feature information for the *System Setup and Software Installation Guide for Cisco NCS 5500 Series Routers*.

- New and Changed System Setup Features, on page 1

New and Changed System Setup Features

Feature	Description	Changed in Release
None	No new features are introduced in this release.	Release 7.6.1



CHAPTER 2

Cisco NCS 5500 Product Overview

Cisco NCS 5500 system is a high-fault-resilient platform, which provides the next generation data-center switching environment with high bandwidth and low latency.

Cisco NCS 5500 system provides:

- A modular router with a centralized route processor with multiple line cards per chassis.
- High density, high performance, and merchant silicon-based line cards.
- IP and MPLS switching at a low cost per 100G.
- Label Switched Router (LSR) and possible Light Label switched Edge Router (LER) features and functionality with limited hardware scale and software functionality.

Note The Cisco Network Convergence System (NCS) 5700 Series builds on the Cisco NCS 5500 fixed systems by combining the forwarding ASIC design with the Cisco IOS XR7 OS. The Cisco NCS 5700 series includes the following variants:

- NCS-57B1-6D24
- NCS-57B1-5DSE

These variants of the NCS 5700 series run on the Cisco IOS XR7 operating system. For information about setting up the routers, see Setup Cisco NCS 5700 Series Routers with XR7 OS, on page 135. For information about installing the XR7 OS on NCS 5700 series, see Install Cisco IOS XR7 on NCS 5700 Series Routers, on page 151.

- Command Modes, on page 3

Command Modes

The router runs on virtualized Cisco IOS XR software. Therefore, the CLI commands must be executed on virtual machines, namely the XR LXC and the System Admin LXC.

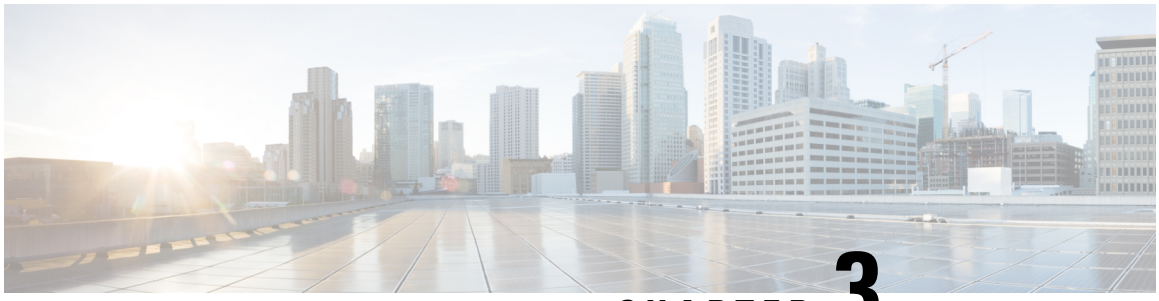
The command modes are applicable to the Cisco NCS 5500 Series Routers. This table lists the command modes for the LXCs.

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

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

System Admin EXEC mode
(System Admin LXC execution mode)
	Run commands on the System Admin LXC to display and monitor the operational state of the router hardware. The chassis or individual hardware modules can be reloaded from this mode.
	Example:
	`RP/0/RP0/CPU0:router#admin`
	`sysadmin-vm:0_RP0#`

Admin Configuration modeSystem Admin Config mode
(System Admin LXC configuration mode)
	Run configuration commands on the System Admin LXC to manage and operate the hardware modules of the entire chassis.
	Example:
	`RP/0/RP0/CPU0:router#admin`
	`sysadmin-vm:0_RP0#config`
	`sysadmin-vm:0_RP0(config)#`



CHAPTER 3

Bring-up the Router

After installing the hardware, boot the router. Connect to the XR console port and power on the router. The router completes the boot process using the pre-installed operating system (OS) image. If no image is available within the router, the router can be booted using PXE boot or an external bootable USB drive.

After booting is complete, create the root username and password, and then use it to log on to the XR console and get the router prompt. The first user created in XR console is synchronized to the System Admin console. From the XR console, access the System Admin console to configure system administration settings.

In a large-scale environment, to provision routers remotely without any manually intervention, we recommend you to use the Zero Touch Provisioning (ZTP) mechanism. ZTP offers the following implementation choices worth considering in advance:

- You can use Classic Zero Touch Provisioning, when you want to provision the devices within a secured network. See Deploy Router Using Classic ZTP, on page 81.
- You can use Secure ZTP when you must securely provision remote network devices, transverse through public internet for provisioning, or when the devices are from third-party manufacturers. See Deploy Router Using Secure ZTP, on page 101.

- Boot the Router, on page 5
- Boot the router using USB, on page 6
- Boot the Router Using iPXE, on page 9
- Setup Root User Credentials, on page 12
- Access the System Admin Console, on page 13
- Configure the Management Port, on page 14
- Perform Clock Synchronization with NTP Server, on page 16

Boot the Router

Use the console port on the Route Processor (RP) to connect to a new router. The console port connect to the XR console by default. If necessary, subsequent connections can be established through the management port, after it is configured.

Procedure

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

Step 2 Start the terminal emulation program on your workstation.

In the **COM1 Properties** window, select the **Port Settings** tab, and enter these console settings:

- For modular chassis RP, the console settings are baud rate 9600 bps, no parity, 1 stop bits and 8 data bits.
- For fixed chassis, the console settings are baud rate 115200 bps, no parity, 1 stop bits and 8 data bits.

The baud rate is set by default and cannot be changed.

Step 3 Power on the router.

Connect the power cord to the power module. Turn on the router by switching the power switch to the "ON" position. The power switch is usually located near the power module. The router boots up. The boot process details are displayed on the console screen of the terminal emulation program.

Step 4 Press **Enter**.

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

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

Note
We recommended that you check the `md5sum` of the image after copying the image from the source location to the server from where the router boots up with the new version. If you observe an `md5sum` mismatch, you can remove the corrupted file and ensure that a working copy of the image file is available for the setup to begin.

What to do next

Specify the root username and password. For more information, see Setup Root User Credentials, on page 12.

Boot the router using USB

The bootable USB drive is used to re-image the router for the purpose of system upgrade, password recovery or boot the router in case of boot failure. The USB on router is mounted as disk 2.

Before you begin

Ensure that these prerequisites are met before you boot the router using USB:

- You have access to a USB drive with a storage capacity of 8 GB to 32 GB. Both USB 2.0 and USB 3.0 are supported.
- Copy the compressed boot file, `ncs5500-usb_boot-<release_number>.zip`, from the Software Download Center to your local machine.

Procedure

Step 1 Create a bootable USB drive.

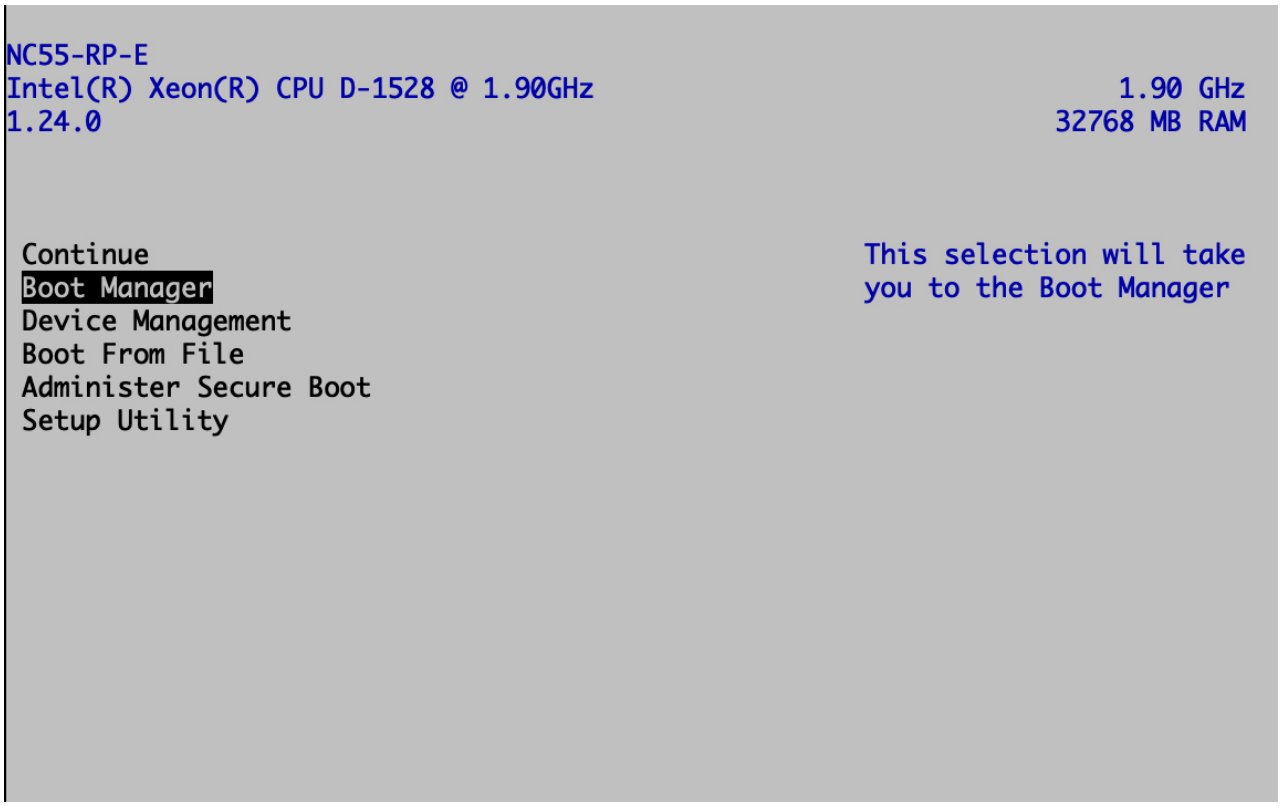
Note
The content of the zipped file ("EFI" and "boot" directories) should be extracted directly into root folder of the USB drive. If the unzipping application places the extracted files in a new folder, move the "EFI" and "boot" directories to root folder of the USB drive.

a) Connect the USB drive to your local machine and format it with the FAT32 or MS-DOS file system using the Windows Operating System or Apple MAC Disk Utility.
b) Copy the `ncs5500-usb_boot-<release_number>.zip` compressed boot file to the USB drive.
c) Verify that the copy operation is successful. To verify, compare the file size at source and destination. Additionally, verify the MD5 checksum value.
d) Extract the content of the compressed boot file by unzipping it inside the USB drive. This converts the USB drive to a bootable drive.
e) Eject the USB drive from your local machine.

Step 2 Insert the USB on the active RP, and reload or reset the power of the router.

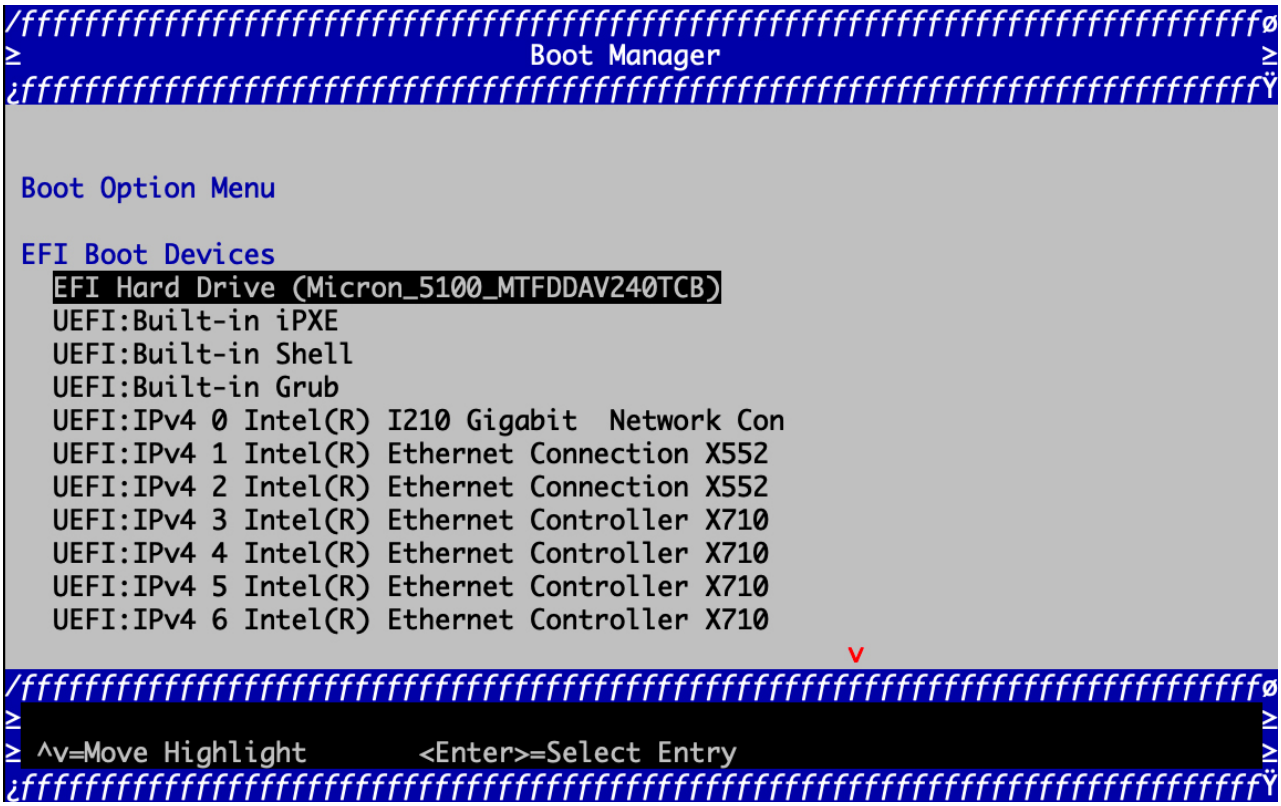
Note
Use this procedure only on active RP; the standby RP must either be removed from the chassis, or stopped at the boot menu. After the active RP is installed with images from USB, boot the standby RP.

Step 3 On active XR console, press `Delete` or `Esc` to view BIOS menu. From the BIOS menu, press the up/down arrow and select the **Boot Manager** option to view Boot menu.



If active and standby RPs are not stopped at the boot menu, the previously used boot option is used. If the system is inactive in the boot menu for 30 minutes, the system resets automatically.

Step 4 From the Boot menu, press the up/down arrow to select the required USB boot option from the menu.



Step 5 If standby RP is present and it was stopped in step 2, boot the standby RP after the active RP starts to boot. From the boot options, select `UEFI:Built-in iPXE` and proceed with the boot-up procedure.

Boot the Router Using iPXE

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

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.

Note PID and serial number is supported only if iPXE is invoked using the command (admin) hw-module location all bootmedia network reload all. If iPXE is selected manually from BIOS, PID and serial number is not supported.

iPXE boot can be performed during the following scenarios:

- migration from 32-bit to 64-bit using migration script

- recover password
- boot-up failure with 64-bit image

Before you begin

Take a backup of configuration to a TFP or FTP path to load the configuration back after the iPXE boot.

Procedure

Step 1 Login to the system admin console.

Example:

```
sysadmin-vm:0_RSP0# hw-module location all reload
Tue Mar  6  08:12:47.605 UTC
Reload hardware module ? [no,yes] yes
result Card graceful reload request on all acknowledged.
sysadmin-vm:0_RSP0#
```

Step 2 If the router is unable to boot, press Ctrl +C to stop the boot process when the following information is displayed.

Note
Use this procedure only on active RP; the standby RP must either be removed from the chassis, or stopped at the boot menu. After the active RP is installed with images from iPXE boot, boot the standby RP.

Example:

```
IOFPGA Information:
Booted from : Primary FPGA
Revison : 0x1001B
ID : 0x20171FD3
Date : 0x20191205
Fab Revision : 0x5
Base Board Presence : 0x80000015

Board is : Turin CPU Board
Booting from Primary BIOS
Booting IOS-XR (32 bit Classic XR) - Press Ctrl-c to stop
```

Step 3 Choose option 4 for iPXE boot.

Example:

```
Please select the operating system and the boot device:
     1) IOS-XR (32 bit Classic XR)
     2) IOS-XR 64 bit Boot previously installed image
     3) IOS-XR 64 bit Mgmt Network boot using DHCP server
     4) IOS-XR 64 bit Mgmt Network boot using local settings (iPXE)
     5) IOS-XR 64 bit Internal network boot from RSP/RP
     6) IOS-XR 64 bit Local boot using embedded USB media
     7) IOS-XR 64 bit Local boot using front panel USB media
Selection [1/2/3/4/5/6/7]:
```

Step 4 Manually update iPXE ROMMON details before booting using FTP or TFTP.

Note
If you are using an iPXE server, skip Step4 and proceed to Steps5 and 6.

Example:

```
iPXE>set cisco/cisco-server-url:string tftp://<path>/ncs5500-mini-x.iso
iPXE>set cisco/cisco-ipv4-address:string 1.3.24.202
iPXE>set cisco/cisco-netmask-address:str 255.255.0.0
iPXE>set cisco/cisco-gateway-address:str 1.3.0.1
```

Step 5 Open the connected management port (0/1).

Example:

```
iPXE>ifclose net0
iPXE>ifclose net1
iPXE>ifopen net1
```

where net0 and net1 represents management port0 and port1 respectively.

Step 6 Boot the required image from FTP or TFTP location.

Example:

```
iPXE> set net0/ip 5.26.8.50
iPXE> set net0/netmask 255.255.0.0
iPXE> set net0/gateway 5.26.0.1
iPXE> ifopen net0
iPXE> boot t ftp://<path>/ncs5500-mini-x-<release-number>.iso
t ftp://<path>/ncs5500-mini-x-<release-number>.iso... Operation canceled ( http://ipxe.org/0b072095)
iPXE>
iPXE> ping 5.0.0.183
64 bytes from 5.0.0.183: seq=1
64 bytes from 5.0.0.183: seq=2
64 bytes from 5.0.0.183: seq=3
Finished: Operation canceled ( http://ipxe.org/0b072095)
iPXE> boot
http://<path>/ncs5500-goldenk9-x-<release-number>-PROD_BUILD_<release-number>_DT_IMAGE__OPTIMISED.iso
http://<path>/ncs5500-goldenk9-x-<release-number>-PROD_BUILD_<release-number>_DT_IMAGE__OPTIMISED.iso...
 ok
Memory required for
image[ncs5500-goldenk9-x-<release-number>-PROD_BUILD_<release-number>_DT_IMAGE__OPTIMISED.iso]:
2345863168, available: 29061079040
Certificate parsing success
```

Step 7 After the active RP is up and running, boot the standby RP. From the boot options select `IOS-XR 64 bit Internal network boot from RSP/RP`.

Example:

```
Please select the operating system and the boot device:
     1) IOS-XR (32 bit Classic XR)
     2) IOS-XR 64 bit Boot previously installed image
     3) IOS-XR 64 bit Mgmt Network boot using DHCP server
     4) IOS-XR 64 bit Mgmt Network boot using local settings (iPXE)
     5) IOS-XR 64 bit Internal network boot from RSP/RP
     6) IOS-XR 64 bit Local boot using embedded USB media
     7) IOS-XR 64 bit Local boot using front panel USB media
Selection [1/2/3/4/5/6/7]:
```

Setup Root User Credentials

When you boot the router for the first time, the system prompts you to configure root credentials (username and password). These credentials have been set up for the root user on the XR console (root-lr), the System Admin VM (root-system), and for disaster recovery purposes.

Procedure

Step 1 **Enter root-system username:** *username*

Enter the username of the root user. The character limit is 1023. In this example, the name of the root user is "root".

Important
The specified username is mapped to the "root-lr" group on the XR console. It is also mapped as the "root-system" user on the System Admin console.

When starting the router for the first time, or after resetting the router's operating system to its default state, the router does not have any user configuration. In such cases, the router prompts you to specify the "root-system username". However, if the router has been configured previously, the router prompts you to enter the "username", as described in Step 4.

Step 2 **Enter secret:** *password*

Enter the password for the root user. The character range of the password is from 6 through 253 characters. The password that you type is not displayed on the CLI for security reasons.

The root-system username and password must be safeguarded as they have superuser privileges. They are used to access the complete router configuration.

Step 3 **Enter secret again:** *password*

Reenter the password for the root-system user. The password that you type is not displayed on the CLI for security reasons.

Step 4 **Username:** *username*

Enter the root-system username to login to the XR VM console.

Step 5 **Password:** *password*

Enter the password of the root-system user. The correct password displays the router prompt. You are now logged into the XR VM console.

Step 6 (Optional) **show run username**

Displays user details.

```
username root
 group root-lr
 group cisco-support
 secret 5 $1$NBg7$fHs1inKPZVvzqxMv775UE/
!
```

What to do next

- Configure routing functions from the XR console.
- Configure system administration settings from the System Admin prompt. The System Admin prompt is displayed on accessing the System Admin console. For details on how to get the System Admin prompt, see Access the System Admin Console, on page 13.

Access the System Admin Console

You must log in to the System Admin console through the XR console to perform all system administration and hardware management setup.

Procedure

Step 1 Log in to the XR console as the root user.

Step 2 (Optional) Disable the login banner on console port when accessing the System Admin mode from XR mode.

a) **configure**

b) **service sysadmin-login-banner disable**

Example:

```
RP/0/RP0/CPU0:router(config)#service sysadmin-login-banner disable
```

Disable the login banner on console port in System Admin mode.

c) **commit**

d) **end**

Step 3 **admin**

Example:

The login banner is enabled by default. The following example shows the command output with the login banner enabled:

```
RP/0/RP0/CPU0:router#admin

Mon May 22 06:57:29.350 UTC

root connected from 127.0.0.1 using console on host
sysadmin-vm:0_RP0# exit
Mon May  22 06:57:32.360 UTC
```

The following example shows the command output with the login banner disabled:

```
RP/0/RP0/CPU0:router#admin
Thu Mar 01:07:14.509 UTC
sysadmin-vm:0_RP0# exit
```

Step 4 (Optional) **exit**

Return to the XR mode from the System Admin mode.

Configure the Management Port

To use the Management port for system management and remote communication, you must configure an IP address and a subnet mask for the management ethernet interface. To communicate with devices on other networks (such as remote management stations or TFTP servers), you need to configure a default (static) route for the router.

Before you begin

- Consult your network administrator or system planner to procure IP addresses and a subnet mask for the management interface.
- Physical port Ethernet 0 and Ethernet 1 on RP are the management ports. Ensure that the port is connected to management network.

SUMMARY STEPS

1. **configure**
2. **interface MgmtEth** *rack/slot/port*
3. (Optional) **vrf** *vrf-id*
4. **ipv4 address** *ipv4-address subnet-mask*
5. **ipv4 address** *ipv4 virtual address subnet-mask*
6. **no shutdown**
7. **exit**
8. **router static address-family ipv4 unicast** *0.0.0.0/0 default-gateway*
9. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

Step 1 **configure**

Example:

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

Enters global configuration mode.

Step 2 **interface MgmtEth** *rack/slot/port*

Example:

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

Enters interface configuration mode for the management interface of the primary RP.

Step 3 (Optional) **vrf** *vrf-id*

Example:

```
RP/0/RP0/CPU0:router(config-sg-tacacs+)# vrf vrf-id
```

Specifies the Virtual Private Network (VPN) routing and forwarding (VRF) reference.

Step 4 **ipv4 address** *ipv4-address subnet-mask*

Example:

```
RP/0/RP0/CPU0:router(config-if)#ipv4 address 10.1.1.1/8
```

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

Step 5 **ipv4 address** *ipv4 virtual address subnet-mask*

Example:

```
RP/0/RP0/CPU0:router(config-if)#ipv4 address 1.70.31.160 255.255.0.0
```

Assigns a virtual IP address and a subnet mask to the interface.

Step 6 **no shutdown**

Example:

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

Places the interface in an "up" state.

Step 7 **exit**

Example:

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

Exits the Management interface configuration mode.

Step 8 **router static address-family ipv4 unicast** *0.0.0.0/0 default-gateway*

Example:

```
RP/0/RP0/CPU0:router(config)#router static address-family ipv4 unicast 0.0.0.0/0 12.25.0.1
```

Specifies the IP address of the default-gateway to configure a static route; this is to be used for communications with devices on other networks.

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

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

end —Prompts user to take one of these actions:

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

What to do next

Connect to the management port to the ethernet network. With a terminal emulation program, establish a SSH or telnet connection to the management interface port using its IP address. Before establishing a telnet session, use the **telnet ipv4|ipv6 server max-servers** command in the XR Config mode, to set number of allowable telnet sessions to the router.

Perform Clock Synchronization with NTP Server

There are independent system clocks for the XR console and the System Admin console. To ensure that these clocks do not deviate from true time, they need to be synchronized with the clock of a NTP server. In this task you will configure a NTP server for the XR console. After the XR console clock is synchronized, the System Admin console clock will automatically synchronize with the XR console clock.

Before you begin

Configure and connect to the management port.

Procedure

Step 1 **configure**

Example:

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

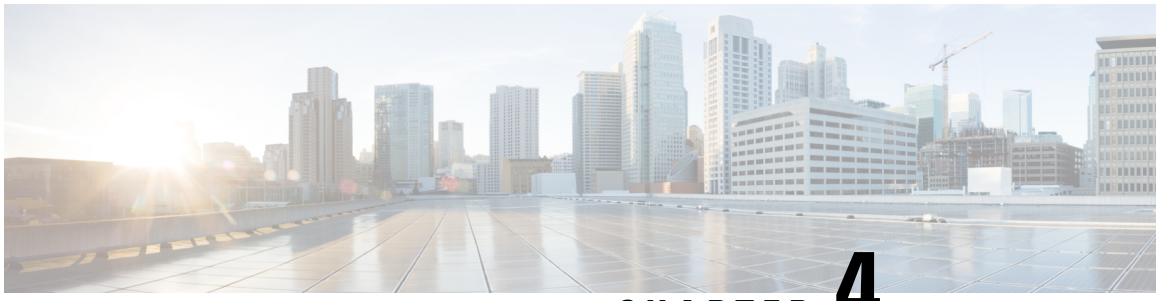
Enters global configuration mode.

Step 2 **ntp server** *server_address*

Example:

```
RP/0/RP0/CPU0:router(config)#ntp server 64.90.182.55
```

The XR console clock is configured to be synchronized with the specified sever.



CHAPTER 4

Perform Preliminary Checks

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

- Verify Software Version, on page 17
- Verify Status of Hardware Modules, on page 18
- Verify Firmware Version, on page 19
- Verify SDR Information, on page 21
- Verify Interface Status, on page 23

Verify Software Version

The router is shipped with the Cisco IOS XR software pre-installed. Verify that the latest version of the software is installed. If a newer version is available, perform a system upgrade. This will install the newer version of the software and provide the latest feature set on the router.

Perform this task to verify the version of Cisco IOS XR software running on the router.

SUMMARY STEPS

1. **show version**

DETAILED STEPS

Procedure

show version

Example:

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

Displays the version of the various software components installed on the router. The result includes the version of Cisco IOS XR software and its various components.

Example

```
Cisco IOS XR Software, Version <release-version>
Copyright (c) 2013-2015 by Cisco Systems, Inc.

Build Information:
Built By : <user>
Built On : <date and time stamp>
Build Host :
Version : <release-version>
Location : /opt/cisco/XR/packages/

cisco NCS-5500 () processor
System uptime is 3 hours, 42 minutes
```

What to do next

Verify the result to ascertain whether a system upgrade or additional package installation is required. If that is required, refer to the tasks in the chapter Perform System Upgrade and Install Feature Packages.

Verify Status of Hardware Modules

Hardware modules include RPs, LCs, fan trays, and so on. On the router, multiple hardware modules are installed. Perform this task to verify that all hardware modules are installed correctly and are operational.

Before you begin

Ensure that all required hardware modules have been installed on the router.

SUMMARY STEPS

1. **admin**
2. **show platform**

DETAILED STEPS

Procedure

Step 1 **admin**

Example:

```
RP/0/RP0/CPU0:router# admin
```

Enters System Admin EXEC mode.

Step 2 **show platform**

Example:

```
sysadmin-vm:0_RP0#show platform
```

Displays the list of hardware modules detected on the router.

```
Location  Card Type                HW State      SW State      Config State
------------------------------------------------------------------------------

0/0       NC55-36X100G             OPERATIONAL   OPERATIONAL   NSHUT
0/1       NC55-36X100G             OPERATIONAL   OPERATIONAL   NSHUT
0/2       NC55-36X100G             OPERATIONAL   OPERATIONAL   NSHUT
0/3       NC55-36X100G             OPERATIONAL   OPERATIONAL   NSHUT
0/4       NC55-36X100G             OPERATIONAL   OPERATIONAL   NSHUT
0/5       NC55-36X100G             OPERATIONAL   OPERATIONAL   NSHUT
0/6       NC55-36X100G             OPERATIONAL   OPERATIONAL   NSHUT
0/7       NC55-36X100G             OPERATIONAL   OPERATIONAL   NSHUT
0/RP0     NC55-RP                  OPERATIONAL   OPERATIONAL   NSHUT
0/RP1     NC55-RP                  OPERATIONAL   OPERATIONAL   NSHUT
0/FC0     NC55-5508-FC             OPERATIONAL   OPERATIONAL   NSHUT
0/FC1     NC55-5508-FC             OPERATIONAL   OPERATIONAL   NSHUT
0/FC2     NC55-5508-FC             OPERATIONAL   OPERATIONAL   NSHUT
0/FC3     NC55-5508-FC             OPERATIONAL   OPERATIONAL   NSHUT
0/FC4     NC55-5508-FC             OPERATIONAL   OPERATIONAL   NSHUT
0/FC5     NC55-5508-FC             OPERATIONAL   OPERATIONAL   NSHUT
0/FT0     NC55-5508-FAN            OPERATIONAL   N/A           NSHUT
0/FT1     NC55-5508-FAN            OPERATIONAL   N/A           NSHUT
0/FT2     NC55-5508-FAN            OPERATIONAL   N/A           NSHUT
0/SC0     NC55-SC                  OPERATIONAL   OPERATIONAL   NSHUT
0/SC1     NC55-SC                  OPERATIONAL   OPERATIONAL   NSHUT
```

From the result, verify that all the hardware modules installed on the chassis are listed. If a module is not listed, it indicates either that module is malfunctioning, or it is not properly installed. Remove and reinstall the hardware module.

Verify Firmware Version

The firmware on various hardware components of the router must be compatible with the Cisco IOS XR image installed. Incompatibility might cause the router to malfunction. Complete this task to verify the firmware version.

SUMMARY STEPS

1. **show hw-module fpd**

DETAILED STEPS

Procedure

show hw-module fpd

Example:

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

                                                                  FPD Versions
                                                                 ==============
Location  Card type         HWver FPD device       ATR Status   Run    Programd
------------------------------------------------------------------------------
0/0       NC55-36X100G      0.108 Bootloader           CURRENT    1.15    1.15
```

```
0/0       NC55-36X100G      0.108 IOFPGA              CURRENT    0.08    0.08
0/1       NC55-36X100G      0.203 Bootloader          CURRENT    1.15    1.15
0/1       NC55-36X100G      0.203 IOFPGA              CURRENT    0.08    0.08
0/2       NC55-36X100G      0.203 Bootloader          CURRENT    1.15    1.15
0/2       NC55-36X100G      0.203 IOFPGA              CURRENT    0.08    0.08
0/3       NC55-36X100G      0.203 Bootloader          CURRENT    1.15    1.15
0/3       NC55-36X100G      0.203 IOFPGA              CURRENT    0.08    0.08
0/4       NC55-36X100G      0.203 Bootloader          CURRENT    1.15    1.15
0/4       NC55-36X100G      0.203 IOFPGA              CURRENT    0.08    0.08
0/5       NC55-36X100G      0.203 Bootloader          CURRENT    1.15    1.15
0/5       NC55-36X100G      0.203 IOFPGA              CURRENT    0.08    0.08
0/6       NC55-36X100G      0.203 Bootloader          CURRENT    1.15    1.15
0/6       NC55-36X100G      0.203 IOFPGA              CURRENT    0.08    0.08
0/7       NC55-36X100G      0.203 Bootloader          CURRENT    1.15    1.15
0/7       NC55-36X100G      0.203 IOFPGA              CURRENT    0.08    0.08
0/RP0     NC55-RP           1.1   Bootloader          CURRENT    9.19    9.19
0/RP0     NC55-RP           1.1   IOFPGA              CURRENT    0.06    0.06
0/RP1     NC55-RP           1.1   Bootloader          CURRENT    9.19    9.19
0/RP1     NC55-RP           1.1   IOFPGA              CURRENT    0.06    0.06
0/FC0     NC55-5508-FC      0.109 Bootloader          CURRENT    1.64    1.64
0/FC0     NC55-5508-FC      0.109 IOFPGA              CURRENT    0.11    0.11
0/FC1     NC55-5508-FC      0.109 Bootloader          CURRENT    1.64    1.64
0/FC1     NC55-5508-FC      0.109 IOFPGA              CURRENT    0.11    0.11
0/FC2     NC55-5508-FC      0.109 Bootloader          CURRENT    1.64    1.64
0/FC2     NC55-5508-FC      0.109 IOFPGA              CURRENT    0.11    0.11
0/FC3     NC55-5508-FC      0.109 Bootloader          CURRENT    1.64    1.64
0/FC3     NC55-5508-FC      0.109 IOFPGA              CURRENT    0.11    0.11
0/FC4     NC55-5508-FC      0.109 Bootloader          CURRENT    1.64    1.64
0/FC4     NC55-5508-FC      0.109 IOFPGA              CURRENT    0.11    0.11
0/FC5     NC55-5508-FC      0.109 Bootloader          CURRENT    1.64    1.64
0/FC5     NC55-5508-FC      0.109 IOFPGA              CURRENT    0.11    0.11
0/SC0     NC55-SC           1.4   Bootloader          CURRENT    1.64    1.64
0/SC0     NC55-SC           1.4   IOFPGA              CURRENT    0.06    0.06
0/SC1     NC55-SC           1.4   Bootloader          CURRENT    1.64    1.64
0/SC1     NC55-SC           1.4   IOFPGA              CURRENT    0.06    0.06
```

Displays the list of hardware modules detected on the router.

Note
This command can be run from both XR VM and System Admin VM modes.

In the above output, some of the significant fields are:

- FPD Device- Name of the hardware component such as FPD, CFP, and so on.
- ATR-Attribute of the hardware component. Some of the attributes are:
  - B- Backup Image
  - S-Secure Image
  - P-Protected Image
- Status- Upgrade status of the firmware. The different states are:
  - 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 newer firmware version is available in the installed image. It is recommended that an upgrade be performed.

- RLOAD REQ-The upgrade has been completed, 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 corrupted. Reinstall the firmware.
- UPGD SKIP-The upgrade has been skipped because the installed firmware version is higher than the one available in the image.

- Running- Current version of the firmware running on the FPD.

What to do next

If it is required to replace a line card or route processor, use one of the two methods:

- Manual FPD upgrade:
  1. Insert the new line card or route processor.
  2. If `auto fpd upgrade` option is enabled in running configuration, use the **show hw-module fpd** command to check the status of the FPDs that are not activated. If the status is `RELOAD_REQ`, reload the line card or route processor.
  3. If `auto fpd upgrade` option is not enabled, use the **show hw-module fpd** command to check the FPDs that need to be upgraded. It is recommended to upgrade all the FPDs at once.
  4. Use manual FPD upgrade to upgrade all FPDs for line cards and route processors. Reload the line cards or route processors once the FPD upgrade is successful.

- Automatic FPD upgrade:
  1. If automatic FPD upgrade is not configured, use **fpd auto-upgrade enable** command to configure.
  2. Insert the line card or route processor.
  3. After the line card or route processor comes up, use the **show hw-module fpd** command to check the status of the FPDs that are not activated. If the status is `RELOAD_REQ`, reload the line card or route processor.
  4. Verify that all the other FPDs in the same node are either in `CURRENT` or `RELOAD_REQ` state before starting a manual reload of the router.

Verify SDR Information

Secure domain routers (SDRs) divide a single physical system into multiple logically-separated routers. SDRs are also known as logical routers (LRs). On the router, only one SDR is supported. This SDR is termed the default-sdr. Every router is shipped with the default-sdr, which owns all RPs installed in the routing system. An instance of this SDR runs on line cards and route processors. Complete this task to verify the details of the SDR instances.

Procedure

Step 1 **admin**

Example:

```
RP/0/RP0/CPU0:router# admin
```

Enters mode.

Step 2 **show sdr**

Example:

```
sysadmin-vm:0_RP0# show sdr
```

Displays the SDR information for every node.

```
sysadmin-vm:0_RP0# show sdr

sdr default-sdr
 location 0/0/VM1
  sdr-id             2
  IP Address of VM   192.0.4.3
  MAC address of VM  A4:6C:2A:2B:AA:A6
  VM State           RUNNING
  start-time         2015-12-03T15:38:38.74514+00:00
  Last Reload Reason "SMU:Reboot triggered by install"
  Reboot Count       2
 location 0/1/VM1
  sdr-id             2
  IP Address of VM   192.0.8.3
  MAC address of VM  B0:AA:77:E7:5E:DA
  VM State           RUNNING
  start-time         2015-12-03T15:38:39.730036+00:00
  Last Reload Reason "SMU:Reboot triggered by install"
  Reboot Count       2
 location 0/2/VM1
  sdr-id             2
  IP Address of VM   192.0.12.3
  MAC address of VM  B0:AA:77:E7:67:34
  VM State           RUNNING
  start-time         2015-12-03T15:38:38.886947+00:00
  Last Reload Reason "SMU:Reboot triggered by install"
  Reboot Count       2
 location 0/3/VM1
  sdr-id             2
  IP Address of VM   192.0.16.3
  MAC address of VM  B0:AA:77:E7:58:86
  VM State           RUNNING
  start-time         2015-12-03T15:38:40.391205+00:00
  Last Reload Reason "SMU:Reboot triggered by install"
  Reboot Count       2
 location 0/4/VM1
  sdr-id             2
  IP Address of VM   192.0.20.3
  MAC address of VM  B0:AA:77:E7:46:C2
  VM State           RUNNING
  start-time         2015-12-03T15:38:39.84469+00:00
  Last Reload Reason "SMU:Reboot triggered by install"
  Reboot Count       2
```

```
location 0/5/VM1
 sdr-id             2
 IP Address of VM   192.0.24.3
 MAC address of VM  B0:AA:77:E7:84:40
 VM State           RUNNING
 start-time         2015-12-04T03:48:24.017443+00:00
 Last Reload Reason "VM_REQUESTED_UNGRACEFUL_RELOAD:Headless SDR"
 Reboot Count       3
location 0/6/VM1
 sdr-id             2
 IP Address of VM   192.0.28.3
 MAC address of VM  B0:AA:77:E7:55:FE
 VM State           RUNNING
 start-time         2015-12-03T15:38:38.74753+00:00
 Last Reload Reason "SMU:Reboot triggered by install"
 Reboot Count       2
location 0/7/VM1
 sdr-id             2
 IP Address of VM   192.0.32.3
 MAC address of VM  B0:AA:77:E7:60:C6
 VM State           RUNNING
 start-time         2015-12-03T15:38:38.691481+00:00
 Last Reload Reason "SMU:Reboot triggered by install"
 Reboot Count       2
location 0/RP0/VM1
 sdr-id             2
 IP Address of VM   192.0.108.4
 MAC address of VM  10:05:CA:D7:FE:6F
 VM State           RUNNING
 start-time         2015-12-04T07:03:04.549294+00:00
 Last Reload Reason CARD_SHUTDOWN
 Reboot Count       1
location 0/RP1/VM1
 sdr-id             2
 IP Address of VM   192.0.112.4
 MAC address of VM  10:05:CA:D8:3F:43
 VM State           RUNNING
 start-time         2015-12-04T09:21:42.083046+00:00
 Last Reload Reason CARD_SHUTDOWN
 Reboot Count       1
```

For a functional SDR, the VM State is "RUNNING". If the SDR is not running on a node, no output is shown in the result, for that location.

What to do next

If you find SDR is not running on a node, try reloading the node. To do that, use the **hw-module location** *node-id* **reload** command in the System Admin EXEC mode.

Verify Interface Status

After the router has booted, all available interfaces must be discovered by the system. If interfaces are not discovered, it might indicate a malfunction in the unit. Complete this task to view the number of discovered interfaces.

SUMMARY STEPS

1. **show ipv4 interface summary**

DETAILED STEPS

Procedure

show ipv4 interface summary

Example:

```
RP/0/RP0/CPU0:router#show ipv4 interface summary
```

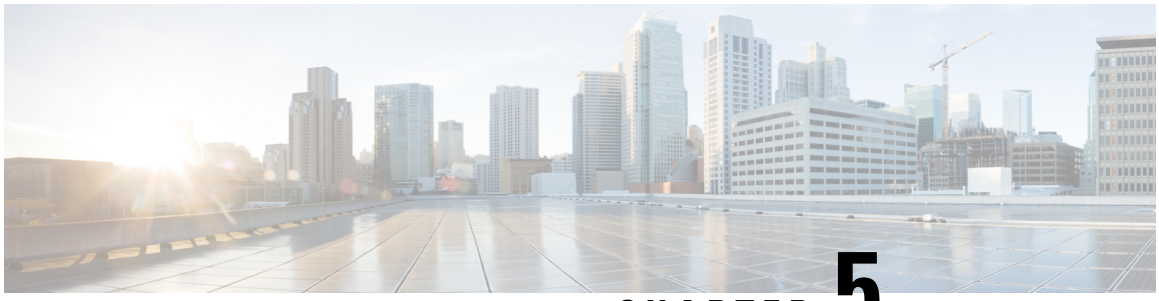
When a router is turned on for the first time, all interfaces are in the 'unassigned' state. Verify that the total number of interfaces displayed in the result matches with the actual number of interfaces present on the router.

```
IP address      State   State           State           State
config          up,up   up,down         down,down       shutdown,down
--------------------------------------------------------------------------
Assigned        0       0               0               0
Unnumbered      0       0               0               0
Unassigned      0       0               0               4
```

In the above result:

- Assigned— An IP address is assigned to the interface.
- Unnumbered— Interface which has borrowed an IP address already configured on one of the other interfaces of the router.
- Unassigned—No IP address is assigned to the interface.

You can also use the **show interfaces brief** and **show interfaces summary** commands in the XR EXEC mode to verify the interface status.



CHAPTER 5

Create User Profiles and Assign Privileges

To provide controlled access to the XR and System Admin configurations on the router, user profiles are created with assigned privileges. The privileges are specified using command rules and data rules.

The authentication, authorization, and accounting (aaa) commands are used for the creation of users, groups, command rules, and data rules. The `aaa` commands are also used for changing the disaster-recovery password.

Note You cannot configure the external AAA server and services from the System Admin VM. It can be configured only from the XR VM.

Configure AAA authorization to restrict users from uncontrolled access. If AAA authorization is not configured, the command and data rules associated to the groups that are assigned to the user are bypassed. An IOS-XR user can have full read-write access to the IOS-XR configuration through Network Configuration Protocol (NETCONF), google-defined Remote Procedure Calls (gRPC) or any YANG-based agents. In order to avoid granting uncontrolled access, enable AAA authorization before setting up any configuration.

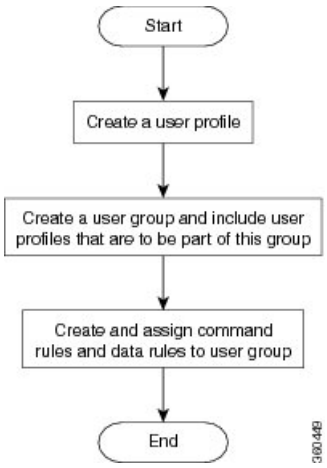
Note If any user on XR is deleted, the local database checks whether there is a first user on System Admin VM.

- If there is a first user, no syncing occurs.
- If there is no first user, then the first user on XR (based on the order of creation) is synced to System Admin VM.
- When a user is added in XR, if there is no user on System Admin mode, then the user is synced to sysadmin-vm. After the synchronization, any changes to the user on XR VM does not synchronize on the System Admin VM.
- A user added on the System Admin VM does not synchronize with XR VM.
- Only the first user or disaster-recovery user created on System Admin VM synchronizes with the host VM.
- Changes to credentials of first user or disaster-recovery user on System Admin VM synchronizes with the host VM.
- The first user or disaster-recovery user deleted on System Admin VM does not synchronize with the host VM. The host VM retains the user.

Users are authenticated using username and password. Authenticated users are entitled to execute commands and access data elements based on the command rules and data rules that are created and applied to user groups. All users who are part of a user group have such access privileges to the system as defined in the command rules and data rules for that user group.

The workflow for creating user profile is represented in this flow chart:

Figure 1: Workflow for Creating User Profiles



Note The root-lr user, created for the XR VM during initial router start-up, is mapped to the root-system user for the System Admin VM. The root-system user has superuser permissions for the System Admin VM and therefore has no access restrictions.

Use the **show run aaa** command in the Config mode to view existing aaa configurations.

The topics covered in this chapter are:

- Create User Groups, on page 26
- Create Users , on page 29
- Create Command Rules, on page 34
- Create Data Rules, on page 37
- Change Disaster-recovery Username and Password, on page 39
- Recover Password using PXE Boot, on page 41

Create User Groups

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.

For extensive information about creating user groups, task groups, RADIUS and TACACS configurations, see the *Configuring AAA Services* chapter in the *System Security Configuration Guide for Cisco NCS 5500 Series Routers*. For detailed information about commands, syntax and their description, see the *Authentication, Authorization, and Accounting Commands* chapter in the *System Security Command Reference for Cisco NCS 5500 Series Routers and Cisco NCS 540 and NCS 560 Series Routers*.

Configure User Groups in XR VM

User groups are configured with the command parameters for a set of users, such as task groups. Entering the **usergroup** command accesses the user group configuration submode. Users can remove specific user groups by using the **no** form of the **usergroup** command. Deleting a usergroup that is still referenced in the system results in a warning.

Before you begin

Note Only users associated with the WRITE:AAA task ID can configure user groups. User groups cannot inherit properties from predefined groups, such as owner-sdr.

Procedure

Step 1 **configure**

Example:

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

Enters global configuration mode.

Step 2 **usergroup** *usergroup-name*

Example:

```
 RP/0/RP0/CPU0:router(config)# usergroup beta
```

Creates a name for a particular user group and enters user group configuration submode.

- Specific user groups can be removed from the system by specifying the **no** form of the **usergroup** command.

Step 3 **description** *string*

Example:

```
 RP/0/RP0/CPU0:router(config-ug)#
description this is a sample user group description
```

(Optional) Creates a description of the user group named in Step 2.

Step 4 **inherit usergroup** *usergroup-name*

Example:

```
 RP/0/RP0/CPU0:router(config-ug)#
inherit usergroup sales
```

- Explicitly defines permissions for the user group.

Step 5 **taskgroup** *taskgroup-name*

Example:

```
 RP/0/RP0/CPU0:router(config-ug)# taskgroup beta
```

Associates the user group named in Step 2 with the task group named in this step.

- The user group takes on the configuration attributes (task ID list and permissions) already defined for the entered task group.

Step 6 Repeat Step for each task group to be associated with the user group named in Step 2.

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

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

end —Prompts user to take one of these actions:

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

Create a User Group in System Admin VM

Create a user group for the System Admin VM.

The router supports a maximum of 32 user groups.

Before you begin

Create a user profile. See the *Create User* section.

Procedure

Step 1 **admin**

Example:

```
RP/0/RP0/CPU0:router# admin
```

Enters mode.

Step 2 **config**

Example:

```
sysadmin-vm:0_RP0#config
```

Enters System Admin Config mode.

Step 3 **aaa authentication groups group** *group_name*

Example:

```
sysadmin-vm:0_RP0(config)#aaa authentication groups group gr1
```

Creates a new user group (if it is not already present) and enters the group configuration mode. In this example, the user group "gr1" is created.

Note
By default, the user group "root-system" is created by the system at the time of root user creation. The root user is part of this user group. Users added to this group will get root user permissions.

Step 4 **users** *user_name*

Example:

```
sysadmin-vm:0_RP0(config-group-gr1)#users us1
```

Specify the name of the user that should be part of the user group.

You can specify multiple user names enclosed withing double quotes. For example, **users** "*user1 user2 ...*".

Step 5 **gid** *group_id_value*

Example:

```
sysadmin-vm:0_RP0(config-group-gr1)#gid 50
```

Specify a numeric value. You can enter any 32 bit integer.

Step 6 Use the **commit** or **end** command.

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

end —Prompts user to take one of these actions:

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

What to do next

- Create command rules.
- Create data rules.

Create Users

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

Note Users created in the System Admin VM are different from the ones created in XR VM. As a result, the username and password of a System Admin VM user cannot be used to access the XR VM, and vice versa.

XR VM and System Admin VM User Profile Synchronization

Initial User Profile Synchronization: When a user profile is created for the first time within the XR VM, the username and password are synchronized with the System Admin VM, but only if the user does not already

exist in the System Admin VM. This initial synchronization ensures consistent user information between the two VMs.

Limitations on Subsequent Changes: However, it is important to note that the System Admin VM does not synchronize subsequent password changes or user deletions made within the XR VM. Consequently, the passwords in the XR VM and the System Admin VM may differ, and user profiles may not be updated in real time to reflect deletions within the XR VM.

User Deleting Handling: Additionally, when a user is deleted within the XR VM, the corresponding user profile in the System Admin VM remains unaffected. In other words, user deletion in the XR VM does not automatically remove the user's profile in the System Admin VM.

For extensive information about creating user groups, task groups, RADIUS and TACACS configurations, see the *Configuring AAA Services* chapter in the *System Security Configuration Guide for Cisco NCS 5500 Series Routers*. For detailed information about commands, syntax and their description, see the *Authentication, Authorization, and Accounting Commands* chapter in the *System Security Command Reference for Cisco NCS 5500 Series Routers and Cisco NCS 540 and NCS 560 Series Routers*.

Create a User Profile in XR VM

Table 1: Feature History Table

Feature name	Release Information	Feature Description
Enhanced Login Banner Standards	Release 7.3.1	To comply with the US DoD, an option to enable display of login banner is introduced. The login banner provides information such as number of successful and unsuccessful login attempts, time stamp, login method, and so on. The login-history command is introduced.

Each user is identified by a username that is unique across the administrative domain. Each user must be a member of at least one user group. Deleting a user group may orphan the users associated with that group. The AAA server authenticates orphaned users but most commands are not authorized.

For more information about AAA, and creating users, see the *Configuring AAA Services* chapter in the *System Security Configuration Guide for Cisco NCS 5500 Series Routers*. For detailed information about related commands, syntax and their description, see the *Authentication, Authorization, and Accounting Commands* chapter in the *System Security Command Reference for Cisco NCS 5500 Series Routers and Cisco NCS 540 and NCS 560 Series Routers*.

Procedure

Step 1 **configure**

Example:

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

Enters global configuration mode.

Step 2 **username** *user-name*

Example:

```
RP/0/RP0/CPU0:router(config)# username user1
```

Creates a name for a new user (or identifies a current user) and enters username configuration submode.

- The *user-name* argument can be only one word. Spaces and quotation marks are not allowed.

Step 3 Do one of the following:

- **password** {**0** | **7**} *password*
- **secret** {**0** | **5** | **8** | **9** | **10**} *secret*

Example:

```
Router(config-un)# password 0 pwd1
```

or

```
Router(config-un)# secret 0 sec1
```

Specifies a password for the user named in Step 2.

- Use the **secret** command to create a secure login password for the user names specified in Step 2.
- Entering **0** following the **password** command specifies that an unencrypted (clear-text) password follows. Entering **7** following the **password** command specifies that an encrypted password follows.
- For the **secret** command, the following values can be entered:
  - **0** : specifies that a secure unencrypted (clear-text) password follows
  - **5** : specifies that a secure encrypted password follows that uses MD5 hashing algorithm
  - **8** : specifies that Type 8 secret that uses SHA256 hashing algorithm follows
  - **9** : specifies that Type 9 secret that uses SCrypt hashing algorithm follows

    Note
    The Type 8 and Type 9 secrets are supported on the IOS XR 64-bit operating system starting from Cisco IOS XR Software Release 7.0.1. Prior to this release, it was supported only on the IOS XR 32-bit operating system.

  - **10** : specifies Type 10 secret that uses SHA512 hashing algorithm

    Note
    - Type 10 secret is supported only for Cisco IOS XR 64 bit platform.
    - Backward compatibility issues such as configuration loss, authentication failure, and so on, are expected when you downgrade to lower versions that still use **MD5** or **SHA256** encryption algorithms. If there are any type 10 secrets, convert the **secrets** to type 5 if you are downgrading the system from versions 7.0.1 and above to versions 6.5.3 and above. If you are downgrading the system from versions 7.0.1 and above to versions below 6.5.3, then un-configure all users from the XR-vm and sysadmin-vm before executing install activate.
    - In a first user configuration scenario or when you reconfigure a user, the system synchronises only the Type 5 and Type 10 secrets from XR VM to System Admin VM and Host VM. It does not synchronize the Type 8 and Type 9 secrets in such scenarios.

- Type **0** is the default for the **password** and **secret** commands.
- From Cisco IOS XR Software Release 7.0.1 and later, the default hashing type is 10 (SHA512) when clear text secret is configured without choosing the type in the configuration.

Step 4 **group** *group-name*

Example:

```
RP/0/RP0/CPU0:router(config-un)# group sysadmin
```

Assigns the user named in Step 2 to a user group that has already been defined through the **usergroup** command.

- The user takes on all attributes of the user group, as defined by that user group's association to various task groups.
- Each user must be assigned to at least one user group. A user may belong to multiple user groups.

Step 5 Repeat step 4 for each user group to be associated with the user specified in step 2.

Step 6 (Optional) You can enable the display of the US Department of Defense DOD-approved login banner. The banner is displayed before granting access to devices. The banner also ensures privacy and security that is consistent with applicable federal laws. In addition, the system keeps track of logins, right from the system boot, or as soon as the user profile is created.

Note
When you reload a router, login notifications get reset.

Enable or disable the login banner using these commands:

Example:

```
Router(config-un)#login-history enable
Router(config-un)#login-history disable
```

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

```
Router(config-un)# show running-config username NAME1
Fri Jan 29 13:55:28.261 UTC
username NAME1
 group UG1
 secret * **********
 password * ******
 login-history enable
```

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

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

end —Prompts user to take one of these actions:

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

Create a User Profile in System Admin VM

Create new users for the System Admin VM. Users are included in a user group and assigned certain privileges. The users have restricted access to the commands and configurations in the System Admin VM console, based on assigned privileges.

The router supports a maximum of 1024 user profiles.

The root-lr user of XR VM can access the System Admin VM by entering **Admin** command in the XR EXEC mode. The router does not prompt you to enter any username and password. The XR VM root-lr user is provided full access to the System Admin VM.

Procedure

Step 1 **admin**

Example:

```
RP/0/RP0/CPU0:router# admin
```

Enters mode.

Step 2 **config**

Example:

```
sysadmin-vm:0_RP0#config
```

Enters System Admin Config mode.

Step 3 **aaa authentication users user** *user_name*

Example:

```
sysadmin-vm:0_RP0(config)#aaa authentication users user us1
```

Creates a new user and enters user configuration mode. In the example, the user "us1" is created.

Step 4 **password** *password*

Example:

```
sysadmin-vm:0_RP0(config-user-us1)#password pwd1
```

Enter the password that will be used for user authentication at the time of login into System Admin VM.

Step 5 **uid** *user_id_value*

Example:

```
sysadmin-vm:0_RP0(config-user-us1)#uid 100
```

Specify a numeric value. You can enter any 32 bit integer.

Step 6 **gid** *group_id_value*

Example:

```
sysadmin-vm:0_RP0(config-user-us1)#gid 50
```

Specify a numeric value. You can enter any 32 bit integer.

Step 7 **ssh_keydir** *ssh_keydir*

Example:

```
sysadmin-vm:0_RP0(config-user-us1)#ssh_keydir dir1
```

Specify any alphanumeric value.

Step 8 **homedir** *homedir*

Example:

```
sysadmin-vm:0_RP0(config-user-us1)#homedir dir2
```

Specify any alphanumeric value.

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

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

end —Prompts user to take one of these actions:

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

Create Command Rules

Command rules are rules based on which users of a user group are either permitted or denied the use of certain commands. Command rules are associated to a user group and get applied to all users who are part of the user group.

A command rule is created by specifying whether an operation is permitted, or denied, on a command. This table lists possible operation and permission combinations:

Operation	**Accept Permission**	**Reject Permission**
Read (R)	Command is displayed on the CLI when "?" is used.	Command is not displayed on the CLI when "?" is used.
Execute (X)	Command can be executed from the CLI.	Command cannot be executed from the CLI.
Read and execute (RX)	Command is visible on the CLI and can be executed.	Command is neither visible nor executable from the CLI.

By default, all permissions are set to **Reject**.

Each command rule is identified by a number associated with it. When multiple command rules are applied to a user group, the command rule with a lower number takes precedence. For example, cmdrule 5 permits read access, while cmdrule10 rejects read access. When both these command rules are applied to the same user group, the user in this group gets read access because cmdrule 5 takes precedence.

As an example, in this task, the command rule is created to deny read and execute permissions for the "show platform" command.

Before you begin

Create an user group. See Create a User Group in System Admin VM, on page 28.

SUMMARY STEPS

1. **admin**
2. **config**
3. **aaa authorization cmdrules cmdrule** *command_rule_number*
4. **command** *command_name*
5. **ops {r | x | rx}**
6. **action** {**accept** | **accept_log** | **reject**}
7. **group** *user_group_name*
8. **context** *connection_type*
9. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

Step 1 **admin**

Example:

```
RP/0/RP0/CPU0:router# admin
```

Enters mode.

Step 2 **config**

Example:

```
sysadmin-vm:0_RP0#config
```

Enters System Admin Config mode.

Step 3 **aaa authorization cmdrules cmdrule** *command_rule_number*

Example:

```
sysadmin-vm:0_RP0(config)#aaa authorization cmdrules cmdrule 1100
```

Specify a numeric value as the command rule number. You can enter a 32 bit integer.

Important
Do no use numbers between 1 to 1000 because they are reserved by Cisco.

This command creates a new command rule (if it is not already present) and enters the command rule configuration mode. In the example, command rule "1100" is created.

Note
By default "cmdrule 1" is created by the system when the root-system user is created. This command rule provides "accept" permission to "read" and "execute" operations for all commands. Therefore, the root user has no restrictions imposed on it, unless "cmdrule 1" is modified.

Step 4 **command** *command_name*

Example:

```
sysadmin-vm:0_RP0(config-cmdrule-1100)#command "show platform"
```

Specify the command for which permission is to be controlled.

If you enter an asterisk '*' for **command**, it indicates that the command rule is applicable to all commands.

Step 5 **ops {r | x | rx}**

Example:

```
sysadmin-vm:0_RP0(config-cmdrule-1100)#ops rx
```

Specify the operation for which permission has to be specified:

- **r** — Read
- **x** — Execute
- **rx** — Read and execute

Step 6 **action** {**accept** | **accept_log** | **reject**}

Example:

```
sysadmin-vm:0_RP0(config-cmdrule-1100)#action reject
```

Specify whether users are permitted or denied the use of the operation.

- **accept** — users are permitted to perform the operation
- **accept_log**— users are permitted to perform the operation and every access attempt is logged.
- **reject**— users are restricted from performing the operation.

Step 7 **group** *user_group_name*

Example:

```
sysadmin-vm:0_RP0(config-cmdrule-1100)#group gr1
```

Specify the user group on which the command rule is applied.

Step 8 **context** *connection_type*

Example:

```
sysadmin-vm:0_RP0(config-cmdrule-1100)#context *
```

Specify the type of connection to which this rule applies. The connection type can be *netconf* (Network Configuration Protocol), *cli* (Command Line Interface), or *xml* (Extensible Markup Language). It is recommended that you enter an asterisk '*'; this indicates that the command rule applies to all connection types.

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

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

end —Prompts user to take one of these actions:

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

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

What to do next

Create data rules. See Create Data Rules, on page 37.

Create Data Rules

Data rules are rules based on which users of the user group are either permitted, or denied, accessing and modifying configuration data elements. The data rules are associated to a user group. The data rules get applied to all users who are part of the user group.

Each data rule is identified by a number associated to it. When multiple data rules are applied to a user group, the data rule with a lower number takes precedence.

Before you begin

Create an user group. See Create a User Group in System Admin VM, on page 28.

SUMMARY STEPS

1. **admin**
2. **config**
3. **aaa authorization datarules datarule** *data_rule_number*
4. **keypath** *keypath*
5. **ops** *operation*
6. **action** {**accept** | **accept_log** | **reject**}
7. **group** *user_group_name*
8. **context** *connection type*
9. **namespace** *namespace*
10. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

Step 1 **admin**

Example:

```
RP/0/RP0/CPU0:router# admin
```

Enters mode.

Step 2 **config**

Example:

```
sysadmin-vm:0_RP0#config
```

Enters System Admin Config mode.

Step 3 **aaa authorization datarules datarule** *data_rule_number*

Example:

```
sysadmin-vm:0_RP0(config)#aaa authorization datarules datarule 1100
```

Specify a numeric value as the data rule number. You can enter a 32 bit integer.

Important
Do no use numbers between 1 to 1000 because they are reserved by Cisco.

This command creates a new data rule (if it is not already present) and enters the data rule configuration mode. In the example, data rule "1100" is created.

Note
By default "datarule 1" is created by the system when the root-system user is created. This data rule provides "accept" permission to "read", "write", and "execute" operations for all configuration data. Therefore, the root user has no restrictions imposed on it, unless "datarule 1" is modified.

Step 4 **keypath** *keypath*

Example:

```
sysadmin-vm:0_RP0(config-datarule-1100)#keypath  /aaa/disaster-recovery
```

Specify the keypath of the data element. The keypath is an expression defining the location of the data element. If you enter an asterisk '*' for **keypath** , it indicates that the command rule is applicable to all configuration data.

Step 5 **ops** *operation*

Example:

```
sysadmin-vm:0_RP0(config-datarule-1100)#ops rw
```

Specify the operation for which permission has to be specified. Various operations are identified by these letters:

- c—Create
- d—Delete
- u—Update
- w— Write (a combination of create, update, and delete)
- r—Read
- x—Execute

Step 6 **action** {**accept** | **accept_log** | **reject**}

Example:

```
sysadmin-vm:0_RP0(config-datarule-1100)#action reject
```

Specify whether users are permitted or denied the operation.

- **accept** — users are permitted to perform the operation
- **accept_log**— users are permitted to perform the operation and every access attempt is logged
- **reject**— users are restricted from performing the operation

Step 7 **group** *user_group_name*

Example:

```
sysadmin-vm:0_RP0(config-datarule-1100)#group gr1
```

Specify the user group on which the data rule is applied. Multiple group names can also be specified.

Step 8 **context** *connection type*

Example:

```
sysadmin-vm:0_RP0(config-datarule-1100)#context *
```

Specify the type of connection to which this rule applies. The connection type can be *netconf* (Network Configuration Protocol), *cli* (Command Line Interface), or *xml* (Extensible Markup Language). It is recommended that you enter an asterisk '*', which indicates that the command applies to all connection types.

Step 9 **namespace** *namespace*

Example:

```
sysadmin-vm:0_RP0(config-datarule-1100)#namespace *
```

Enter asterisk '*' to indicate that the data rule is applicable for all namespace values.

Step 10 Use the **commit** or **end** command.

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

end —Prompts user to take one of these actions:

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

Change Disaster-recovery Username and Password

When you define the root-system username and password initially after starting the router, the same username and password gets mapped as the disaster-recovery username and password for the System Admin console. However, it can be changed.

The disaster-recovery username and password is useful in these scenarios:

- Access the system when the AAA database, which is the default source for authentication in System Admin console is corrupted.
- Access the system through the management port, when, for some reason, the System Admin console is not working.
- Create new users by accessing the System Admin console using the disaster-recovery username and password, when the regular username and password is forgotten.

Note On the router, you can configure only one disaster-recovery username and password at a time.

SUMMARY STEPS

1. **admin**
2. **config**
3. **aaa disaster-recovery username** *username* **password** *password*
4. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

Step 1 **admin**

Example:

```
RP/0/RP0/CPU0:router# admin
```

Enters mode.

Step 2 **config**

Example:

```
sysadmin-vm:0_RP0#config
```

Enters System Admin Config mode.

Step 3 **aaa disaster-recovery username** *username* **password** *password*

Example:

```
sysadmin-vm:0_RP0(config)#aaa disaster-recovery username us1 password pwd1
```

Specify the disaster-recovery username and the password. You have to select an existing user as the disaster-recovery user. In the example, 'us1' is selected as the disaster-recovery user and assigned the password as 'pwd1'. The password can be entered as a plain text or md5 digest string.

When you need to make use of the disaster recovery username, you need to enter it as *username***@localhost**.

Step 4 Use the **commit** or **end** command.

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

end —Prompts user to take one of these actions:

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

Recover Password using PXE Boot

If you are unable to login or lost your XR and System administration passwords, use the following steps to create new password. A lost password cannot be recovered, instead a new username and password must be created with a non-graceful PXE boot.

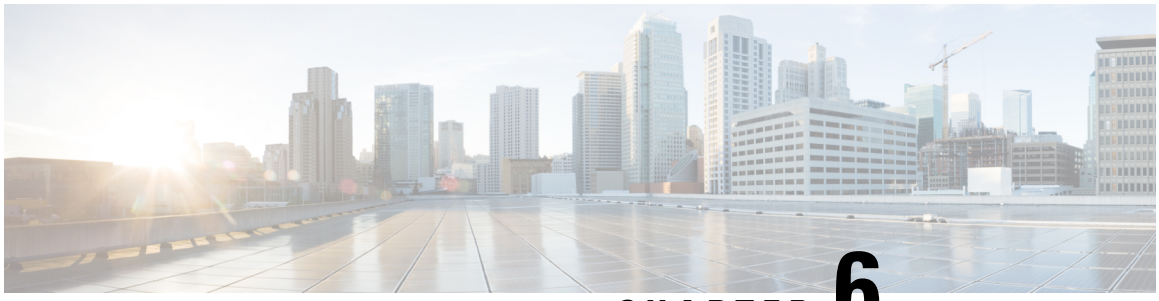
Procedure

Step 1 Boot the router using PXE.

Note
PXE boot is fully intrusive. The router state, configuration and image is reset.

To PXE boot a router, see Boot the Router Using iPXE, on page 130.

Step 2 Reset the password.



CHAPTER 6

Perform System Upgrade and Install Feature Packages

The system upgrade and package installation processes are executed using **install** commands on the router. The processes involve adding and activating the iso images (*.iso*) and feature packages on the router. These files are accessed from a network server and then activated on the router. If the installed package or SMU causes any issue on the router, it can be uninstalled.

The topics covered in this chapter are:

- Upgrading the System, on page 43
- View supported software upgrade or downgrade versions, on page 44
- Upgrading Features, on page 48
- Install Prepared Packages, on page 50
- Install Packages, on page 53
- Uninstall Packages, on page 58

Upgrading the System

Upgrading the system is the process of installing a new version of the Cisco IOS XR operating system on the router. The router comes preinstalled with the Cisco IOS XR image. However, you can install the new version in order to keep router features up to date. The system upgrade operation is performed from the XR VM. However, during system upgrade, the software that runs on both the XR VM and the System Admin VM get upgraded.

Note The 1G interface flaps twice instead of once in the Modular Port Adapter (MPA) NC55-MPA-12T-S after you reload any of these NCS 55A2 Fixed Chassis - NCS-55A2-MOD-SL, NCS-55A2-MOD-HD-S, NCS-55A2-MOD-HX-S, or NCS-55A2-MOD-SE-S.

Note If you insert a line card on a router that is running a lower version than the one the line card supports, the line card fails to boot. You must first upgrade the router to a software version that supports the line card, insert the line card and iPXE boot the line card.

Note If an interface on a router doesn't have a configuration and is brought up by performing no-shut operation, then upon router reload, the interface state changes to **admin-shutdown** automatically.

Note

- Ensure that you have adequate disk space.
- Run the **fsck** command to check the status of the file system, for a successful IOS XR upgrade. You must run the **fsck** command in the System Admin EXEC mode to install a System Admin package, and in the XR EXEC mode to install the XR package.
- All install commands are applicable in both the System Admin EXEC mode and in XR EXEC mode. System Admin install operations are done from XR EXEC mode.

Perform a system upgrade by installing a base package–Cisco IOS XR Unicast Routing Core Bundle. To install this bundle, run the **install** command. The filename for the Cisco IOS XR Unicast Routing Core Bundle bundle is *ncs5500-mini-x.iso*.

Caution Do not perform any install operations when the router is reloading.

Do not reload the router during an upgrade operation.

Note To enable hardware programming after upgrading the chassis from an older software version to IOS XR Release 7.6.x or later through ISSU, initiate a chassis reload. The chassis reload is mandatory, if you must enable a maximum transmission unit (MTU) value of 9646 on applicable interfaces.

Note Ensure that the system is on Cisco IOS XR Software Release 7.3.x, for a successful upgrade to Cisco IOS XR Software Release 7.6.x.

Cisco IOS XR supports RPM signing and signature verification for Cisco IOS XR RPM packages in the ISO and upgrade images. All RPM packages in the Cisco IOS XR ISO and upgrade images are signed to ensure cryptographic integrity and authenticity. This guarantees that the RPM packages haven't been tampered with and the RPM packages are from Cisco IOS XR. The private key, which is used for signing the RPM packages, is created and securely maintained by Cisco.

View supported software upgrade or downgrade versions

Cisco routers come preinstalled with Cisco IOS XR software. You can upgrade the software to access new features and fixes, or downgrade it if needed. To take advantage of the latest features and software improvements, we recommend that you keep your router updated with the current version.

Table 2: Feature History Table

Feature Name	Release Information	Description
Supported software upgrade or downgrade IOS XR versions	Release 7.5.1	You can determine whether a software version can be upgraded or downgraded to another version using this functionality. Before an actual upgrade or downgrade process, you can also view the hardware or software limitations that could cause the upgrade or downgrade to fail. This feature helps you plan successful software upgrades or downgrades. This feature introduces the **show install upgrade-matrix** command.

Compatibility checks for Cisco IOS XR software upgrades and downgrades

The compatibility check feature for Cisco IOS XR software facilitates choosing a release that follows Cisco-certified upgrade and downgrade paths, asking critical questions such as:

- Which upgrade or downgrade releases are supported for my current release
- If I want to upgrade from Release X to Release Y, does my router support this upgrade
- Are there any bridging SMUs that must be installed before upgrading?

This feature checks whether your current release can upgrade or downgrade to a specified target release. This automatic validation occurs during the start of a software upgrade or downgrade using the **install replace** command. If the validation fails, the upgrade is blocked, and the system notifies you of the reason for the failure. This validation allows you to proactively determine upgrade or downgrade compatibility thus saving plannig effort.

The feature provides prerequisites and limitations for a specific upgrade or downgrade such as:

- Required bridging SMU RPMs
- Blocking SMU RPMs
- Unsupported hardware
- Caveats or restrictions

If needed, bypass automatic validation by including the **force** keyword with the **install replace** command. When you use this option, the system displays warning messages if the upgrade fails, but it does not stop the software upgrade. Explore **force ?** for any impacts beyond this.

Show commands for software upgrade and downgrade

You can view the software upgrade and downgrade information using the **show** commands in this table or through the operational data.

Command	Description
show install upgrade-matrix running	Displays all supported software upgrades from the current version according to the support data installed on the running system
show install upgrade-matrix iso *path-to-ISO*	Displays details about the software upgrade from the current version to the version of the target ISO according to the support data in both the running system and the ISO image
show install upgrade-matrix iso *path-to-ISO* **all**	Displays all supported software upgrades from any version according to the support data in the target ISO image
show install upgrade-matrix iso *path-to-ISO* **from-running**	Displays details about the software upgrade from the current version to the version of ISO according to the support matrices in both the running system and the target ISO image

Supported software upgrade from running version

This example shows all supported releases for upgrade from the current version on the NCS 5500 router:

```
Router#show install upgrade-matrix running
Fri Jul 29 10:12:47.740 IST
This may take a while ...

The current software can be upgraded from and downgraded to the following releases:

================================================================================
From      To        Bridge SMUs Required        Caveats
================================================================================
7.0.2     7.5.1     None                        None
--------------------------------------------------------------------------------
7.5.1     7.0.2     None                        None
--------------------------------------------------------------------------------
7.5.1     7.4.1     None                        None
--------------------------------------------------------------------------------
7.5.1     7.1.3     None                        None
--------------------------------------------------------------------------------
7.5.1     7.1.2     None                        None
--------------------------------------------------------------------------------
7.5.1     7.2.1     None                        None
--------------------------------------------------------------------------------
7.5.1     7.2.2     None                        None
--------------------------------------------------------------------------------
7.5.1     7.3.1     None                        None
--------------------------------------------------------------------------------
7.5.1     7.3.2     None                        None
--------------------------------------------------------------------------------
7.4.1     7.5.1     None                        None
--------------------------------------------------------------------------------
7.1.3     7.5.1     None                        None
--------------------------------------------------------------------------------
7.1.2     7.5.1     None                        None
--------------------------------------------------------------------------------
7.2.1     7.5.1     None                        None
```

```
--------------------------------------------------------------------------------
7.2.2     7.5.1     None                          None
--------------------------------------------------------------------------------
7.3.1     7.5.1     None                          None
--------------------------------------------------------------------------------
7.3.2     7.5.1     None                          None
--------------------------------------------------------------------------------
```

Supported releases to upgrade software from current version to target version

This example shows the supported release to upgrade software from the current version to a target version.

```
Router#show install upgrade-matrix iso /harddisk:/ncs5500-goldenk9-x-7.5.2-rev1.iso
Fri Jul 29 10:16:43.233 IST
This may take a while ...
Upgrade from the current software [7.5.1] to 7.5.2 is supported

================================================================================
From      To        Bridge SMUs Required        Caveats
================================================================================
7.5.1     7.5.2     None                        None
--------------------------------------------------------------------------------
```

The current image has the upgrade matrix that specifies only its supported upgrade or downgrade versions up to a certain version. If you want to determine the upgrade path of a newer version of ISO that is higher than the version in the current matrix, the upgrade matrix from the new ISO provides the supported upgrade or downgrade paths.

Supported releases from current version to an ISO version

This example shows the software upgrade paths, downgrade paths, and restrictions to an upgrade from the current version to the target ISO version:

```
Router#show install upgrade-matrix iso /harddisk:/ncs5500-goldenk9-x-7.5.2-rev1.iso all
Fri Jul 29 10:17:02.826 IST
This may take a while ...

7.5.2 can be upgraded from and downgraded to the following releases:


================================================================================
From      To        Bridge SMUs Required        Caveats
================================================================================
7.5.1     7.5.2     None                        None
--------------------------------------------------------------------------------
7.4.2     7.5.2     None                        None
--------------------------------------------------------------------------------
7.4.1     7.5.2     None                        None
--------------------------------------------------------------------------------
7.5.2     7.4.2     None                        None
--------------------------------------------------------------------------------
7.5.2     7.6.1     None                        None
--------------------------------------------------------------------------------
7.5.2     7.4.1     None                        None
--------------------------------------------------------------------------------
7.5.2     7.2.1     None                        None
--------------------------------------------------------------------------------
7.5.2     7.2.2     None                        None
```

```
--------------------------------------------------------------------------------
7.5.2     7.3.1     None                        None
--------------------------------------------------------------------------------
7.5.2     7.3.2     None                        None
--------------------------------------------------------------------------------
7.2.1     7.5.2     None                        None
--------------------------------------------------------------------------------
7.2.2     7.5.2     None                        None
--------------------------------------------------------------------------------
7.6.1     7.5.2     None                        None
--------------------------------------------------------------------------------
7.3.1     7.5.2     None                        None
--------------------------------------------------------------------------------
7.3.2     7.5.2     None                        None
--------------------------------------------------------------------------------
```

Supported releases from running version to an ISO version

This example displays details about the software upgrade from the current version to the version of ISO according to the support matrices in both the running system and the target ISO image:

```
Router#show install upgrade-matrix iso /harddisk:/ncs5500-goldenk9-x-7.5.2-rev1.iso
from-running
Fri Jul 29 10:17:35.583 IST
This may take a while ...
Upgrade from the current software [7.5.1] to 7.5.2 is supported

================================================================================
From      To        Bridge SMUs Required        Caveats
================================================================================
7.5.1     7.5.2     None                        None
--------------------------------------------------------------------------------
```

Upgrading Features

Upgrading features is the process of deploying new features and software patches on the router. Perform a feature upgrade by installing packages. Perform a software patch installation by installing Software Maintenance Upgrade (SMU) files.

Installing a package on the router installs specific features that are part of that package. Cisco IOS XR Software is divided into various software packages; this enables you to select the features to run on your router. Each package contains components that perform a specific set of router functions, such as routing, security, and so on.

For example, the components of the routing package are split into individual RPMs such as BGP and OSPF. BGP is a part of the base software version and is a mandatory RPM, and hence can't be removed. However, you can add and remove optional RPMs such as OSPF as required.

The naming convention of the package is `<platform>-<pkg>-<pkg version>-<release version>.<architecture>.rpm`.

Feature	Package
Forwarding	ncs5500-fwding-1.0.0.0-<*release-number*>.x86_64.rpm

Feature	Package
BGP	ncs5500-bgp-1.0.0.0-<*release-number*>.x86_64.rpm
mpls-te-rsvp	ncs5500-mpls-te-rsvp-1.0.0.0-<*release-number*>.x86_64.rpm
k9sec	ncs5500-k9sec-1.0.0.0-<*release-number*>.x86_64.rpm
mgbl	ncs5500-mgbl-2.0.0.0-<*release-number*>.x86_64.rpm
mpls	ncs5500-mpls-1.0.0.0-<*release-number*>.x86_64.rpm
infra	ncs5500-infra-1.0.0.0-<*release-number*>.x86_64.rpm
os	ncs5500-os-1.0.0.0-<*release-number*>.x86_64.rpm
routing	ncs5500-routing-1.0.0.0-<*release-number*>.x86_64.rpm
security	ncs5500-security-1.0.0.0-<*release-number*>.x86_64.rpm
os-support	ncs5500-os-support-1.0.0.0-<*release-number*>.x86_64.rpm

Use the **install** commands to install packages and SMUs. For more information about the install process, see Install Packages, on page 53.

Note

- Ensure that you have adequate disk space.
- Run the **fsck** command to check the status of the file system, for a successful IOS XR upgrade. You must run the **fsck** command in the System Admin EXEC mode to install a System Admin package, and in the XR EXEC mode to install the XR package.
- All install commands are applicable in both the System Admin EXEC mode and in XR EXEC mode. System Admin install operations are done from XR EXEC mode.

There are separate packages and SMUs for the XR VM and the System Admin VM. They can be identified by their filenames.

The XR and System Admin packages and SMUs can be activated from XR and System Admin VMs.

You can alternatively perform a cross VM operation, by activating or deactivating the System Admin packages and SMUs from XR.

Third-Party SMUs

Consider these points while activating and deactivating third-party SMUs:

- To activate a third-party SMU, you should have a corresponding base package.
- When you activate a third-party SMU, the corresponding third-party base package state is inactive, this is an expected behavior.
- To deactivate a third-party SMU, ensure that you activate the corresponding third-party base package. Third-party SMUs deactivated explicitly might lead to triages to the install team.

Note All SMUs are bundled together with the base package in a TAR file

Note All Cisco RPMs have the platform name in the filename. For example, **ncs5500**-sysadmin.

Install Prepared Packages

A system upgrade or feature upgrade is performed by activating the ISO image file, packages, and SMUs. It is possible to prepare these installable files before activation. During the prepare phase, preactivation checks are made and the components of the installable files are loaded on to the router setup. The prepare process runs in the background and the router is fully usable during this time. When the prepare phase is over, all the prepared files can be activated instantaneously. The advantages of preparing before activation are:

- If the installable file is corrupted, the prepare process fails. This provides an early warning of the problem. If the corrupted file was activated directly, it might cause router malfunction.
- Directly activating an ISO image for system upgrade takes considerable time during which the router is not usable. However, if the image is prepared before activation, not only does the prepare process run asynchronously, but when the prepared image is subsequently activated, the activation process too takes less time. As a result, the router downtime is considerably reduced.
- It performs a disk-space check that is required for a successful operation. This quantifies the disk-space deficit, and provides you possible alternatives to free up space in the filesystem.
- It performs a package compatibility check. This ensures that all the required installation packages are available. For any package compatibility check error, details of the package and version are logged.

Complete this task to upgrade the system and install packages by making use of the prepare operation.

Note Depending on whether you are installing a System Admin package or a XR package, execute the **install** commands in the System Admin EXEC mode or XR EXEC mode respectively. All **install** commands are applicable in both these modes. System Admin install operations can be done from XR mode.

Procedure

Step 1 Add the required ISO image and packages to the repository.

For details, see Install Packages, on page 53.

Step 2 **show install repository**

Example:

```
RP/0/RP0/CPU0:router#show install repository
```

Perform this step to verify that the required installable files are available in the repository. Packages are displayed only after the "install add" operation is complete.

Step 3 **show install request**

Example:

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

(Optional) Displays the operation ID of the add operation and its status. The operation ID can be later used to execute the **activate** command.

```
Install operation 8 is still in progress
```

Step 4 Execute one of these:

- **install prepare** *package_name*
- **install prepare id** *operation_id*

Example:

```
RP/0/RP0/CPU0:router#install prepare ncs5500-mpls-1.0.0.0-r60023I.x86_64.rpm
```

or

```
RP/0/RP0/CPU0:router#install prepare id 8
```

The prepare process takes place. This operation is performed in asynchronous mode. The **install prepare** command runs in the background, and the EXEC prompt is returned as soon as possible.

If you use the operation ID, all packages that were added in the specified operation are prepared together. For example, if 5 packages are added in operation 8, by executing **install prepare id 8**, all 5 packages are prepared together. You do not have to prepare the packages individually.

Step 5 **show install prepare**

Example:

```
RP/0/RP0/CPU0:router#show install prepare
```

Displays packages that are prepared. From the result, verify that all the required packages have been prepared.

Step 6 **install activate**

Example:

```
RP/0/RP0/CPU0:router#install activate
```

All the packages that have been prepared are activated together to make the package configurations active on the router.

Note
You should not specify any package name or operation ID in the CLI.

Activations of some SMUs require manual reload of the router. When such SMUs are activated, a warning message is displayed to perform reload. The components of the SMU get activated only after the reload is complete. Perform router reload immediately after the execution of the **install activate** command is completed.

Step 7 **show install active**

Example:

```
RP/0/RP0/CPU0:router#show install active
```

Displays packages that are active.

```
Node 0/RP0/CPU0 [RP]
Boot Partition: xr_lv70
Active Packages: 24
ncs5500-xr-6.0.0 version=6.0.0 [Boot image]
ncs5500-k9sec-1.0.0.0-r600
ncs5500-mgbl-2.0.0.0-r600
ncs5500-mpls-1.0.0.0-r600
ncs5500-mpls-te-rsvp-1.0.0.0-r600
ncs5500-infra-2.0.0.2-r600.CSCxr22222
ncs5500-iosxr-fwding-2.0.0.2-r600.CSCxr22222
ncs5500-iosxr-fwding-2.0.0.5-r600.CSCxr90016
ncs5500-iosxr-fwding-2.0.0.1-r600.CSCxr55555
ncs5500-iosxr-fwding-2.0.0.6-r600.CSCxr90017
ncs5500-dpa-1.0.0.1-r600.CSCxr90002
ncs5500-dpa-1.0.0.2-r600.CSCxr90004
ncs5500-dpa-fwding-1.0.0.1-r600.CSCxr90005
ncs5500-k9sec-1.0.0.1-r600.CSCxr80008
ncs5500-os-support-1.0.0.1-r600.CSCxr90013
ncs5500-os-support-1.0.0.2-r600.CSCxr90014
ncs5500-fwding-1.0.0.2-r600.CSCxr90011
ncs5500-fwding-1.0.0.5-r600.CSCxr90019
ncs5500-fwding-1.0.0.1-r600.CSCxr90010
ncs5500-fwding-1.0.0.4-r600.CSCxr90018
ncs5500-mgbl-2.0.0.2-r600.CSCxr80009
ncs5500-mpls-1.0.0.1-r600.CSCxr33333
ncs5500-mpls-te-rsvp-1.0.0.2-r600.CSCxr33335
```

From the result, verify that on all RPs and LCs, the same image and package versions are active.

Step 8 **install commit**

Example:

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

Installing Packages: Related Commands

Related Commands	Purpose
show install log	Displays the log information for the install process; this can be used for troubleshooting in case of install failure.
show install package	Displays the details of the packages that have been added to the repository. Use this command to identify individual components of a package.
install prepare clean	Clears the prepare operation and removes all the packages from the prepared state.

What to do next

- After performing a system upgrade, upgrade FPD by using the **upgrade hw-module location all fpd all** command from the System Admin EXEC mode. The progress of FPD upgrade process can be monitored using the **show hw-module fpd** command in the System Admin EXEC mode. Reload the router after the FPD upgrade is completed.
- Verify the installation using the **install verify packages** command.

- Uninstall the packages or SMUs if their installation causes any issues on the router. See Uninstall Packages.

Note ISO images cannot be uninstalled. However, you can perform a system downgrade by installing an older ISO version.

Install Packages

Complete this task to upgrade the system or install a patch. The system upgrade is done using an ISO image file, while the patch installation is done using packages and SMUs. You can also include SMUs in an upgrade operation along with mini ISO.

This task is also used to install *.rpm* files. The *.rpm* file contains multiple packages and SMUs that are merged into a single file. The packaging format defines one RPM per component, without dependency on the card type.

Note

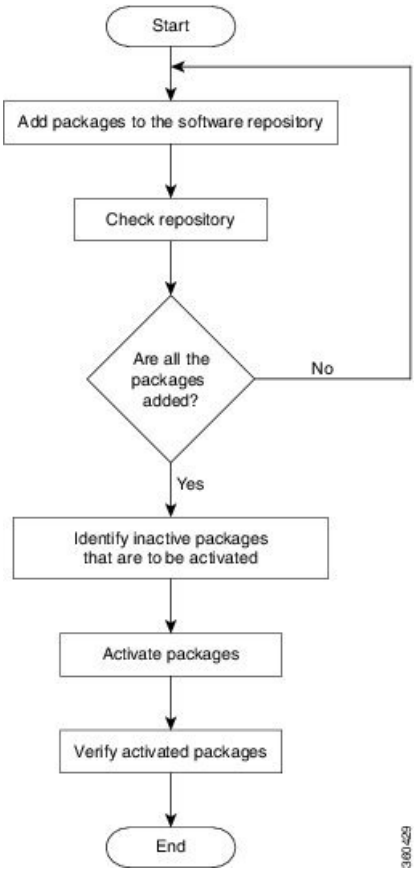
- Ensure that you have adequate disk space.
- Run the **fsck** command to check the status of the file system, for a successful IOS XR upgrade. You must run the **fsck** command in the System Admin EXEC mode to install a System Admin package, and in the XR EXEC mode to install the XR package.
- All install commands are applicable in both the System Admin EXEC mode and in XR EXEC mode. System Admin install operations are done from XR EXEC mode.

Note

- The system upgrade is supported only from XR EXEC mode.
- While the System Admin package can be executed using **install** commands in the System Admin EXEC mode and XR EXEC mode, the XR package can only be executed using the install commands in XR EXEC mode. All **install** commands are applicable in both these modes.
- While the System Admin SMUs can be installed in System Admin EXEC mode and XR EXEC mode, the XR SMUs can only be installed through the XR EXEC mode.
- Install operation over IPv6 is not supported.

The workflow for installing a package is shown in this flowchart.

Figure 2: Installing Packages Workflow



Before you begin

- You can add a package to the repository from a local disk in the router, from an inserted USB disk, or from a remote repository accessible through the management port or any data port.

  If the installable file is located on a USB in the router's USB port, use the **show media** or **show filesystem** commands. The installable file is labeled as either **usb:** or **disk2:** in the command outputs.

  If the installable file is located in a remote location that can be accessed through the management port, you must configure and bring up the management port to ensure reachability to the remote location. For more information, see the *Configure the Management Port* section in the *Bring-up the Router* chapter.

Procedure

Step 1 Execute one of these:

- **install add source** *<http or shttp transfer protocol>/package_path/ filename1 filename2 ...*
- **install add source** *<tftp transfer protocol>/package_path/ filename1 filename2 ...*
- **install add source** *<ftp or sftp transfer protocol>//user@server:/package_path/ filename1 filename2 ...*
- **install add source** *disk2: filename1 filename2 ...*

Example:

```
RP/0/RP0/CPU0:router#install add source /harddisk:/ ncs5500-mpls-1.0.0.0-r731.x86_64.rpm
ncs5500-mgbl-1.0.0.0-r732.x86_64.rpm
```

or

```
RP/0/RP0/CPU0:router#install add source sftp://root@8.33.5.15:/auto/ncs/package/
ncs55000-mcast-1.0.0.0-731.x86_64.rpm ncs5500-iosxr-mpls-1.0.0.0-732.x86_64.rpm
```

or

```
RP/0/RP0/CPU0:router#install add source /harddisk:/ ncs5500-mpls-1.0.0.0-<release-number>.x86_64.rpm
 ncs5500-mgbl-1.0.0.0-<release-number>.x86_64.rpm
```

or

```
RP/0/RP0/CPU0:router#install add source sftp://root@8.33.5.15:/auto/ncs/package/
ncs55000-mcast-1.0.0.0-<release-number>.x86_64.rpm
ncs5500-iosxr-mpls-1.0.0.0-<release-number>.x86_64.rpm
```

Note
A space must be provided between the *package_path* and *filename*.

The software files are unpacked from the package, validated, and then added to the software repository. This operation might take time depending on the size of the files being added. The operation is performed in asynchronous mode. The **install add** command runs in the background, and the EXEC prompt is returned when all files are unpacked.

Note
The repositories for the XR VM and the System Admin VM are different. The system automatically adds a routing package to the XR VM repository and a system administration package to the System Admin VM repository.

Step 2 **show install request**

Example:

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

(Optional) Displays the operation ID of the add operation and its status. The operation ID can be later used to execute the **activate** command.

```
Install operation 8 is still in progress
```

Step 3 **show install repository**

Example:

```
RP/0/RP0/CPU0:router#show install repository
```

Displays packages that are added to the repository. Packages are displayed only after the `install add` operation is complete.

Step 4 **show install inactive**

Example:

```
RP/0/RP0/CPU0:router#show install inactive
```

Displays inactive packages that are present in the repository. Only inactive packages can be activated.

Step 5 Execute one of these:

- **install activate** *package_name*
- **install activate id** *operation_id*

Example:

```
RP/0/RP0/CPU0:router#install activate ncs5500-mpls-1.0.0.0-<release-number>.x86_64.rpm
ncs5500-mgbl-2.0.0.0-<release-number>.x86_64.rpm
```

or

```
RP/0/RP0/CPU0:router#install activate id 8
```

The *operation_id* is that of the **install add** operation, see Install Packages, on page 53 Step Step 2, on page 55. This command can also be run from the Sys Admin mode. The package configurations are made active on the router. As a result, new features and software fixes take effect. This operation is performed in asynchronous mode, as this is the default. The **install activate** command runs in the background, and the EXEC prompt is returned.

You can run the activate operation either through the synchronous mode or by selecting the `sync` option from the CLI.

If you use the operation ID, all packages that were added in the specified operation are activated together. For example, if 5 packages are added in operation ID 8, by executing **install activate id 8**, all 5 packages are activated together. You do not have to activate the packages individually.

Activation does not happen instantaneously, but takes some time. When activation completes, the system reloads automatically. For restart SMU activation, the SMU takes effect once the processes impacted by the SMU are restarted.

If the SMU has dependency on both XR VM and System Admin VM, perform the reload after activating the SMU in both VMs so that they take effect simultaneously. To reload the router, use the **hw-module location all reload** command from the System Admin EXEC mode.

Step 6 **show install active**

Example:

```
RP/0/RP0/CPU0:router#show install active
```

Displays packages that are active.

```
Node 0/RP0/CPU0 [RP]
Boot Partition: xr_lv70
Active Packages: 24
ncs5500-xr-<release-number> version=<release-number> [Boot image]
ncs5500-k9sec-1.0.0.0-<release-number>
ncs5500-mgbl-2.0.0.0-<release-number>
ncs5500-mpls-1.0.0.0-<release-number>
ncs5500-mpls-te-rsvp-1.0.0.0-<release-number>
ncs5500-infra-2.0.0.2-<release-number>.CSCxr22222
ncs5500-iosxr-fwding-2.0.0.2-<release-number>.CSCxr22222
ncs5500-iosxr-fwding-2.0.0.5-<release-number>.CSCxr90016
ncs5500-iosxr-fwding-2.0.0.1-<release-number>.CSCxr55555
ncs5500-iosxr-fwding-2.0.0.6-<release-number>.CSCxr90017
ncs5500-dpa-1.0.0.1-<release-number>.CSCxr90002
ncs5500-dpa-1.0.0.2-<release-number>.CSCxr90004
ncs5500-dpa-fwding-1.0.0.1-<release-number>.CSCxr90005
ncs5500-k9sec-1.0.0.1-<release-number>.CSCxr80008
ncs5500-os-support-1.0.0.1-<release-number>.CSCxr90013
ncs5500-os-support-1.0.0.2-<release-number>.CSCxr90014
ncs5500-fwding-1.0.0.2-<release-number>.CSCxr90011
ncs5500-fwding-1.0.0.5-<release-number>.CSCxr90019
ncs5500-fwding-1.0.0.1-<release-number>.CSCxr90010
ncs5500-fwding-1.0.0.4-<release-number>.CSCxr90018
ncs5500-mgbl-2.0.0.2-<release-number>.CSCxr80009
ncs5500-mpls-1.0.0.1-<release-number>.CSCxr33333
ncs5500-mpls-te-rsvp-1.0.0.2-<release-number>.CSCxr33335
```

From the result, verify that the same image and package versions are active on all RPs and LCs.

Table 3: Example: Installing Packages: Related Commands

Related Commands	Purpose
show install log	Displays the log information for the install process; this can be used for troubleshooting in case of install failure.
show install package	Displays the details of the packages that have been added to the repository. Use this command to identify individual components of a package.
install prepare	Makes pre-activation checks on an inactive package, to prepare it for activation.
show install prepare	Displays the list of package that have been prepared and are ready for activation.

Step 7 **install commit**

Example:

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

Commits the Host, XR, and System Admin newly active software.

Note
On Multi-SDR mode, you can use the **install commit sdr** to commit just the sdr from where the CLI is being triggered. For more information, see Secure Domain Router Commands.

What to do next

- After performing a system upgrade, upgrade FPD by using the **upgrade hw-module location all fpd all** command from the System Admin EXEC mode. The progress of FPD upgrade process can be monitored using the **show hw-module fpd** command in the System Admin EXEC mode. Reload the router after the FPD upgrade is completed.
- Verify the installation using the **install verify packages** command.
- Uninstall the packages or SMUs if their installation causes any issues on the router. See Uninstall PackagesUninstall Packages, on page 58.

Note If you are upgrading power supply modules for NC55-PWR-3KW-DC and NC55-PWR-3KW-2HV, ensure that you first upgrade SC IO FPGA by using **upgrade hw-module location <SC0/SC1> fpd all** command from Sysadmin prompt followed by the **upgrade hw-module location pm-all fpd** command, to upgrade FPD.

Finally use **hw-module location <SC0/SC1> reload**command from Sysadmin prompt to reload the shelf controller.

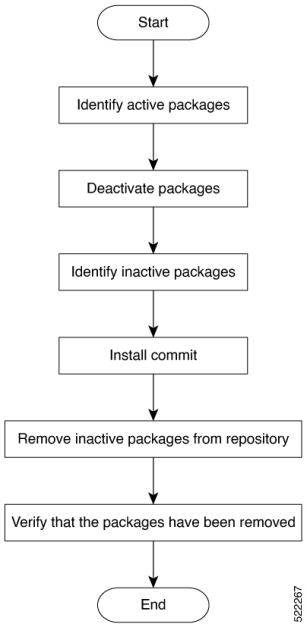
Uninstall Packages

Complete this task to uninstall a package. All router functionalities that are part of the uninstalled package are deactivated. Packages that are added in the XR VM cannot be uninstalled from the System Admin VM. However, the cross VM operation allows System Admin packages to be deactivated from XR as well.

Note Installed ISO images cannot be uninstalled. Also, kernel SMUs that install third party SMU on host, XR VM and System Admin VM, cannot be uninstalled. However, subsequent installation of ISO image or kernel SMU overwrites the existing installation.

The workflow for uninstalling a package is shown in this flowchart.

Figure 3: Uninstalling Packages Workflow



This task uninstalls XR VM packages. If you need to uninstall System Admin packages, run the same commands from the System Admin EXEC mode.

Procedure

Step 1 **show install active**

Example:

```
RP/0/RP0/CPU0:router#show install active
```

Displays active packages. Only active packages can be deactivated.

```
Node 0/RP0/CPU0 [RP]
```

```
Boot Partition: xr_lv70
Active Packages: 24
ncs5500-xr-6.0.0 version=6.0.0 [Boot image]
ncs5500-k9sec-1.0.0.0-r600
ncs5500-mgbl-2.0.0.0-r600
ncs5500-mpls-1.0.0.0-r600
ncs5500-mpls-te-rsvp-1.0.0.0-r600
ncs5500-infra-2.0.0.2-r600.CSCxr22222
ncs5500-iosxr-fwding-2.0.0.2-r600.CSCxr22222
ncs5500-iosxr-fwding-2.0.0.5-r600.CSCxr90016
ncs5500-iosxr-fwding-2.0.0.1-r600.CSCxr55555
ncs5500-iosxr-fwding-2.0.0.6-r600.CSCxr90017
ncs5500-dpa-1.0.0.1-r600.CSCxr90002
ncs5500-dpa-1.0.0.2-r600.CSCxr90004
ncs5500-dpa-fwding-1.0.0.1-r600.CSCxr90005
ncs5500-k9sec-1.0.0.1-r600.CSCxr80008
ncs5500-os-support-1.0.0.1-r600.CSCxr90013
ncs5500-os-support-1.0.0.2-r600.CSCxr90014
ncs5500-fwding-1.0.0.2-r600.CSCxr90011
ncs5500-fwding-1.0.0.5-r600.CSCxr90019
ncs5500-fwding-1.0.0.1-r600.CSCxr90010
ncs5500-fwding-1.0.0.4-r600.CSCxr90018
ncs5500-mgbl-2.0.0.2-r600.CSCxr80009
ncs5500-mpls-1.0.0.1-r600.CSCxr33333
ncs5500-mpls-te-rsvp-1.0.0.2-r600.CSCxr33335
```

Step 2 Execute one of these:

- **install deactivate** *package_name*
- **install deactivate id** *operation_id*

Example:

```
RP/0/RP0/CPU0:router#install deactivate ncs5500-mpls-1.0.0.0-r60023I.x86_64.rpm
ncs5500-mgbl-2.0.0.0-r60023I.x86_64.rpm
```

or

```
RP/0/RP0/CPU0:router#install deactivate id 8
```

The *operation_id* is the ID from **install add** operation. All features and software patches associated with the package are deactivated. You can specify multiple package names and deactivate them simultaneously.

If you use the operation ID, all packages that were added in the specified operation are deactivated together. You do not have to deactivate the packages individually. If System admin packages were added as a part of the **install add** operation (of the ID used in deactivate) then those packages will also be deactivated.

Step 3 **show install inactive**

Example:

```
RP/0/RP0/CPU0:router#show install inactive
```

The deactivated packages are now listed as inactive packages. Only inactive packages can be removed from the repository.

Step 4 **install commit**

Step 5 **install remove** *package_name*

Example:

```
RP/0/RP0/CPU0:router#install remove ncs5500-mpls-1.0.0.0-r60023I.x86_64.rpm
ncs5500-mgbl-2.0.0.0-r60023I.x86_64.rpm
```

The inactive packages are removed from the repository.

Use the **install remove** command with the **id** *operation-id* keyword and argument to remove all packages that were added for the specified operation ID.

You can also use the **install remove inactive all** to remove all inactive packages from XR and System Admin.

Step 6 **show install repository**

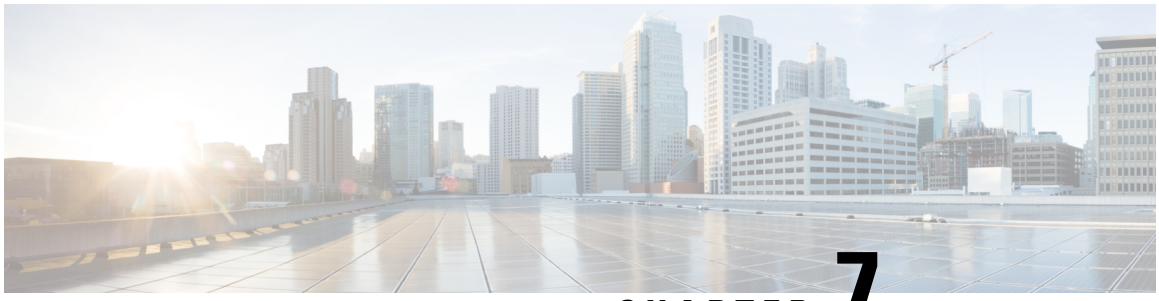
Example:

```
RP/0/RP0/CPU0:router#show install repository
```

Displays packages available in the repository. The package that are removed are no longer displayed in the result.

What to do next

Install required packages. .

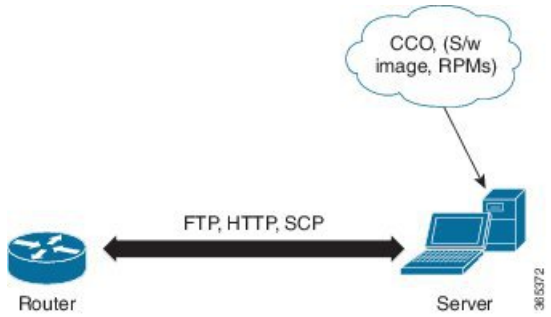


CHAPTER 7

Manage Automatic Dependency

Flexible packaging supports automatic dependency management. While you update an RPM, the system automatically identifies all relevant dependent packages and updates them.

Figure 4: Flow for Installation (base software, RPMs and SMUs)



Until this release, you downloaded the software image and required RPMs from CCO on a network server (the repository), and used the **install add** and the **install activate** commands to add and activate the downloaded files on the router. Then, you manually identify relevant dependent RPMs, to add and activate them.

With automatic dependency management, you need not identify dependent RPMs to individually add and activate them. You can execute new install command to identify and install dependent RPMs automatically.

The command **install source** adds and activates packages. The command **install replace** adds and activates packages in a given golden ISO (GISO).

Note

1. Cisco IOS XR Version 6.0.2 and later does not provide third party and host package SMUs as part of automatic dependency management (**install source** command). The third party and host package SMUs must be installed separately, and in isolation from other installation procedures (installation of SMUs and RPMs in IOS XR or admin containers).
2. From Cisco IOS XR Version 6.5.2 onwards, it is possible to update the `mini.iso` file by using the **install source** command.

The rest of this chapter contains these sections:

- Update RPMs and SMUs, on page 62
- Upgrade Base Software Version, on page 63

• Downgrade an RPM, on page 64

Update RPMs and SMUs

An RPM may contain a fix for a specific defect, and you may need to update the system with that fix. To update RPMs and SMUs to a newer version, use the **install source** command. When this command is issued for a particular RPM, the router communicates with the repository, and downloads and activates that RPM. If the repository contains a dependent RPM, the router identifies that dependent RPM and installs that too.

The syntax of the **install source** command is:

install source *repository* **[rpm]**

Four scenarios in which you can use the **install source** command are:

- **When a package name is not specified**

  When no package is specified, the command updates the latest SMUs of all installed packages.

  ```
  install source [repository]
  ```

Note From Cisco IOS XR Version 6.1.1 onwards, if the `mini.iso` file is not specified, then it is not added as part of the update. Even if the repository contains the `mini.iso` file, it is not installed.

```
install source scp://<username>@<server>/my/path/of/packages
noprompt
```

- **When a package name is specified**

  If the package name is specified, the command installs that package, updates the latest SMUs of that package, along with its dependencies. If the package is already installed, only the SMUs of that package are installed. (SMUs that are already installed are skipped.)

  ```
  install source [repository] ncs5500-mpls.rpm
  ```

- **When a package name and version number are specified**

  If a particular version of package needs to be installed, the complete package name must be specified; that package is installed along with the latest SMUs of that package present in the repository.

  ```
  install source [repository] ncs5500-mpls-1.0.2.0-r710.x86_64.rpm
  ```

- **When an SMU is specified**

  If an SMU is specified, that SMU is downloaded and installed, along with its dependent SMUs.

  ```
  install source [repository] ncs5500-mpls-1.2.0.1-r611.CSCus12345.x86_64.rpm
  ```

- **When a list of packages (containing the mini.iso file) is specified**

  From Cisco IOS XR Version 6.5.2 onwards, if a list of packages (containing the `mini.iso` file) is specified, all the packages in the list and the `mini.iso` file are automatically added as part of the update.

  ```
  install source scp://<username>@<server>/my/path/of/packages [List of packages]
  noprompt
  ```

- **When the mini.iso file is specified**

From Cisco IOS XR Version 6.1.1 onwards, if the `mini.iso` file is specified during the update, then the file is installed with all RPMs and SMUs from the repository.

```
install source scp://<username>@<server>/my/path/of/packages [mini.iso] noprompt
```

Upgrade Base Software Version

You can upgrade to a newer version of the base software when it becomes available. To upgrade to the latest base software version, use the **install source** command. With the upgrade of the base version, RPMs that are currently available on the router are also upgraded.

Note SMUs are not upgraded as part of this process.

The syntax of the **install source** command is:

```
install source repository
```

Note VRF and TPA on dataport is not supported. If the server is reachable only through non-default VRF interface, the file must already be retrieved using ftp, sfp, scp, http or https protocols.

Note Default routes (0.0.0.0/0) cannot be copied onto Linux due to TPA implementation.

You can use the **install source** command when:

- **The version number is specified**

  The base software (.mini) is upgraded to the specified version; all installed RPMs are upgraded to the same release version.

  ```
  install source [repository] version <version> asr9k-mini-x64-<version>.iso
  ```

  For example,

  ```
  install source repository version 7.0.1 asr9k-mini-x64-7.0.1.iso
  ```

  You can also automatically fetch the .mini file and RPMs of the required release and proceed with the upgrade.

  ```
  install source repository asr9k-mini-x64-7.0.1.iso
  ```

- **The version number for an RPM is specified**

  When performing a system upgrade, the user can choose to have an optional RPM to be of a different release (from that of the base software version); that RPM can be specified.

  ```
  install source repository version 6.2.2
  ncs5500-mpls-1.0.2.0-r623.x86_64.rpm
  ```

Downgrade an RPM

An RPM can be downgraded after it is activated. RPMs are of the following types:

- **Hostos RPM:** The RPM contains `hostos` in the name.

  For example:

  - <platform>-sysadmin-hostos-6.5.1-r651.CSChu77777.host.arm
  - <platform>-sysadmin-hostos-6.5.1-r651.CSChu77777.admin.arm
  - <platform>-sysadmin-hostos-6.5.1-r651.CSChu77777.host.x86_64
  - <platform>-sysadmin-hostos-6.5.1-r651.CSChu77777.admin.x86_64

- **Non-hostos RPM:** The RPM does not contain `hostos` in the name.

  For example:

  - <platform>-sysadmin-system-6.5.1-r651.CSCvc12346

To deactivate the RPMs, perform the following steps:

- **Downgrade Hostos RPM**
  - Scenario 1: To downgrade to version 06 from the active version 09:
    1. Download the version 06 hostos RPMs, and add the RPMs.

       ```
       install add source [repository]
       <platform>-sysadmin-hostos-6.5.1.06-r65108I.CSChu44444.host.arm
       <platform>-sysadmin-hostos-6.5.1.06-r65108I.CSChu44444.admin.arm
       <platform>-sysadmin-hostos-6.5.1.06-r65108I.CSChu44444.host.x86_64
       <platform>-sysadmin-hostos-6.5.1.06-r65108I.CSChu44444.admin.x86_64
       ```

    2. Activate the downloaded RPMs.

       ```
       install activate [repository]
       <platform>-sysadmin-hostos-6.5.1.06-r65108I.CSChu44444.host.arm
       <platform>-sysadmin-hostos-6.5.1.06-r65108I.CSChu44444.admin.arm
       <platform>-sysadmin-hostos-6.5.1.06-r65108I.CSChu44444.host.x86_64
       <platform>-sysadmin-hostos-6.5.1.06-r65108I.CSChu44444.admin.x86_64
       ```

    3. Commit the configuration.

       ```
       install commit
       ```

  - Scenario 2: Deactivate hostos RPM by activating base RPM, consider version 09 is active:
    1. Activate the base RPM.

       ```
       install activate <platform>-sysadmin-hostos-6.5.1.08I-r65108I.admin.arm
       <platform>-sysadmin-hostos-6.5.1.08I-r65108I.host.arm
       <platform>-sysadmin-hostos-6.5.1.08I-r65108I.admin.x86_64
       <platform>-sysadmin-hostos-6.5.1.08I-r65108I.host.x86_64
       ```

       For example, if RPM `ncs5500-sysadmin-hostos-6.5.1-r651.CSChu44444.host.arm` is the RPM installed, then `ncs5500-sysadmin-hostos-6.5.1-r651.host.arm` is its base RPM.
    2. Commit the configuration.

```
install commit
```

The downgrade for third-party RPMs is similar to the hostos RPMs. To downgrade a SMU, activate the lower version of the SMU. If only one version of SMU is present, the base RPM of the SMU must be activated.

Note Hostos and third-party RPMs cannot be deactivated. Only activation of different versions is supported.

- **Downgrade Non-Hostos RPM**

  1. Deactivate the RPM to downgrade to earlier version of RPM.

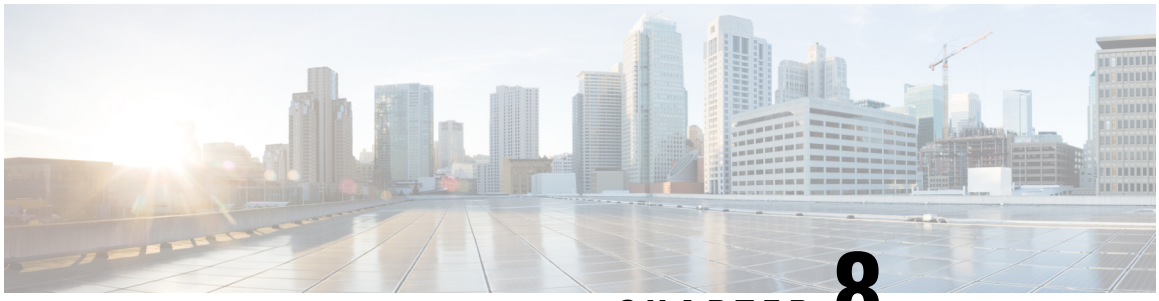
     ```
     install deactivate <platform>-<rpm-name>
     ```

  2. Check the active version of the RPM.

     ```
     show install active
     ```

  3. Commit the configuration.

     ```
     install commit
     ```



CHAPTER 8

Customize Installation using Golden ISO

Golden ISO (GISO) is a customized ISO that a user can build to suit the installation requirement. The user can customize the installable image to include the standard base image with the basic functional components, and add additional RPMs, SMUs and configuration files based on requirement.

The ease of installation and the time taken to seamlessly install or upgrade a system plays a vital role in a cloud-scale network. An installation process that is time-consuming and complex affects the resiliency and scale of the network. The GISO simplifies the installation process, automates the installation workflow, and manages the dependencies in RPMs and SMUs automatically.

GISO is built using a build script `gisobuild.py` available on the github location Github location.

When a system boots with GISO, additional SMUs and RPMs in GISO are installed automatically, and the router is pre-configured with the XR configuration in GISO. For more information about downloading and installing GISO, see Install Golden ISO, on page 77.

The capabilities of GISO can be used in the following scenarios:

- Initial deployment of the router
- Software disaster recovery
- System upgrade from one base version to another
- System upgrade from same base version but with additional SMUs
- Install update to identify and update dependant packages

- Limitations, on page 68
- Customize Installation using Golden ISO, on page 68
- Golden ISO Workflow, on page 68
- Build Golden ISO, on page 69
- Install Golden ISO, on page 77

Limitations

The following are the known problems and limitations with the customized ISO:

- Building and booting GISO for asynchronous package (a package of different release than the ISO) is not supported.
- Verifying the XR configuration is not supported in the GISO build script `gisobuild.py`.

- Renaming a GISO build and then installing from the renamed GISO build is not supported.
- Install operation over IPv6 is not supported.

Customize Installation using Golden ISO

Golden ISO (GISO) is a customized ISO that a user can build to suit the installation requirement. The user can customize the installable image to include the standard base image with the basic functional components, and add additional RPMs, SMUs and configuration files based on requirement.

The ease of installation and the time taken to seamlessly install or upgrade a system plays a vital role in a cloud-scale network. An installation process that is time-consuming and complex affects the resiliency and scale of the network. The GISO simplifies the installation process, automates the installation workflow, and manages the dependencies in RPMs and SMUs automatically.

GISO is built using a build script `gisobuild.py` available on the github location Github location.

When a system boots with GISO, additional SMUs and RPMs in GISO are installed automatically, and the router is pre-configured with the XR configuration in GISO. For more information about downloading and installing GISO, see Install Golden ISO, on page 77.

The capabilities of GISO can be used in the following scenarios:

- Initial deployment of the router
- Software disaster recovery
- System upgrade from one base version to another
- System upgrade from same base version but with additional SMUs
- Install update to identify and update dependant packages

Limitations

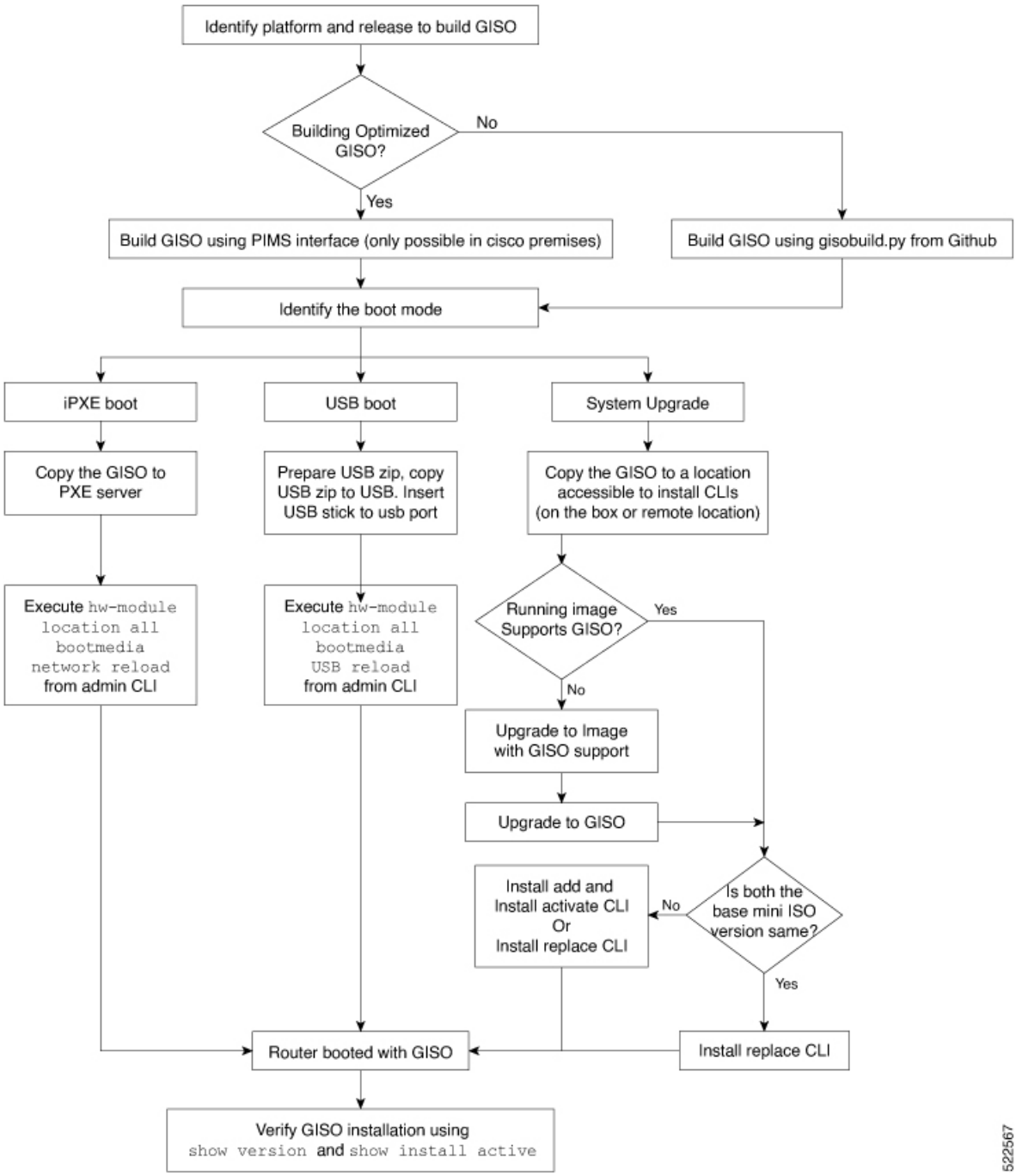
The following are the known problems and limitations with the customized ISO:

- Building and booting GISO for asynchronous package (a package of different release than the ISO) is not supported.
- Verifying the XR configuration is not supported in the GISO build script `gisobuild.py`.
- Renaming a GISO build and then installing from the renamed GISO build is not supported.
- Install operation over IPv6 is not supported.

Golden ISO Workflow

The following image shows the workflow for building and installing golden ISO.

Figure 5: Golden ISO Workflow



Build Golden ISO

The customized ISO is built using Cisco Golden ISO (GISO) build script `gisobuild.py` available on the Github location.

The GISO build script supports automatic dependency management, and provides these functionalities:

- Builds RPM database of all the packages present in package repository.
- Scans the repositories and selects the relevant Cisco RPMs that matches the input iso.
- Skips and removes third-party RPMs that are not SMUs of already existing third-party base package in mini-x.iso.
- Displays an error and exits build process if there are multiple base RPMs of same release but different versions.
- Performs compatibility check and dependency check for all the RPMs. For example, the child RPM ncs5500-mpls-te-rsvp is dependent on the parent RPM ncs5500-mpls . If only the child RPM is included, the Golden ISO build fails.

Build Golden ISO Using Script

Table 4: Feature History Table

Feature Name	Release Information	Description
Enhanced Golden ISO Build Tool	Release 7.5.1	This enhancement provides you with the flexibility to use the `gisobuild.py` tool to build GISO images using Cisco IOS XR software commands, YAML-based template file, or docker capability to suit your customized install requirements. When you build a GISO, you can also specify Zero Touch Provisioning (ZTP) initialization file, script initialization file, Cisco IOS XR configuration file, and SMUs in addition to using the base image and optional RPMs to automatically provision the router.

To build GISO, provide the following input parameters to the script:

- Base mini-x.iso (mandatory)
- XR configuration file (optional)
- one or more Cisco-specific SMUs for host, XR and System admin (optional)
- one or more third-party SMUs for host, XR and System admin (optional)
- Label for golden ISO (optional)
- Optional RPMs
- ZTP initialization `ztp.ini` file (optional)

- Script initialization `script.ini` file (optional)

The GISO script does not support verification of XR configuration.

Note To successfully add k9sec RPM to GISO, change the permission of the file to `644` using the **chmod** command.

```
chmod 644 [k9 sec rpm]
```

Cisco IOS XR, Release 7.5.1 introduces enhancements to the `gisobuild.py` GISO build tool. You can also add a `ztp.ini` ZTP initialization and `script.ini` Script initialization file. The ZTP configuration is applied on the router when the current software version is replaced or rolled back to a version with GISO image, and is used whenever ZTP is run to automatically provision the router. The tool supports more than one repository. You can use CLI command, docker, or a YAML file to build GISO.

Note

- For Cisco NCS 5500 and Cisco NCS 5000 series routers, set the `migration` value to `false`.
- Set the `clean` option to `true` if you use the same build directory after the first GISO is created. Ensure that you set the option to `true` for every successive GISO build.
- Set the `docker` option to `true` if you are building GISO using docker.
- Ensure that the format and syntax of the YAML file is intact to avoid errors when building a GISO. For example, if the `:` symbol is missing, or if an unsupported symbol is used in the template, the GISO build displays errors.

The `gisobuild.py` tool can be run either natively or on systems where docker service is enabled and has the ability to pull published docker images. Prefer building the image using the docker as it does not require additional privileges:.

Note The `full-iso` option is used to build a full ISO image `xrv9k-full-x-7.5.1.iso` specific to Cisco IOS XRv 9000 routers. Starting Cisco IOS XR, Release 7.8.1, the full ISO image must not be used to build GISO.

To build GISO, perform the following steps:

Before you begin

- The system where GISO is built must meet the following requirements:
  - System must have Python version 3.6 and later.
  - System must have free disk space of minimum 12 GB.
  - Verify that the Linux utilities `mount`, `rm`, `cp`, `umount`, `zcat`, `chroot`, `mkisofs` are present in the system. These utilities will be used by the script. Ensure privileges are available to execute all of these Linux commands. However, if you are using docker, these utilities are not required.
  - Kernel version of the system must be later than 3.16 or later than the version of kernel of Cisco ISO.

- Verify that a `libyaml rpm` supported by the Linux kernel is available to successfully `import yaml` in the tool.
- User should have proper permission for security rpm(k9sec-rpm) in rpm repository, else security rpm would be ignored for Golden ISO creation.

- The system from where the `gisobuild.py` script is executed must have root credentials. This is not mandatory if you are building the image within a docker container.
- We recommend that you perform a `git pull` operation before you use the `gisobuild.py` script to ensure you obtain the latest version of the script for the Python version.

Procedure

Step 1 Copy the script `gisobuild.py` from the Github repository to an offline system or external server where the GISO will be built. Ensure that this system meets the pre-requisites described above in the *Before You Begin* section.

Step 2 Run the script `gisobuild.py` and provide parameters to build the golden ISO off the router. Ensure that all RPMs and SMUs are present in the same directory or on a repository. The number of RPMs and SMUs that can be used to build the Golden ISO is 64.

```
usage: gisobuild.py [-h] [--iso ISO] [--repo REPO [REPO ...]]
                    [--bridging-fixes BRIDGE_FIXES [BRIDGE_FIXES ...]]
                    [--xrconfig XRCONFIG] [--ztp-ini ZTP_INI] [--label LABEL]
                    [--out-directory OUT_DIRECTORY] [--yamlfile CLI_YAML] [--clean]
                    [--pkglist PKGLIST [PKGLIST ...]] [--script SCRIPT] [--docker]
                    [--x86-only] [--migration]
                    [--remove-packages REMOVE_PACKAGES [REMOVE_PACKAGES ...]]
                    [--skip-usb-image] [--copy-dir COPY_DIRECTORY]
                    [--clear-bridging-fixes] [--verbose-dep-check] [--debug]
                    [--version]

Utility to build Golden ISO for IOS-XR.

optional arguments:
  -h, --help            show this help message and exit
  --iso ISO             Path to Mini.iso/Full.iso file
  --repo REPO [REPO ...]
                        Path to RPM repository. For LNT, user can specify .rpm, .tgz,
                        .tar filenames, or directories. RPMs are only used if already
                        included in the ISO, or specified by the user via the
                        --pkglist option.
  --bridging-fixes BRIDGE_FIXES [BRIDGE_FIXES ...]
                        Bridging rpms to package. For EXR, takes from-release or rpm
                        names; for LNT, the user can specify the same file types as for
                        the --repo option.
  --xrconfig XRCONFIG   Path to XR config file
  --ztp-ini ZTP_INI     Path to user ztp ini file
  --label LABEL, -l LABEL
                        Golden ISO Label
  --out-directory OUT_DIRECTORY
                        Output Directory
  --yamlfile CLI_YAML   Cli arguments via yaml
  --clean               Delete output dir before proceeding
  --pkglist PKGLIST [PKGLIST ...]
                        Packages to be added to the output GISO. For eXR: optional rpm
                        or smu to package. For LNT: either full package filenames or
                        package names for user installable packages can be specified.
```

```
                        Full package filenames can be specified to choose a particular
                        version of a package, the rest of the block that the package is
                        in will be included as well. Package names can be specified to
                        include optional packages in the output GISO.
  --docker, --use-container
                        Build GISO in container environment.Pulls and run pre-built
                        container image to build GISO.
  --version             Print version of this script and exit

EXR only build options:
  --script SCRIPT       Path to user executable script executed as part of bootup post
                        activate.
  --x86-only            Use only x86_64 rpms even if other architectures are
                        applicable.
  --migration           To build Migration tar only for ASR9k

LNT only build options:
  --remove-packages REMOVE_PACKAGES [REMOVE_PACKAGES ...]
                        Remove RPMs, specified in a comma separated list. These are are
                        matched against user installable package names, and must be the
                        whole package name, e.g: xr-bgp
  --skip-usb-image      Do not build the USB image
  --copy-dir COPY_DIRECTORY
                        Copy built artefacts to specified directory if provided. The
                        specified directory must already exist, be writable by the
                        builder and must not contain a previously built artefact with
                        the same name.
  --clear-bridging-fixes
                        Remove all bridging bugfixes from the input ISO
  --verbose-dep-check   Verbose output for the dependency check.
  --debug               Output debug logs to console
```

Example

Example: Build Docker-Based GISO Image

In this example, a GISO image is built using docker.

```
[root@xr src]# ./gisobuild.py --docker --iso /auto/ncs5500giso/ncs5500-mini-x-7.5.1.iso
--repo /auto/ncs5500giso --pkglist ncs5500-bgp-2.0.0.0-r751.x86_64.rpm
ncs5500-eigrp-1.0.0.0-r751.x86_64.rpm ncs5500-isis-2.1.0.0-r751.x86_64.rpm
ncs5500-k9sec-3.1.0.0-r751.x86_64.rpm
ncs5500-li-1.0.0.0-r751.x86_64.rpm ncs5500-mcast-3.0.0.0-r751.x86_64.rpm
ncs5500-mgbl-3.0.0.0-r751.x86_64.rpm
ncs5500-mpls-2.1.0.0-r751.x86_64.rpm ncs5500-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
ncs5500-ospf-2.0.0.0-r751.x86_64.rpm
ncs5500-parser-2.0.0.0-r751.x86_64.rpm --label dockerbasedgiso

Local System requirements check [PASS]
Pulling gisobuild image from hub. Please wait...
\
Done...
System requirements check [PASS]

Platform: ncs5500 Version: 7.5.1

Scanning repository [/auto/ncs5500giso]...

Building RPM Database...

Total 11 RPM(s) present in the repository path provided in CLI
```

```
[ 1] ncs5500-mpls-2.1.0.0-r751.x86_64.rpm
[ 2] ncs5500-mgbl-3.0.0.0-r751.x86_64.rpm
[ 3] ncs5500-bgp-2.0.0.0-r751.x86_64.rpm
[ 4] ncs5500-parser-2.0.0.0-r751.x86_64.rpm
[ 5] ncs5500-isis-2.1.0.0-r751.x86_64.rpm
[ 6] ncs5500-mcast-3.0.0.0-r751.x86_64.rpm
[ 7] ncs5500-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
[ 8] ncs5500-ospf-2.0.0.0-r751.x86_64.rpm
[ 9] ncs5500-li-1.0.0.0-r751.x86_64.rpm
[10] ncs5500-eigrp-1.0.0.0-r751.x86_64.rpm
[11] ncs5500-k9sec-3.1.0.0-r751.x86_64.rpm

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

        (+) ncs5500-ospf-2.0.0.0-r751.x86_64.rpm
        (+) ncs5500-bgp-2.0.0.0-r751.x86_64.rpm
        (+) ncs5500-parser-2.0.0.0-r751.x86_64.rpm
        (+) ncs5500-mcast-3.0.0.0-r751.x86_64.rpm
        (+) ncs5500-li-1.0.0.0-r751.x86_64.rpm
        (+) ncs5500-eigrp-1.0.0.0-r751.x86_64.rpm
        (+) ncs5500-mgbl-3.0.0.0-r751.x86_64.rpm
        (+) ncs5500-mpls-2.1.0.0-r751.x86_64.rpm
        (+) ncs5500-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
        (+) ncs5500-isis-2.1.0.0-r751.x86_64.rpm
        (+) ncs5500-k9sec-3.1.0.0-r751.x86_64.rpm

        ...RPM signature check [PASS]

Skipping following rpms from repository since they are already present in base ISO:

        (-) ncs5500-parser-2.0.0.0-r751.x86_64.rpm
        (-) ncs5500-bgp-2.0.0.0-r751.x86_64.rpm

        ...RPM compatibility check [PASS]

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

XR rpms:
        ncs5500-mcast-3.0.0.0-r751.x86_64.rpm
        ncs5500-mgbl-3.0.0.0-r751.x86_64.rpm
        ncs5500-isis-2.1.0.0-r751.x86_64.rpm
        ncs5500-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
        ncs5500-eigrp-1.0.0.0-r751.x86_64.rpm
        ncs5500-mpls-2.1.0.0-r751.x86_64.rpm
        ncs5500-ospf-2.0.0.0-r751.x86_64.rpm
        ncs5500-li-1.0.0.0-r751.x86_64.rpm
        ncs5500-k9sec-3.1.0.0-r751.x86_64.rpm

        ...Golden ISO creation SUCCESS.

Golden ISO Image Location: /var/tmp/giso/gisobuild-toolkit-master/src/output_gisobuild/
                           ncs5500-golden-x-7.5.1-dockerbasedgiso.iso
```

View that the GISO file is created succesfully.

```
[root@xr src]# ls
exrmod  gisobuild.py  lntmod  output_gisobuild  utils

[root@xr src]# cd output_gisobuild/
[root@xr output_gisobuild]# ls
img_built_name.txt logs ncs5500-golden-x-7.5.1-dockerbasedgiso.iso
rpms_packaged_in_giso.txt
```

Example: Build YAML-Based GISO Image

YAML is a markup file that serves as a template to provide the package list and manage the build options.

The following example shows a sample YAML template:

```
# Options below correspond to the tool input options.
# --iso ISO             Path to Mini.iso/golden.iso file
# --repo REPO [REPO ...]
#                       Path to list of RPM repositories. RPMs are only used if already
#                       included in the ISO, or specified by the user via the --pkglist
option.
# --pkglist PKGLIST [PKGLIST ...]
#                       Optional list of rpm or smu to add to the ISO.
# --remove-packages REMOVE_PACKAGES [REMOVE_PACKAGES ...]
#                      Remove named RPMs, specified in a space separated list. Valid build
#                       option for LNT only. eXR builds simply ignores this option.
# --bridging-fixes BRIDGE_FIXES [BRIDGE_FIXES ...]
#                       Bridging rpms to package. Takes from-release (supported for eXR)
#                       or rpm names.
# --xrconfig XRCONFIG   Path to XR config file
# --ztp-ini ZTP_INI     Path to user ztp ini file
# --script SCRIPT       Path to user executable script executed as part of
#                       bootup post activate. Valid build option for eXR only.
#                       LNT builds simply ignores.
# --label LABEL         Golden ISO Label
# --out-directory OUT_DIRECTORY
#                       Output Directory. Built GISO and logs will be available post
gisobuild.
# --copy-directory COPY_DIRECTORY
#                      Copy built artefacts to specified directory if provided. Valid build
#                       option for LNT only. eXR build ignores this option.
# --yamlfile CLI_YAML   Cli arguments via yaml.
# --clean               Delete output dir before proceeding.
# --migration           To build Migration tar only for ASR9k. Valid build option for eXR
only.
#                       LNT builds simply ignore this option.
# --docker              Load and run pre-built docker image. Valid build option for eXR
only.
#                       LNT builds simply ignore this option.
# --x86-only            Use only x86_64 rpms even if other architectures are applicable.
Valid build
#                       option for eXR only. LNT builds simply ignore this option.
# --version             Print version of this script and exit

packages:
    iso: <path-to-iso>
    repo:
        - <path-to-repo1>
        - <path-to-repo2>
    pkglist:
        - <pkg1>
        - <pkg2>
    bridge-fixes:
        upgrade-from-release:
            - <dotted-release-1>
            - <dotted-release-2>
        rpms:
            - <pkg1>
            - <pkg2>
    remove_packages:
        - <pkg1>
        - <pkg2>

user-content:
```

```
    script: <path-to-script-sh>
    xrconfig: <path-to-router.cfg>
    ztp-ini: <path-to-ztp.ini>

output:
    label: <giso-label>
    out-directory: <path-to-output-directory>
    clean: <true/false>

options:
    docker: <true/false>
    migration: <true/false>
    x86-only: <true/false>
```

In this example, you configure a YAML file with the required files:

```
packages:
    iso: /auto/751_repo/ncs5500-mini-x-7.5.1.iso
    repo:
        - /auto/751_repo/
    pkglist:
        - ncs5500-bgp-2.0.0.0-r751.x86_64.rpm
        - ncs5500-eigrp-1.0.0.0-r751.x86_64.rpm
        - ncs5500-isis-2.1.0.0-r751.x86_64.rpm
        - ncs5500-li-1.0.0.0-r751.x86_64.rpm
        - ncs5500-mcast-3.0.0.0-r751.x86_64.rpm
        - ncs5500-mgbl-3.0.0.0-r751.x86_64.rpm
        - ncs5500-mpls-2.1.0.0-r751.x86_64.rpm
        - ncs5500-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
        - ncs5500-ospf-2.0.0.0-r751.x86_64.rpm
        - ncs5500-parser-2.0.0.0-r751.x86_64.rpm
        - ncs5500-k9sec-3.1.0.0-r751.x86_64.rpm
        - ncs5500-mcast-3.0.0.1-r751.CSCxr33333.x86_64.rpm
        - ncs5500-os-5.0.0.1-r751.CSCxr11111.x86_64.rpm
        - ncs5500-sysadmin-hostos-7.5.1-r751.CSCho99999.admin.x86_64.rpm
        - ncs5500-sysadmin-hostos-7.5.1-r751.CSCho99999.host.x86_64.rpm
        - ncs5500-sysadmin-topo-7.5.1-r751.CSCcv55555.x86_64.rpm
        - ncs5500-sysadmin-system-7.5.1-r751.CSCcv44444.x86_64.rpm
        - openssh-scp-6.6p1.p1-r0.5.0.r751.CSCtp11111.xr.x86_64.rpm
        - cisco-klm-zermatt-0.1.p1-r0.0.r751.CSCtp11111.xr.x86_64.rpm

    remove_rpms: []

user-content:
    script: script.sh
    xrconfig: /auto/751_repo/gisoxrconfig.cfg
    ztp-ini:  /auto/751_repo/ztp.ini

output:
    label: 751_yaml_install
    out-directory: /auto/751_repo/
    clean: true

options:
    docker: false
    full-iso: false
    migration: false
    x86-only: false
```

If you do not want to specify the list of packages and parameters via CLI, you can use the YAML file template.

```
[directory-path]$ ./src/gisobuild.py --yamlfile <input-yaml-cfg>
```

To override any input in the YAML configuration file, use the corresponding CLI options.

```
[directory-path]$ ./src/gisobuild.py --yamlfile <input-yaml-cfg> --label <new-label>
```

This new label overrides the label specified in the YAML file.

When the host machine does not have its package dependencies met, but allows pulling and running docker images, enable the docker option in YAML file to `true` and run the command:

```
[directory-path]$ ./src/gisobuild.py --yamlfile <input-yaml-cfg>
```

where, the `input-yaml-cfg` has the docker option set to `true`.

What to do next

Install the GISO image on the router.

Install Golden ISO

Golden ISO (GISO) automatically performs the following actions:

- Installs host and system admin RPMs.
- Partitions repository and TFTP boot on RP.
- Creates software profile in system admin and XR modes.
- Installs XR RPMs. Use **show install active** command to see the list of RPMs.
- Applies XR configuration. Use **show running-config** command in XR mode to verify.

Procedure

Step 1 Download GISO image to the router using one of the following options:

- **PXE boot:** when the router is booted, the boot mode is identified. After detecting PXE as boot mode, all available ethernet interfaces are brought up, and DHClient is run on each interface. DHClient script parses HTTP or TFTP protocol, and GISO is downloaded to the box.

  When you bring up a router using the PXE boot mode, existing configurations are removed. To recover smart licensing configurations like Permanent License Reservation (PLR), enable these configurations after the router comes up.

  ```
  Router#configure
  Router(config)#license smart reservation
  Router(config)#commit
  ```

- **USB boot or Disk Boot:** when the USB mode is detected during boot, and GISO is identified, the additional RPMs and XR configuration files are extracted and installed.
- **System Upgrade:** when the system is upgraded, GISO can be installed using **install add**, **install activate**, or using **install replace** commands.

  Important
  To replace the current version and packages on the router with the version from GISO, note the change in command and format.

  - In versions prior to Cisco IOS XR Release 6.3.3, 6.4.x and 6.5.1, use the **install update** command:

```
install update source <source path> <Golden-ISO-name> replace
```

- In Cisco IOS XR Release 6.5.2 and later, use the **install replace** command.

```
install replace <absolute-path-of-Golden-ISO>
```

Note
To create a Bootable External USB Disk, do the following:

- Ensure that the USB Boot Disk has a minimum storage of 8GB, and that you have root/admin or appropriate permission to create bootable disk on linux machine.

a. Copy and execute usb-install script on the Linux machine to create a bootable external USB.

```
Router#admin

Router#./linux/misc/scripts/create_usb_zip ncs5500 ncs5500-goldenk9-x.iso-7.5.01.v1
  adding: EFI/ (stored 0%)
  adding: EFI/boot/ (stored 0%)
  adding: EFI/boot/grub.cfg (deflated 66%)
  adding: EFI/boot/bootx64.efi (deflated 67%)
  adding: boot/ (stored 0%)
  adding: boot/install-image.iso (deflated 1%)
Zip file created - usb_boot.zip
Router# ls -ltr usb_boot.zip
-rw-r--r-- 1 user eng 1448680576 Sep 14 04:13 usb_boot.zip
Router#
```

b. Reset the RSP/RP and plug in bootable USB to RSP/RP's front panel. The USB will get detected in ROMMON. Note that when the system is in ROMMON, and if you add a front panel external USB, the USB will not be detected until the RSP/RP is reset.

The options to upgrade the system are as follows:

- **system upgrade from a non-GISO (image that does not support GISO) to GISO image:** If a system is running a version1 with an image that does not support GISO, the system cannot be upgraded directly to version2 of an image that supports GISO. Instead, the version1 must be upgraded to version2 mini ISO, and then to version2 GISO.
- **system upgrade in a release from version1 GISO to version2 GISO:** If both the GISO images have the same base version but different labels, **install add** and **install activate** commands does not support same version of two images. Instead, using **install source** command installs only the delta RPMs. System reload is based on restart type of the delta RPMs.

  Using **install replace** command performs a system reload, irrespective of the difference between ISO and the existing version.
- **system upgrade across releases from version1 GISO to version2 GISO:** Both the GISO images have different base versions. Use **install add** and **install activate** commands, or **install replace** command to perform the system upgrade. The router reloads after the upgrade with the version2 GISO image.

Step 2 Run the **show install repository all** command in System Admin mode to view the RPMs and base ISO for host, system admin and XR.

```
sysadmin-vm:0_RP0#show install repository all
Admin repository
---------------------
```

```
ncs5500-sysadmin-6.2.2
ncs5500-sysadmin-hostos-6.2.2-r622.CSCcv10001.admin.x86_64
ncs5500-sysadmin-hostos-6.2.2-r622.CSCcv10001.admin.arm
ncs5500-sysadmin-system-6.2.2-r622.CSCcv10005.x86_64
ncs5500-sysadmin-system-6.2.2-r622.CSCcv10005.arm
....
XR repository
------------------
ncs5500-iosxr-mgbl-3.0.0.0-r622.x86_64
ncs5500-xr-6.2.2
....
Host repository
---------------------
host-6.2.2
```

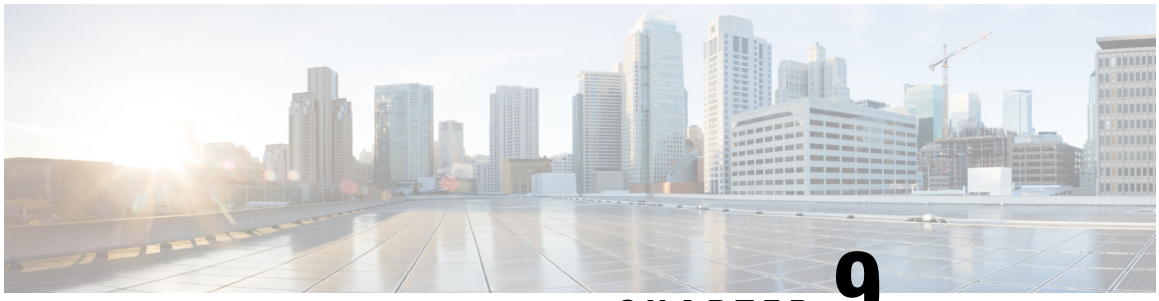
Step 3 Run the **show install package <golden-iso>** command to display the list of RPMs, and packages built in GISO.

Note
To list RPMs in the GISO, the GISO must be present in the install repository.

```
Router#show install package ncs5500-goldenk9-x64-6.2.2

This may take a while ...
    ISO Name: ncs5500-goldenk9-x64-6.2.2
    ISO Type: bundle
    ISO Bundled: ncs5500-mini-x64-6.2.2
    Golden ISO Label: temp
    ISO Contents:
        ISO Name: ncs5500-xr-6.2.2
        ISO Type: xr
        rpms in xr ISO:
            iosxr-os-ncs5500-64-5.0.0.0-r622
            iosxr-ce-ncs5500-64-3.0.0.0-r622
            iosxr-infra-ncs5500-64-4.0.0.0-r622
            iosxr-fwding-ncs5500-64-4.0.0.0-r622
            iosxr-routing-ncs5500-64-3.1.0.0-r6122

        ISO Name: ncs5500-sysadmin-6.2.2
        ISO Type: sysadmin
        rpms in sysadmin ISO:
            ncs5500-sysadmin-topo-6.2.2-r622
            ncs5500-sysadmin-shared-6.2.2-r622
            ncs5500-sysadmin-system-6.2.2-r622
            ncs5500-sysadmin-hostos-6.2.2-r622.admin
        ...

        ISO Name: host-6.2.2
        ISO Type: host
        rpms in host ISO:
            ncs5500-sysadmin-hostos-6.2.2-r622.host

        Golden ISO Rpms:
            xr rpms in golden ISO:
                ncs5500-k9sec-x64-2.2.0.1-r622.CSCxr33333.x86_64.rpm
                openssh-scp-6.6p1.p1-r0.0.CSCtp12345.xr.x86_64.rpm
                openssh-scp-6.6p1-r0.0.xr.x86_64.rpm
                ncs5500-mpls-x64-2.1.0.0-r622.x86_64.rpm
                ncs5500-k9sec-x64-2.2.0.0-r622.x86_64.rpm

            sysadmin rpms in golden ISO:
                ncs5500-sysadmin-system-6.2.2-r622.CSCcv11111.x86_64.rpm
                ncs5500-sysadmin-system-6.2.2-r622.CSCcv11111.arm.rpm
                openssh-scp-6.6p1-r0.0.admin.x86_64.rpm
                openssh-scp-6.6p1-r0.0.admin.arm.rpm
```

```
    openssh-scp-6.6p1.p1-r0.0.CSCtp12345.admin.x86_64.rpm
    openssh-scp-6.6p1.p1-r0.0.CSCtp12345.admin.arm.rpm
    ncs5500-sysadmin-hostos-6.2.2-r622.CSCcv10001.admin.x86_64.rpm
    ncs5500-sysadmin-hostos-6.2.2-r622.CSCcv10001.admin.arm.rpm

host rpms in golden ISO:
    openssh-scp-6.6p1-r0.0.host.x86_64.rpm
    openssh-scp-6.6p1-r0.0.host.arm.rpm
    openssh-scp-6.6p1.p1-r0.0.CSCtp12345.host.x86_64.rpm
    openssh-scp-6.6p1.p1-r0.0.CSCtp12345.host.arm.rpm
```

The ISO, SMUs and packages in GISO are installed on the router.



CHAPTER 9

Deploy Router Using Classic ZTP

Manually deploying network devices in a large-scale environment requires skilled workers and is time consuming.

With Zero Touch Provisioning (ZTP), you can seamlessly provision thousands of network devices accurately within minutes and without any manual intervention. This can be easily defined using a configuration file or script using shell or python.

ZTP provides multiple options, such as:

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

Table 5: Feature History Table

Feature Name	Release Information	Feature Description
Zero Touch Provisioning	Release 7.3.1	With this release, you can seamlessly provision thousands of network devices accurately within minutes and without any manual intervention. This can be easily defined using a configuration file or script using shell or phyton.

Benefits of Using ZTP

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

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

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

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

Use Cases

The following are some of the useful use cases for ZTP:

- Using ZTP to install Chef
- Using ZTP to integrate IOS-XR with NSO
- Using ZTP to install Puppet

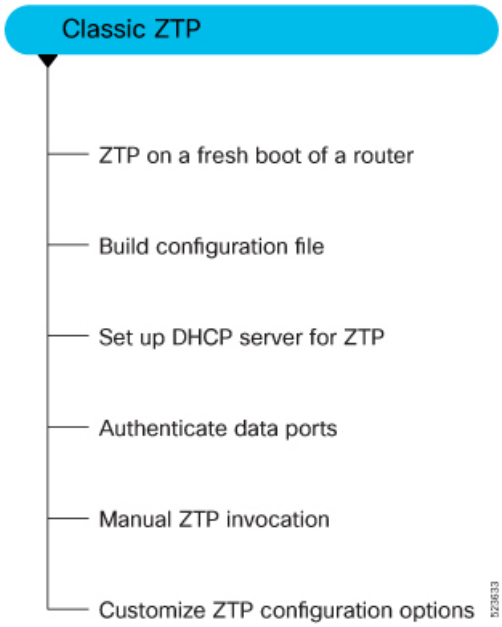
You can initiate ZTP in one of the following ways:

- **Fresh Boot**: Use this method for devices that has no pre-loaded configuration. See Getting Started with ZTP on a Fresh Boot of a Router. See Zero Touch Provisioning on a Fresh Boot of a Router, on page 95
- **Manual Invocation:** Use this method when you want to forcefully initiate ZTP on a fully configured device. See Invoke ZTP Manually, on page 98.

When to use Zero Touch Provisioning: Use Zero Touch Provisioning when the devices are in a secured network, but in an insecure network, we recommend you to Deploy Router Using Secure ZTP, on page 101.

The following figure lists the tasks to perform to configure classic ZTP.

Figure 6: Workflow to Configure Classic ZTP



- Build your configuration file, on page 83
- Authentication on Data Ports, on page 90
- Set Up DHCP Server, on page 91

- Customize ZTP Initialization File, on page 94
- Zero Touch Provisioning on a Fresh Boot of a Router, on page 95
- Invoke ZTP Manually, on page 98

Build your configuration file

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

Note When you use a USB flash drive as a source for ZTP, you cannot use the script file for provisioning. The script file is not supported in the USB fetcher. Fetcher defines which port the ZTP process should use to get the provisioning details as defined in the `ztp.ini` file.

The configuration file content starts with *!! IOS XR* and the script file content starts with *#! /bin/bash, #! /bin/sh, #! /usr/bin/python*.

Once you create the configuration file, apply it to the device using the *ztp_helper* function *xrapply*.

The following is the sample configuration file:

```
!! IOS XR
username root
group root-lr
password 0 lablab
!

hostname ios
alias exec al show alarms brief system active

interface HundredGigE 0/0/0/24
ipv4 address 10.10.10.55 255.255.255.0
no shutdown
!
```

Create User Script

This script or binary is executed in the IOS-XR Bash shell and can be used to interact with IOS-XR CLI to configure, verify the configured state and even run exec commands based on the workflow that the operator chooses.

Build your ZTP script with either shell and python.

Note ZTP does not have its own Python implementation. Instead, ZTP uses the default Python version and libraries available on the device. Python versions and supported libraries vary across Cisco IOS XR software versions and platforms (PID). Make sure that your script works correctly on the specific platform (PID) and required image version before using it in ZTP.

ZTP includes a set of CLI commands and a set of shell utilities that can be used within the user script.

Note We recommend that you do not execute the APIs on a router that is already provisioned. ZTP Utility APIs are designed to be executed from the ZTP script when you boot the router for the first time. The APIs perform additional operations to run the requested actions during the boot process and bring changes in the existing configuration before executing any action.

ZTP utility APIs have prerequisites that are executed in the ZTP workflow before running the ZTP utility APIs. These prerequisites help with running specific actions during the boot process and in making necessary configuration changes.

We recommend that you do not use ZTP utilities outside the scope of ZTP script. The APIs in this script use username as `ztp` or `ztp-user` in every action. The ZTP utility executed outside the scope of the ZTP script may fail as it is not executed from the ZTP workflow. This may modify the configurations on the device and affect other related operations. If the ZTP utility is executed outside the scope ZTP script, the logs display that the script is executed using username `ztp` or `ztp-user`, misleading that the script is executed from the workflow.

ZTP Shell Utilities

ZTP includes a set of shell utilities that can be sourced within the user script. `ztp_helper.sh` is a shell script that can be sourced by the user script. `ztp_helper.sh` provides simple utilities to access some XR functionalities. You can invoke the following bash functions:

- **xrcmd**—Used to run a single XR exec command:`xrcmd "show running"`
- **xrapply**—Applies the block of configuration, specified in a file:

```
cat >/tmp/config <<%%
!! XR config example
hostname node1-mgmt-via-xrapply
%%
xrapply /tmp/config
```

- **xrapply_with_reason**—Used to apply a block of XR configuration along with a reason for logging purpose:

```
cat >/tmp/config <<%%
!! XR config example
hostname node1-mgmt-via-xrapply
%%
xrapply_with_reason "this is a system upgrade" /tmp/config
```

- **xrapply_string**—Used to apply a block of XR configuration in one line:

```
xrapply_string "hostname foo\interface HundredGigE0/0/0/24\nipv4 address 1.2.3.44
255.255.255.0\n"
```

- **xrapply_string_with_reason**—Used to apply a block of XR configuration in one line along with a reason for logging purposes:

```
xrapply_string_with_reason ”system renamed again" "hostname venus\n interface
HundredGigE0/0/0/24\n
ipv4 address 172.30.0.144/24\n”
```

- **xrreplace**—Used to apply XR configuration replace in XR namespace via a file.

```
cat rtr.cfg <<%%
!! XR config example
hostname node1-mgmt-via-xrreplace
%%
xrreplace rtr.cfg
```

- **xrapply_with_extra_auth**—Used to apply XR configuration that requires authentication in XR namespace via a file. The **xrapply_with_extra_auth** API is used when configurations that require additional authentication to be applied such as alias, flex groups. This api internally performs authentication and authorization to gain additional privilege.

```
cat >/tmp/config <<%%
!! XR config example
alias exec alarms show alarms brief system active
alias exec version run cat /etc/show_version.txt
%%
xrapply_with_extra_auth >/tmp/config
```

- **xrreplace_with_extra_auth**—Used to apply XR configuration replace in XR namespace via a file. The **xrreplace_with_extra_auth** API is used when configurations that require additional authentication to be applied such as alias, flex groups. This api internally performs authentication and authorization to gain additional privilege.

```
cat >/tmp/config <<%%
!! XR config example
alias exec alarms show alarms brief system active
alias exec version run cat /etc/show_version.txt
%%
xrreplace_with_extra_auth >/tmp/config
```

API Implementation Behavior

Note The **xrcmd**,**xrapply**, and **xrreplace** APIs or utilities carry out a series of internal operations to execute specific actions. These operations, which are performed sequentially, include:

- User Creation—This operation involves generating a `ztp-user` (temporary user) before the execution of any other operations.
- Command Execution or Configuration Application—This operation encompasses executing a command, applying configurations using parser utilities, or applying the configuration through cfg-mgr.
- User Removal—This operation involves removing the `ztp-user` (temporary user) from the XR configuration.

In addition to these internal operations, the **xrapply_with_extra_auth** and **xrreplace_with_extra_auth** APIs performs an authentication process before applying configurations.

ZTP Helper Python Library

The ZTP python library defines a single Python class called `ZtpHelpers`. The helper script is located at `/pkg/bin/ztp_helper.sh`

ZtpHelpers Class Methods

Following are utility methods of the `ZtpHelpers` class:

- `init(self, syslog_server=None, syslog_port=None, syslog_file=None):`

```
__init__ constructor
:param syslog_server: IP address of reachable Syslog Server
       :param syslog_port: Port for the reachable syslog server
        :param syslog_file: Alternative or addon file for syslog
        :type syslog_server: str
        :type syslog_port: int
        :type syslog_file:str

         All parameters are optional. When nothing is specified during object creation,
 then all logs are sent to a log rotated file /tmp/ztp_python.log (max size of 1MB).
```

- `setns(cls, fd, nstype):`

```
Class Method for setting the network namespace
           :param cls: Reference to the class ZtpHelpers
           :param fd: incoming file descriptor
           :param nstype: namespace type for the sentns call
           :type nstype: int
                  0      Allow any type of namespace to be joined.
                  CLONE_NEWNET = 0x40000000 (since Linux 3.0)
                         fd must refer to a network namespace
```

- `get_netns_path(cls, nspath=None, nsname=None, nspid=None):`

```
Class Method to fetch the network namespace filepath
           associated with a PID or name
           :param cls: Reference to the class ZtpHelpers
           :param nspath: optional network namespace associated name
           :param nspid: optional network namespace associate PID
           :type nspath: str
           :type nspid: int
           :return: Return the complete file path
           :rtype:  str
```

- `toggle_debug(self, enable):`

```
Enable/disable debug logging
           :param enable: Enable/Disable flag
           :type enable: int
```

- `set_vrf(self, vrfname=None):`

```
Set the VRF (network namespace)
           :param vrfname: Network namespace name
                           corresponding to XR VRF
```

- `download_file(self, file_url, destination_folder):`

```
Download a file from the specified URL
           :param file_url: Complete URL to download file
           :param destination_folder: Folder to store the
                                      downloaded file
           :type file_url: str
           :type destination_folder: str
           :return: Dictionary specifying download success/failure
                    Failure => { 'status' : 'error' }
                    Success => { 'status' : 'success',
                                 'filename' : 'Name of downloaded file',
                                 'folder' : 'Directory location of downloaded file'}
           :rtype: dict
```

- `setup_syslog(self):`

  Method to Correctly set sysloghandler in the correct VRF (network namespace) and point to a remote syslog Server or local file or default log-rotated log file.

- `xrcmd(self, cmd=None):`

```
Issue an IOS-XR exec command and obtain the output
           :param cmd: Dictionary representing the XR exec cmd
                       and response to potential prompts
                       { 'exec_cmd': '', 'prompt_response': '' }
           :type cmd: dict
           :return: Return a dictionary with status and output
                    { 'status': 'error/success', 'output': '' }
           :rtype: dict
```

- `xrapply(self, filename=None, reason=None):`

```
Apply Configuration to XR using a file
           :param file: Filepath for a config file
                        with the following structure:
                        !
                        XR config command
                        !
                        end

           :param reason: Reason for the config commit.
                          Will show up in the output of:
                          "show configuration commit list detail"
           :type filename: str
           :type reason: str
           :return: Dictionary specifying the effect of the config change
                    { 'status' : 'error/success', 'output': 'exec command based on
status'}
                    In case of Error:  'output' = 'show configuration failed'
                    In case of Success: 'output' = 'show configuration commit changes
 last 1'
           :rtype: dict
```

- `xrapply_string(self, cmd=None, reason=None):`

```
Apply Configuration to XR using  a single line string
           :param cmd: Single line string representing an XR config command
           :param reason: Reason for the config commit.
                          Will show up in the output of:
                          "show configuration commit list detail"
           :type cmd: str
           :type reason: str
           :return: Dictionary specifying the effect of the config change
                    { 'status' : 'error/success', 'output': 'exec command based on
status'}
                    In case of Error:  'output' = 'show configuration failed'
                    In case of Success: 'output' = 'show configuration commit changes
 last 1'
           :rtype: dict
```

- `xrreplace(self, filename=None):`

```
Replace XR Configuration using a file

           :param file: Filepath for a config file
                        with the following structure:

                        !
                        XR config commands
```

```
                        !
                        end
            :type filename: str
            :return: Dictionary specifying the effect of the config change
                     { 'status' : 'error/success', 'output': 'exec command based on
status'}
                     In case of Error:  'output' = 'show configuration failed'
                     In case of Success: 'output' = 'show configuration commit changes
 last 1'
            :rtype: dict
```

API Implementation Behavior

Note The **xrcmd**,**xrapply**, and **xrreplace** APIs or utilities carry out a series of internal operations to execute specific actions. These operations, which are performed sequentially, include:

- User Creation—This operation involves generating a `ztp-user` (temporary user) before the execution of any other operations.
- Command Execution or Configuration Application—This operation encompasses executing a command, applying configurations using parser utilities, or applying the configuration through cfg-mgr.
- User Removal—This operation involves removing the `ztp-user` (temporary user) from the XR configuration.

Example

The following shows the sample script in phyton

```
[testenv]$ python sample_ztp_script.py

###### Debugs enabled ######


###### Change context to user specified VRF ######


###### Using Child class method, setting the root user ######

2016-12-17 04:23:24,091 - DebugZTPLogger - DEBUG - Config File content to be applied  !
                    username netops
                    group root-lr
                    group cisco-support
                    secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1
                    !
                    end
2016-12-17 04:23:28,546 - DebugZTPLogger - DEBUG - Received exec command request: "show
configuration commit changes last 1"
2016-12-17 04:23:28,546 - DebugZTPLogger - DEBUG - Response to any expected prompt ""
Building configuration...
2016-12-17 04:23:29,329 - DebugZTPLogger - DEBUG - Exec command output is ['!! IOS XR
Configuration version = 6.2.1.21I', 'username netops', 'group root-lr', 'group cisco-support',
 'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1', '!', 'end']
2016-12-17 04:23:29,330 - DebugZTPLogger - DEBUG - Config apply through file successful,
last change = ['!! IOS XR Configuration version = 6.2.1.21I', 'username netops', 'group
root-lr', 'group cisco-support', 'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1', '!', 'end']
```

```
###### Debugs Disabled ######


###### Executing a show command ######

Building configuration...
{'output': ['!! IOS XR Configuration version = 6.2.1.21I',
            '!! Last configuration change at Sat Dec 17 04:23:25 2016 by UNKNOWN',
            '!',
            'hostname customer2',
            'username root',
            'group root-lr',
            'group cisco-support',
            'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1',
            '!',
            'username noc',
            'group root-lr',
            'group cisco-support',
            'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1',
            '!',
            'username netops',
            'group root-lr',
            'group cisco-support',
            'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1',
            '!',
            'username netops2',
            'group root-lr',
            'group cisco-support',
            'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1',
            '!',
            'username netops3',
            'group root-lr',
            'group cisco-support',
            'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1',
            '!',
            'cdp',
            'service cli interactive disable',
            'interface MgmtEth0/RP0/CPU0/0',
            'ipv4 address 11.11.11.59 255.255.255.0',
            '!',
            'interface TenGigE0/0/0/0/24',
            'shutdown',
            '!',
            'interface TenGigE0/0/0/0/25',
            'shutdown',
            '!',

            'router static',
            'address-family ipv4 unicast',
            '0.0.0.0/0 11.11.11.2',
            '!',
            '!',
            'end'],
 'status': 'success'}

###### Apply valid configuration using a file ######

Building configuration...
{'status': 'success', 'output': ['!! IOS XR Configuration version = 6.2.1.21I', 'hostname
customer', 'cdp', 'end']}

###### Apply valid configuration using a string ######

Building configuration...
```

```
{'output': ['!! IOS XR Configuration version = 6.2.1.21I',
            'hostname customer2',
            'end'],
 'status': 'success'}

###### Apply invalid configuration using a string ######

{'output': ['!! SYNTAX/AUTHORIZATION ERRORS: This configuration failed due to',
            '!! one or more of the following reasons:',
            '!!  - the entered commands do not exist,',
            '!!  - the entered commands have errors in their syntax,',
            '!!  - the software packages containing the commands are not active,',
```

For information on helper APIs, see https://github.com/ios-xr/iosxr-ztp-python#iosxr-ztp-python.

Authentication on 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 'xr-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 routers 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  "xr-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-XR-CONFIG code width 2 length width 2;
option CISCO-XR-CONFIG.client-identifier code 1 = string;
option CISCO-XR-CONFIG.authCode code 2 = integer 8;
option CISCO-XR-CONFIG.md5sum code 3 = string;
option vsio.CISCO-XR-CONFIG code 9 = encapsulate CISCO-XR-CONFIG;
subnet6 2001:1451:c632:1::/64{
 range6 2001:1451:c632:1::2 2001:1451:c632:1::9;
 option CISCO-XR-CONFIG.client-identifier "xr-config";
 option CISCO-XR-CONFIG.authCode 1;
 #valid md5
 option CISCO-XR-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";
  }
}
```

Set Up 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 router 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 ("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 =  "xr-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 15: If you have configured this option for the server to identify ztp requests, ensure that you update the server configuration, for Linux or ISC servers. Sample server-side configuration required to check user-class for ZTP is shown in the following example:

  ```
  if exists dhcp6.user-class and (substring(option dhcp6.user-class, 0, 9) = "xr-config"
   or substring(option dhcp6.user-class, 2, 9) = "xr-config"){
       #
  }
  ```

- 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 11.11.11.255;
```

```
 option routers 11.11.11.2;
 option domain-name-servers 11.11.11.2;
 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 11.11.11.47 11.11.11.50;
      next-server 11.11.11.2;

      if exists user-class and option user-class = "iPXE" {
         filename="http://11.11.11.2:9090/cisco-mini-x-6.2.25.10I.iso";
      }


      if exists user-class and option user-class =  "xr-config"
       {
          if (substring(option vendor-class.cisco-vendor-id-vendor-class,19,99)="cisco")
          {
           option bootfile-name "http://11.11.11.2:9090/scripts/exhaustive_ztp_script.py";

          }
        }


      ddns-hostname "cisco-local";
      option routers 11.11.11.2;
  }
}
```

Important In Cisco IOS XR Release 7.3.1 and earlier, the system accepts the device sending **user-class = "exr-config"**; however starting Cisco IOS XR Release 7.3.2 and later, you must use only **user-class = "xr-config"**.

In Cisco IOS XR Release 7.3.2 and later, use:

```
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 = "xr-config" {
      filename = "http://172.30.0.22/scripts/cisco-rp0_ztp.sh";
   }
}
```

Also, when upgrading from any release that is Cisco IOS XR Release 7.3.1 or earlier to Cisco IOS XR Release 7.3.2 or later release, use the following:

```
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";
   }
}
```

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 0–10.

Note Lower the number higher the priority. The value 0 has the highest priority and 10 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
```

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

```
Router#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

```
Router#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.
```

Zero Touch Provisioning on a Fresh Boot of a Router

When you boot the device, the ZTP process initiates automatically if the device does not have a prior configuration.

During the process, the router verifies the list of sources and receives the information of the configuration file accordingly. The following are the sources that can provide the configuration file information.

- Removable storage: A directly attached removable storage device, for example, USB flash drive.
- DHCP server

Fresh Boot Using Removable Storage Device

You can automatically provision a network device using ZTP from a removable storage device such as a USB flash drive. The following are the configuration types available in a removable storage device:

- Device-specific configuration: The device-specific configuration is available in the folder that has a name matching the chassis serial number of the device. The sample path for the device-specific configuration is `/USB-path/xr-config/serial-number/router-cfg`. For example, `/USB-path/xr-config/FOC2102R1D0/router-cfg` `FOC2102R1D0` is the chassis serial number.
- Generic configuration: The generic configuration is available in the `xr-config` folder. The sample path for the generic configuration is./`USB-path/xr-config/router-cfg`

Here is the high-level work flow of the ZTP process using a USB flash drive:

1. When you boot the device, the device verifies if the USB is enabled in the `ztp.ini` file. By default, the USB fetcher is enabled and assigned the highest 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.

2. ZTP checks for a USB flash drive on the device If the USB drive isn't available, the ZTP process moves to the next fetcher as defined in the fetcher priority of the `ZTP.ini` file.
3. If a USB flash drive is available, the device scans for the `xr-config` file in the root of the USB mount in the following sequence:
   a. The ZTP process first scans for the `router-cfg` file in the folder that is matching the chassis serial number of the device within the `xr-config` folder and applies the device-specific configuration.

      For example, `/USB-path/xr-config/FOC2102R1D0/router-cfg`

   b. If the device-specific configuration with a serial number isn't available, the ZTP process scans for the `router-cfg` file in the `xr-config` folder and applies a generic configuration.

   c. If the `xr-config` folder isn't available, the ZTP process moves to the next fetcher as defined in the fetcher priority of the `ZTP.ini` file.
4. The device applies the configuration.
5. The network device is now up and running.

Configure ZTP using USB

Follow these steps to configure ZTP using a USB flash drive:

1. Create the configuration file. See Build your configuration file, on page 83.

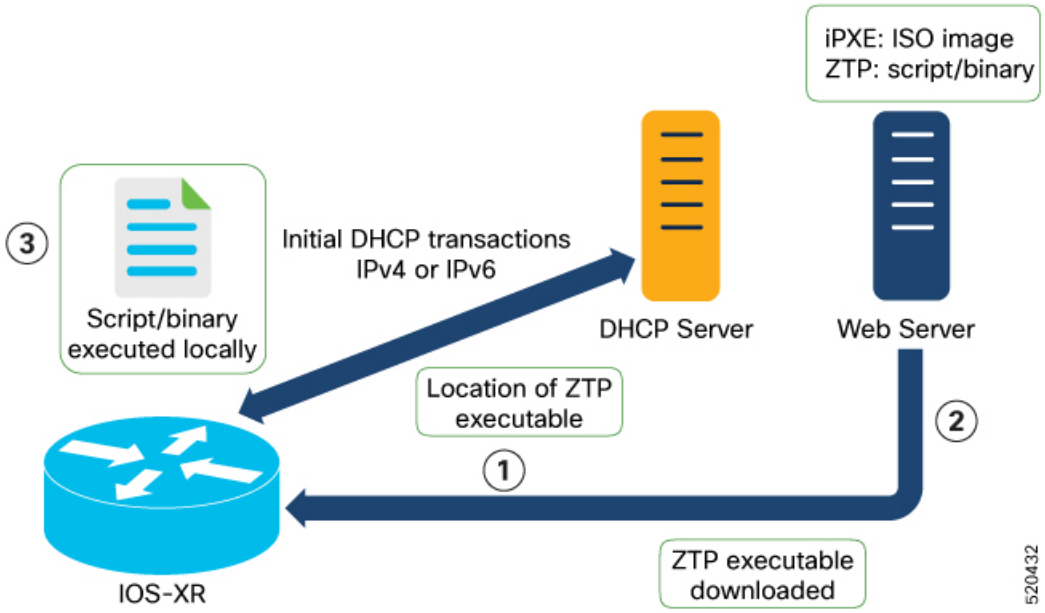
Note When you use a USB flash drive as a source for ZTP, you can't use the script file for provisioning. The script file isn't supported for USB fetcher.

2. Copy the bootstrapping data to the USB flash drive and mount it on the device.

Fresh Boot Using DHCP

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

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



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

Here is the high-level work flow of the ZTP process for the Fresh boot:

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

   The following is the default sequential flow of the ZTP process:
   - ZTP sends IPv4 DHCP request first on all the management port. In case there is a failure, then ZTP sends IPv6 DHCP request on all the management port.

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

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

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

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

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

3. The network device downloads the file from the web server using the URI location that is provided in the DHCP response.
4. The device receives a configuration file or script file from the HTTP server.

Note

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

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

Invoke ZTP Manually

You can invoke Zero Touch Provisioning (ZTP) manually through the Command Line Interface. This method is Ideal for verifying the ZTP configuration without a reboot. This manual approach helps you to provision the router in stages. To invoke ZTP on an interface (data ports or management port), you don't have to bring up and configure the interface first.

Even when the interface is down, you can run the `ztp initiate` command, and the ZTP script will bring it up and invoke `dhclient`. Hence, ZTP can run on all interfaces irrespective of their availability.

Use the following commands to manually invoke the ZTP commands and to force ZTP to run on all interfaces:

- **ztp initiate** — Invokes a new ZTP DHCP session. Logs can be found in **/disk0:/ztp/ztp.log**.

  Configuration Example:

```
Router#ztp initiate debug verbose interface HundredGigE 0/0/0/24
Invoke ZTP? (this may change your configuration) [confirm] [y/n] :
```

- **ztp terminate** —Terminates any ZTP session in progress.

  Configuration Example:

```
Router #ztp terminate verbose
Mon Oct 10 16:52:38.507 UTC
Terminate ZTP? (this may leave your system in a partially configured state) [confirm]
[y/n] :y
ZTP terminated
```

- **ztp enable** —Enables the ZTP at boot.

  Configuration Example:

  ```
  Router#ztp enable
  Fri Jul 12 16:09:02.154 UTC
  Enable ZTP? [confirm] [y/n] :y
  ZTP Enabled.
  ```

- **ztp disable** —Disables the ZTP at boot.

  Configuration Example:

  ```
  Router#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.
  ```

- **ztp clean** —Removes only the ZTP state files.

  Configuration Example:

  ```
  Router#ztp clean verbose
  Mon Oct 10 17:03:43.581 UTC
  Remove all ZTP temporary files and logs? [confirm] [y/n] :y
  All ZTP files have been removed.
  If you now wish ZTP to run again from boot, do 'conf t/commit replace' followed by
  reload.
  ```

  The log file ztp.log is saved in **/var/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/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/** folder.

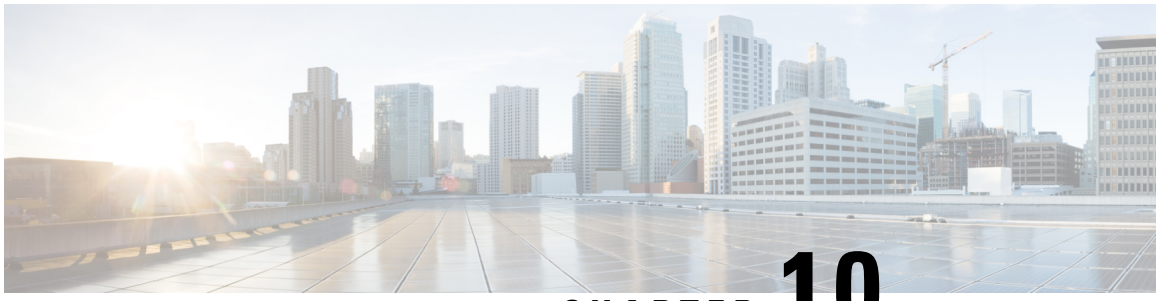
Configuration

This task shows the most common use case of manual ZTP invocation: invoke ZTP.

1. Invoke DHCP sessions on all data ports which are up or could be brought up. ZTP runs in the background. Use show logging or look at /disk0:/ztp/ztp.log to check progress.

Configuration Example:

```
Router# ztp initiate dataport
```



CHAPTER 10

Deploy Router Using Secure ZTP

With Secure Zero Touch Provisioning, you can securely and seamlessly provision thousands of network devices accurately within minutes and without any manual intervention.

Table 6: Feature History Table

Feature	Release Information	Feature Description
Secure Zero Touch Provisioning with Removable Storage Device	Release 7.3.2	This feature allows you to securely sign onboarding data in a removable storage device so that you can use the device for secure ZTP operations. This support gives you the plug-and-play flexibility for ZTP without any additional infrastructure requirements.

In a secured network such as datacenter, the zero-touch provisioning mechanism helps you provision hundreds of remote devices without your intervention. But, the access devices are typically in an insecure network. There is a high risk of malicious actions on the device, such as adding an unauthorized or infected device. Security is a critical aspect while remotely provisioning the network devices.

Secure ZTP combines seamless automation with security. Network devices can securely establish a connection with the ZTP server and authenticate the onboarding information that it receives. The process eliminates any security risks or malicious actions during the provisioning of remote devices.

- ZTP helps you remotely provision a router securely anywhere in the network. Thus, eliminate the risk of malicious attacks or unauthorized ownership claims.
- Secure ZTP authenticates not only the onboarding network device but also validates the server authenticity and provisioning information that it is receiving from the ZTP server.

The following are the topics covered in this chapter:

- On board Devices Using Three-Step Validation, on page 102
- Secure ZTP Components , on page 102
- Secure Zero Touch Provisioning, on page 109
- Upgrade Image Using Secure ZTP, on page 118
- Disable Secure ZTP , on page 121

On board Devices Using Three-Step Validation

The Cisco IOS XR software implements the secure zero touch provisioning capabilities as described in RFC 8572. Secure ZTP uses a three-step validation process to on board the remote devices securely:

1. **Router Validation**: The ZTP server authenticates the router before providing bootstrapping data using the Trust Anchor Certificate (SUDI certificate). Ensure that you have preinstalled the CA certificate chain for Cisco, as this is a prerequisite for the Cisco CA on ZTP server to verify the client/router SUDI certificates. The required certificates are:
   - subject=O = Cisco, CN = ACT2 SUDI CA
   - subject=O = Cisco Systems, CN = Cisco Root CA 2048
   - subject=CN = High Assurance SUDI CA, O = Cisco
   - subject=O = Cisco, CN = Cisco Root CA 2099
2. **Server Validation**: The router device in turn validates the ZTP server to make sure that the on board happens to the correct network. Upon completion, the ZTP server sends the bootstrapping data (for example, a YANG data model) or artifact to the router. See Secure ZTP Components , on page 102.
3. **Artifact Validation**: The configuration validates the bootstrapping data or artifact that is received from the ZTP server.

Secure ZTP Components

Let's first understand the components required for secure ZTP.

Table 7: Components used in Secure ZTP

Components	Description
Onboarding Device (Router)	The router is a Cisco device that you want to provision and connect to your network. Secure ZTP is supported only on platforms that have Hardware TAM support. Routers with HW TAM have the SUDI embedded in TAM.
DHCP Server	The secure ZTP process relies on the DHCP server to provide the URL to access the bootstrapping information.

Components	Description

ZTP Server	A ZTP server is any server used as a source of secure ZTP bootstrapping data and can be a RESTCONF or HTTPs server.
	Note Currently, ZTP only supports single name-server. When the DHCP server has more than one server address configured, ZTP fails to apply the server configuration.
	The ZTP server contains the following artifacts:
	• Cisco IOS XR software images: You can download Cisco images, SMU, and patches using the Cisco Support & Downloads page.
	• ZTP scripts: Contains the following libraries and you can build a script to initiate the ZTP process. See Build your configuration file, on page 83.
	  • Python library: Includes IOS XR CLI (show commands and configuration commands) and YANG-XML (ncclient, native Netconf client).
	  • BASH library: Includes IOS XR CLI show commands, configuration commands
	• Bootstrapping Data

Components	Description
Bootstrapping Data	

Components	Description

	Bootstrapping data is the collection of data that the router obtains from the ZTP server during the secure ZTP process. You must create and upload the bootstrapping data in the ZTP server. For more information, refer RFC 8572.
	• The bootstrapping data mainly has three artifacts:
	• **Conveyed Information**: Conveyed Information contains the required bootstrapping data for the device. It contains either the redirect information or onboarding information to provision the device.
	For example:
	`module: ietf-sztp-conveyed-info`
	`yang-data conveyed-information: +-- (information-type)`
	`+--:(redirect-information) | +-- redirect-information | +-- bootstrap-server* [address] | +-- address inet:host | +-- port? inet:port-number | +-- trust-anchor? cms`
	`+--:(onboarding-information) +-- onboarding-information +-- boot-image | +-- os-name? string | +-- os-version? string | +-- download-uri* inet:uri | +-- image-verification* [hash-algorithm] | +-- hash-algorithm identityref | +-- hash-value yang:hex-string +-- configuration-handling? enumeration +-- pre-configuration-script? script +-- configuration? binary +-- post-configuration-script? script`

Components	Description
	• **Redirect Information**: Redirect information is used to redirect a device to another bootstrap server. The redirect information contains a list of bootstrap servers along with a hostname, an optional port, and an optional trust anchor certificate that the device uses to authenticate the bootstrap server. For Example: `{ "ietf-sztp-conveyed-info:redirect-information" : { "bootstrap-server" : [{ "address" : "sztp1.example.com", "port" : 8443, "trust-anchor" : "base64encodedvalue==" }, { "address" : "sztp2.example.com", "port" : 8443, "trust-anchor" : "base64encodedvalue==" }, { "address" : "sztp3.example.com", "port" : 8443, "trust-anchor" : "base64encodedvalue==" }] } }`

Components	Description
	• **Onboarding Information:** Onboarding information provides data necessary for a device to bootstrap itself and establish secure connections with other systems. It specifies details about the boot image, an initial configuration the device must commit, and scripts that the device must execute. For Example: `{ "ietf-sztp-conveyed-info:onboarding-information" : { "boot-image" : { "os-name" : "VendorOS", "os-version" : "17.2R1.6", "download-uri" : ["https://example.com/path/to/image/file"], "image-verification" : [{ "hash-algorithm" : "ietf-sztp-conveyed-info:sha-256", "hash-value" : "ba:ec:cf:a5:67:82:b4:10:77:c6:67:a6:22:ab:\ 7d:50:04:a7:8b:8f:0e:db:02:8b:f4:75:55:fb:c1:13:b2:33" }] }, "configuration-handling" : "merge", "pre-configuration-script" : "base64encodedvalue==", "configuration" : "base64encodedvalue==", "post-configuration-script" : "base64encodedvalue==" } }`

Components	Description
	• **Owner Certificate**: The owner certificate is installed on the router with the public key of your organization. The router uses the owner certificate to verify the signature in the conveyed information artifact using the public key that is available in the owner certificate. • **Ownership Voucher**: Ownership Voucher is used to identify the owner of the device by verifying the owner certificate that is stored in the device. Cisco supplies Ownership Voucher in response to your request. You must submit the Pinned Domain Certificate and device serial numbers with the request. Cisco generates and provides the Ownership Voucher to you.

Components	Description

Report Progress	When the device obtains the onboarding information from a ZTP server, the router reports the bootstrapping progress to the ZTP server using the API calls.
	See RFC 8572 for the detailed report-progress messages that can be sent to the ZTP server.
	The following is the structure of the `report-progress` sent the progress message to a ZTP server.
	`+---x report-progress {onboarding-server}?`
	`+---w input`
	`+---w progress-type enumeration`
	`+---w message? string`
	`+---w ssh-host-keys`
	`+---w ssh-host-key* []`
	`+---w algorithm string`
	`+---w key-data binary`
	`+---w trust-anchor-certs`
	`+---w trust-anchor-cert* cms`
	The following example illustrates a device using the Yang module to post a progress report to a ZTP server with a `bootstrap complete` message:
	`{ 'progress-type': 'bootstrap-complete', 'message': 'example message', 'trust-anchor-certs': [{ 'trust-anchor-cert': 'base64encodedvalue==' 'ssh-host-keys': [{ 'key-data': 'base64encodedvalue==', 'algorithm': 'ssh-rsa' }, { 'key-data': 'base64encodedvalue==', 'algorithm': 'rsa-sha2-256' }] }`
	`RESPONSE from the ZTP server`
	`HTTP/1.1 204 No Content`
	`Date: Sat, 31 Oct 2015 17:02:40 GMT`
	`Server: example-server`

Secure Zero Touch Provisioning

When you boot the device, the secure ZTP process initiates automatically if the device does not have a prior configuration.

During the process, the router verifies the list of sources and receives the information of the configuration file accordingly. The following are the sources that can provide the configuration file information.

- Removable storage: A directly attached removable storage device, for example, USB flash drive.
- DHCP server

The section covers the following topics:

Secure ZTP with Removable Storage Device

A Removable storage device such as a USB drive is an untrusted source of bootstrapping data. So, the onboarding information present in the removable storage device must always be signed.

Whenever the data is signed, it's mandatory that the Owner Certificate and Ownership Voucher must also be available. The removable storage device must contain the following three artifacts. For more information on the three artifacts, see Secure ZTP Components , on page 102.

- Conveyed Information
- Owner Certificate
- Ownership Voucher

This section covers the following topics:

Prepare Removable Storage Device to Provision Secure ZTP

The network administrator performs the following tasks as part of the initial setup for secure ZTP:

Before performing the following tasks, ensure to enable secure ZTP on the router using the **ztp secure-mode enable** command and then reload the router.

1. Contact Cisco Support to obtain a voucher. Provide the following details to request for ownership voucher certificate:
   - Pinned Domain certificate (PDC): PDC is an X.509 v3 certificate structure that uses Distinguished Encoding Rules (DER). This certificate is used by the router to trust a public key infrastructure in order to verify a domain certificate supplied to the router separately in the bootstrapping data. This certificate could be an end-entity certificate, including a self signed entity.
   - Order details with the Serial numbers of the routers

   For example,

   ```
   {
          "expires-on": "2016-10-21T19:31:42Z",
          "assertion": "verified",
          "serial-number": "JADA123456789",
          "idevid-issuer": "base64encodedvalue==",
          "pinned-domain-cert": "base64endvalue==",
          "last-renewal-date": "2017-10-07T19:31:42Z"
      }
   ```

2. Copy the following data to the removable storage device in the **EN9** directory in its root:
   - Conveyed information: Conveyed information must be named as `conveyed-information.cms` and must contain only the onboarding information and not the redirect information.

- Onboarding Information: The conveyed information consists of the following onboarding information:
  - Cisco IOS XR software images: You can download Cisco images, SMU, and patches using the Cisco Support & Downloads page.
    a. Click **Routers** and select the product that you want to download the image for.
    b. On the product home page, select the required Product model from the **Downloads** tab.
    c. From the **Software Available For This Product** page, download the required Cisco image, SMU, and patches.
  - ZTP scripts that include IOS XR configurations, pre, and post configuration scripts.

    During the secure ZTP process, secure ZTP executes the scripts to provision the router. You can build your script using one of the following methods:
    a. Python library: Includes IOS XR CLI (show commands and configuration commands) and YANG-XML (`ncclient`, `native Netconf client`).
    b. BASH library: Includes IOS XR CLI show commands, configuration commands.

    See Build your configuration file, on page 83.

- Owner certificate: The owner certificate must be named as `owner-certificate.cms`.
- Ownership vouchers: The ownership vouchers must be named as `ownership-voucher.vcj`.

The artifacts must be stored inside the subdirectory named after the RP serial number of the router.The following example shows a directory structure for the router with RP serial number `FOC2202R293` containing all three artifacts:

```
EN9
└── FOC2202R293
    └── bootstrapping-data
        ├── conveyed-information.cms
        ├── owner-certificate.cms
        └── ownership-voucher.vcj
```

3. Plug in the removable storage device into the router.
4. Power on the router.

How Does Secure ZTP Work with Removable Storage Device?

Before you begin, complete the task to prepare the removable storage device. See Prepare Removable Storage Device to Provision Secure ZTP, on page 110.

Here is the high-level workflow of the Secure ZTP process using a removable storage device:

1. When you boot the device with an IOS-XR image, the secure ZTP process verifies if the secure ZTP mode (`secure-ztp mode`) is enabled. If not enabled, the device boots normally.
2. The device verifies if the USB is enabled in the `ztp.ini` file. By default, the USB is enabled and assigned the highest priority in the fetcher priority in the `ztp.ini` file.

Fetcher priority defines how secure ZTP can get the provisioning details. By default, each port has a fetcher priority defined in the `ztp.ini` file. The fetcher priority range is from 0 to 9. The lower the number higher is the priority. The value 0 has the highest priority and 9 has the lowest priority. For more information, see Customize ZTP Initialization File, on page 94.

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
```

3. Secure ZTP checks for a removable storage device on the router. If the removable storage device isn't available, the secure ZTP process moves to the next fetcher as defined in the fetcher priority of the `ztp.ini`file.
4. If a removable storage device is available, the router scans for the `EN9` directory in the root of the removable storage device.

   If the `EN9` directory isn't available, the secure ZTP process moves to the next fetcher as defined in the fetcher priority of the `ztp.ini`file.
5. **Artifact Validation**:

   The router validates the artifacts received from the removable storage device.

   a. The router validates the ownership voucher and extracts the `pinned-domain-cert` node, an X.509 certificate from the ownership voucher to verify the owner certificate.

   b. The router authenticates the owner certificate by performing the X.509 certificate path verification process on the trusted certificate.

   c. Finally, the router verifies whether the conveyed information artifact is signed by the validated owner certificate.
6. **Provision the router**:

   a. The device first processes the boot image information.

   b. Executes the preconfiguration script and then commits the initial configuration.

   c. Execute the post configuration script.
7. After the onboarding process is completed, router is operational.

Note If there is a failure in any of the steps, the secure ZTP process moves to the next fetcher as defined in the fetcher priority of the `ztp.ini` file.

Secure ZTP with DHCP

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

This section covers the following topics:

Initial Set Up for Secure ZTP

The network administrator performs the following tasks as part of the initial setup for secure ZTP:

1. Contact Cisco Support to obtain a voucher. Provide the following details to request for ownership voucher certificate:
   - Pinned Domain Certificate: A trusted digital certificate issued by the Certificate Authority (CA) and pinned by the operator.
   - Order details with the Serial numbers of the routers
   - For example,

```
{
        "expires-on": "2016-10-21T19:31:42Z",
        "assertion": "verified",
        "serial-number": "JADA123456789",
        "idevid-issuer": "base64encodedvalue==",
        "pinned-domain-cert": "base64endvalue==",
        "last-renewal-date": "2017-10-07T19:31:42Z"
    }
```

2. Upload the following bootstrapping data to the ZTP server. Steps to upload may vary depending on the server that you're using, refer to the documentation provided by your vendor.
   - Cisco IOS XR software images: You can download Cisco images, SMU, and patches using the Cisco Support & Downloads page.
   - ZTP scripts that include IOS XR configurations, pre, and post configuration scripts. Build a script to initiate the ZTP process. See Build your configuration file, on page 83.
     - Python library: Includes IOS XR CLI (show commands and configuration commands) and YANG-XML (`ncclient`, `native Netconf client`).
     - BASH library: Includes IOS XR CLI show commands, configuration commands
   - Serial numbers of the routers you plan to onboard using ZTP
   - Owner certificates
   - Pinned Domain Certificate (PDC)
   - Ownership vouchers
3. Set up the DHCP server to provide the redirect URL to the router:

   Before triggering the secure ZTP process, configure the DHCP server to provide the location of the IOS-XR image to the router. For information on how to configure the DHCP server, see your DHCP server documentation.

Configure the following parameters in the DHCP server:

- `option-code`: The DHCP SZTP redirect Option has the following parameters:
  - OPTION_V4_SZTP_REDIRECT (143): Use this DHCP v4 code for IPV4.
  - OPTION_V6_SZTP_REDIRECT (136): Use this DHCP v4 code for IPV6.

  For example, `option dhcp6.bootstrap-servers code 136 = text;`
- `option-length`: The option length in octets
- `bootstrap-servers`: A list of servers for the onboarding device to contact the servers for the bootstrapping data.
- `bootfile-url` : The URI of the SZTP bootstrap server should use the HTTPS URI scheme and it should be in the following format:

  `"https://<ip-address-or-hostname>[:<port>]".`

4. Power on the router.
5. Enable the secure ZTP option on the onboarding device. Execute the following command on your router to enable secure ZTP:

```
Router# ztp secure-mode enable
```

How Does Secure ZTP Work?

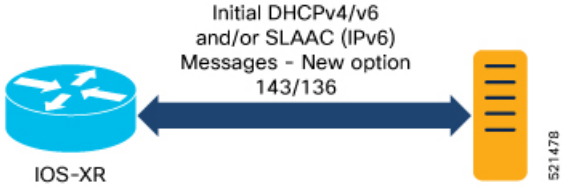
Before you begin, ensure that you configure the network with the DHCP and ZTP server. See Initial Set Up for Secure ZTP, on page 113.

1. When you boot the device with an IOS-XR image, the secure ZTP process verifies if the secure ZTP mode (`secure-ztp mode`) is enabled. If not enabled, the device boots normally.

Note When `secure-ztp mode` is enabled, the ZTP process accepts only the `secure-redirect-URL` and ignores the presence of boot file name option from the DHCP response.

2. **DHCP discovery**:
   a. The router initiates a DHCP request to the DHCP server.
   b. The DHCP server responds with a DHCPv4 143 address option (for IPv4 addressing) or a DHCPv6 136 option (for IPv6 addressing). In addition, URLs to access bootstrap servers for further configuration is also listed.

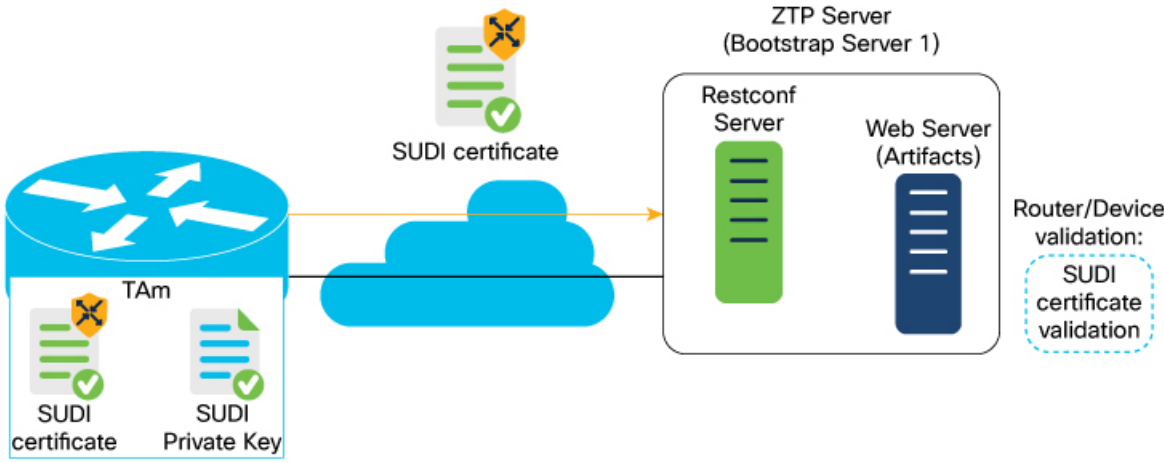
Figure 7: DHCP discovery



3. **Router validation**:

   a. After receiving the URL from the DHCP server, the router sends an HTTPs request to the RESTCONF or HTTPs server using the specified URL. Along with the HTTPs request, the device sends the client certificate that is provided by the manufacturer (also called SUDI certificate). This certificate identifies and authenticates itself to the ZTP server.

   Figure 8: Router Validation



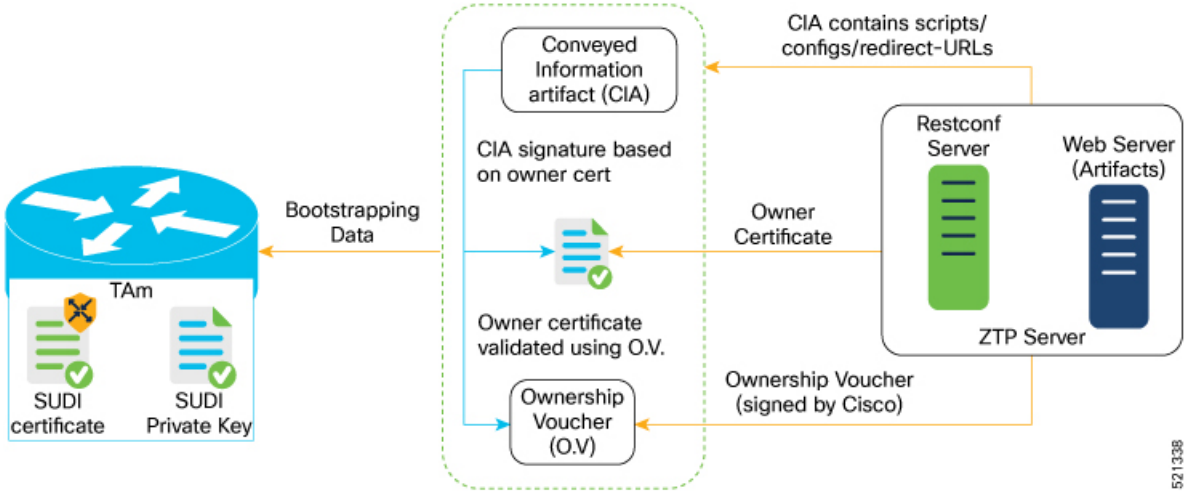
   b. The RESTCONF or HTTPs server verifies the received SUDI certificate with the public certificate that it contains. Cisco issues the public certificate to ensure that the onboarding device is an authorized Cisco device.

   c. After the onboarding device is authenticated, the web server sends the required artifacts along with the secure ZTP yang model to the onboarding device.

4. **Server validation** :

   The router receives the yang model that contains Owner Certificate, Ownership Voucher, and Conveyed Information artifact. The router verifies the ownership voucher by validating its signature to one of its preconfigured trusts anchors and downloads the image. When the router obtains the onboarding information, it reports the bootstrapping progress to the ZTP server. See RFC 8572 for the progress information.

Figure 9: Server Validation



5. **Artifact Validation**:

   The router validates the artifact received from the ZTP server.

   a. The device extracts the `pinned-domain-cert` node, an X.509 certificate from the ownership voucher to verify the owner certificate.

   b. The device authenticates the owner certificate by performing the X.509 certificate path verification process on the trusted certificate.

   c. Finally, the device verifies whether the conveyed information artifact is signed by the validated owner certificate.

6. **Provision the device**:

   a. The device first processes the boot image information.

   b. Executes the pre-configuration script and then commits the initial configuration

   c. Execute the post configuration script.

7. After the onboarding process is completed, the network device is operational.

The following figure illustrates the end-to-end sequence of the Secure ZTP process:

Figure 10: End-to-end sequence of the Secure ZTP process

Upgrade Image Using Secure ZTP

You can upgrade the system image using one of the following methods:

- Default installation method
- Preconfiguration script

Note Use only the preconfiguration script method to upgrade from Cisco IOS XR release 7.3.1 to either 7.7.21, 7.9.2, or 7.10.1.

Even though there are multiple ways of upgrading the software image using the default installation method, each router platform has a different installation behaviour. The default installation method uses a combination of `install add <>` and `install activate reload id <>` commands, which aren't optimized for all platforms. There's no single command in Cisco IOS XR that works on all platforms across all scenarios. Therefore, the recommended method is to use the preconfiguration script, which allows the use of any installation command. Also, this method allows you to modify traffic or rate-related configuration for download.

Here is a sample preconfiguration script:

```
[xr-vm_nodehost_CPU0:/misc/scratch]$ cat /disk0\:/ztp/customer/pre_config.candidate
#!/bin/bash

# !!!!!!!!!!!!!!!!!!!!!!! WARNING !!!!!!!!!!!!!!!!!!!!!!!
# For this script to work on Cisco IOSXR OS use Unix style
# EOL character - LF, not Windows style - CRLF

exec &> /dev/console # send logs to console
source /pkg/bin/ztp_helper.sh
export LOGFILE=/disk0:/ztp/user-script.log

#Set to 1 for GISO image upgrade
GISO_UPGRADE=0

# Crosswork parameters
HOST_IP="5.10.18.112"
PORT="5002"

# Software upgrade parameters
TARGET_SOFTWARE_VERSION="7.9.1.33I"
IMAGE_FILENAME="ncs5500-mini-x.iso"
IMAGE_MD5_CHECKSUM="20d020d9912eb01ce4b242532544cc0e"
#IMAGE_PACKAGE="ncs5500-mini-x-7.9.1.33I"
IMAGE_URL="http://${HOST_IP}:${PORT}/images/ncs7/${IMAGE_FILENAME}"

function ztp_log(){
    echo "$(date +"%b %d %H:%M:%S") "$1 >> $LOGFILE
}


function check_version(){
    # returns 0 is version matches, 1 otherwise
    local current_ver=`xrcmd "show version" | grep Version | grep Cisco | cut -d " " -f 6`;

    ztp_log "### ZTP version check current=$current_ver, target=$TARGET_SOFTWARE_VERSION
###";
```

```
    if [[ "$current_ver" = "$TARGET_SOFTWARE_VERSION" ]]; then
        ztp_log "### ZTP software version check result: match ###";
        return 0
    else
        ztp_log "### ZTP software version check result: mismatch ###";
        return 1
    fi
}

function download_image(){
    # Download image to harddisk:
    ztp_log "### IOS-XR INSTALL - downloading image $IMAGE_FILENAME from $IMAGE_URL ###"
    /usr/bin/wget ${IMAGE_URL} -O /harddisk:/$IMAGE_FILENAME 2>&1 >> $LOGFILE
    if [[ "$?" != 0 ]]; then
        ztp_log "### IOS-XR INSTALL - error downloading $IMAGE_FILENAME, check
/var/log/ztp_user_script.log for details ###"
        exit 1
    else
        ztp_log "### IOS-XR INSTALL - $IMAGE_FILENAME download completed ###";
    fi

    # check MD5 hash checksum
    ztp_log "### IOS-XR INSTALL - verifying image md5 checksum ###"
    local checksum=`xrcmd "show md5 file /harddisk:/$IMAGE_FILENAME"`;
    if [[ "$checksum" != *"$IMAGE_MD5_CHECKSUM"* ]]; then
        ztp_log "### IOS-XR INSTALL - error, image checksum $checksum does not match
$IMAGE_MD5_CHECKSUM, exiting ###"
        exit 1
    else
        ztp_log "### IOS-XR INSTALL - md5 checksum verification successful ###"
    fi
}

#optional
function install_commit_packages(){
    local output

    ztp_log "### IOS-XR INSTALL - Commit packages ###"
    output=$(xrcmd "install commit")
    ztp_log "$output"
}

#optional
function remove_inactive_packages(){
    local output

    # remove inactive packages if any
    ztp_log "### IOS-XR INSTALL - removing inactive packages ###"
    output=$(xrcmd "install remove inactive all synchronous")
    ztp_log "$output"
}

function install_giso_target_image(){
    local output
    touch /disk0\:/ztp/state/state_is_install_started
    # do GISO image install
    ztp_log "### IOS-XR INSTALL - doing GISO install replace ###"
    output=$(xrcmd "install replace /harddisk:/$IMAGE_FILENAME noprompt commit")
    ztp_log "$output"
    if [[ "$output" != *"aborted"* ]]; then
        ztp_log "### IOS-XR INSTALL - GISO install replace completed ###"
       ztp_log "### Upgraded IOS-XR to $TARGET_SOFTWARE_VERSION, device should reboot ###";

    else
```

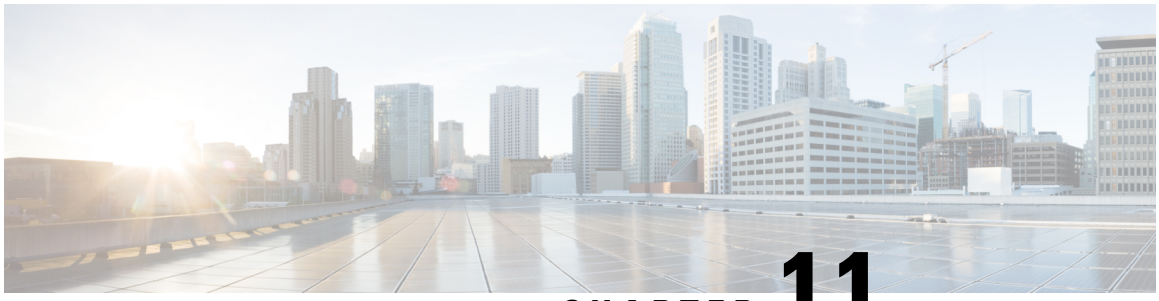
```
       ztp_log "### IOS-XR INSTALL - error, GISO install replace failed, check 'show install
 log' ###"
        return 1
    fi
}

function install_add_image(){
    local output

    # do  image install add
    ztp_log "### IOS-XR INSTALL - doing  install add ###"
    output=$(xrcmd "install add source /misc/disk1/ $IMAGE_FILENAME")
    ztp_log "$output"
    install_add_id=$(echo $output | awk '{print $5}')
    if [[ "$output" != *"aborted"* ]]; then
        ztp_log "### IOS-XR INSTALL -  install add completed ###"
    else
        ztp_log "### IOS-XR INSTALL - error,  install add failed, check 'show install log'
 ###"
        return 1
    fi
}

function install_target_image(){
    local output
    touch /disk0\:/ztp/state/state_is_install_started
    # do  image install
    ztp_log "### IOS-XR INSTALL - doing  install activate ###"
    output=$(xrcmd "install activate id $install_add_id noprompt synchronous")
    ztp_log "$output"
    if [[ "$output" != *"aborted"* ]]; then
        ztp_log "### IOS-XR INSTALL -  install activate completed ###"
       ztp_log "### Upgraded IOS-XR to $TARGET_SOFTWARE_VERSION, device should reboot ###";

    else
        ztp_log "### IOS-XR INSTALL - error,  install activate failed, check 'show install
 log' ###"
        return 1
    fi
}

# ==== Script entry point ==== #
# run version check and decide if need to download target image
check_version;
if [[ "$?" = 1 ]]; then
    ztp_log "### Software Version mismatch, downloading IOS-XR $TARGET_SOFTWARE_VERSION
###";
    download_image;
    NEED_UPGRADE=1
    ztp_log "### Downloaded IOS-XR $TARGET_SOFTWARE_VERSION image ###";
else
    NEED_UPGRADE=0
    ztp_log "### Image Download: Software Version match $TARGET_SOFTWARE_VERSION, nothing
to do ###";
fi

# do software upgrade
if [[ $NEED_UPGRADE = 1 ]]; then
    ztp_log "### Software Version mismatch, upgrading IOS-XR to $TARGET_SOFTWARE_VERSION
###";
    install_commit_packages;
    remove_inactive_packages;
    if [[ $GISO_UPGRADE = 1 ]]; then
        install_giso_target_image;
```

```
    else
        install_add_image;
        install_target_image;
    fi
else
    ztp_log "### Upgrade: Software Version match $TARGET_SOFTWARE_VERSION, nothing to do
###";
fi

ztp_log "### ZTP DONE ###";
```

Disable Secure ZTP

Execute the following commands to disable the secure ZTP:

```
Router# request consent-token generate-challenge secure-ztp auth-timeout 15
Router# request consent-token accept-challenge secure-ztp
```



CHAPTER 11

Disaster Recovery

The topics covered in this chapter are:

- Boot using USB Drive, on page 123
- Boot the Router Using iPXE, on page 125

Boot using USB Drive

The bootable USB drive is used to re-image the router for the purpose of system upgrade or boot the router in case of boot failure. The bootable USB drive can be created using a compressed boot file.

Create a Bootable USB Drive Using Compressed Boot File

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.

Note In case of failure to read or boot from USB drive, ensure that the drive is inserted correctly. If the drive is inserted correctly and still fails to read from USB drive, check the contents of the USB on another system.

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

Before you begin

- You have access to a USB drive with a storage capacity that is between 8GB (min) and 32 GB (max). USB 2.0 and USB 3.0 are supported.

Note The NCS-5501-SE PID supports a USB device with a storage capacity of 128 GB (max).

- Copy the compressed boot file from the software download page at cisco.com to your local machine. The file name for the compressed boot file is in the format *ncs5500-usb-boot-<release_number>.zip*.

Procedure

Step 1 Connect the USB drive to your local machine and format it with FAT32 or MS-DOS file system using the Windows Operating System or Apple MAC Disk Utility.

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

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

Step 4 Extract the content of the compressed boot file by unzipping it inside the USB drive. This converts the USB drive to a bootable drive.

Note
The content of the zipped file ("EFI" and "boot" directories) should be extracted directly into root of the USB drive. If the unzipping application places the extracted files in a new folder, move the "EFI" and "boot" directories to root of the USB drive.

Step 5 Eject the USB drive from your local machine.

What to do next

Use the bootable USB drive to boot the router or upgrade its image.

Boot the Router Using USB

The router can be booted using an external bootable USB drive. This might be required when the router is unable to boot from the installed image. A boot failure may happen when the image gets corrupted. During the USB boot, process the router gets re-imaged with the version available on the USB drive.

Note During the USB boot process, the router is completely re-imaged with the ISO image version present in the bootable USB drive. All existing configurations are deleted because the disk 0 content is erased. No optional packages are installed during the upgrade process; they need to be installed after the upgrade is complete.

Before you begin

Create a bootable USB drive. See Create a Bootable USB Drive Using Compressed Boot File, on page 123.

Procedure

Use one of the two methods to boot the router from USB:

- From Admin EXEC mode - Use this method if Admin LXC is up and Admin Exec prompt is accessible:
  a. Run the **show controller card-mgr inventory summary** command and identify the active RP with the Master chip.
  b. Insert the USB drive to the active RP.

c. Run **hw-module location {<loc> | all} bootmedia usb reload**. The RP boots the image from USB and installs the image onto the hard disk. The router boots from the hard disk after installation.

- From RP BIOS boot manager menu - Use this method if Admin LXC is not running:

Note
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.

a. Insert the USB drive.

b. Connect to the console.

c. Power the router.

d. Press **Esc** or **Del** to pause the boot process and get the RP to BIOS menu.

e. Select the USB from the boot menu on the RP to which the USB is connected to. The RP boot the image from USB and installs the image onto the hard disk. The router boots from the hard disk after installation.

Note
If there is no space in the RP, a prompt to either cancel the installation, or to continue with formatting the disk is displayed.

What to do next

- After the booting process is complete, specify the root username and password.
- Install the required optional packages.

Boot the Router Using iPXE

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

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.

Note PID and serial number is supported only if iPXE is invoked using the command `(admin) hw-module location all bootmedia network reload all`. If iPXE is selected manually from BIOS, PID and serial number is not supported.

Zero Touch Provisioning

Zero Touch Provisioning (ZTP) helps in auto provisioning after the software installation of the router using iPXE.

ZTP auto provisioning involves:

- **Configuration:** Downloads and executes the configuration file. The first line of the file must contain `!! IOS XR` for ZTP to process the file as a configuration.
- **Script:** Downloads and executes the script files. The 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 a script.

Setup DHCP Server

A DHCP server must be configured for IPv4, IPv6 or both communication protocols. The following example shows ISC-DHCP server running on Linux system.

Before you begin

- Consult your network administrator or system planner to procure IP addresses and a subnet mask for the management interface.
- Physical port Ethernet 0 on RP is the management port. Ensure that the port is connected to management network.
- Enable firewall to allow the server to process DHCP packets.
- For DHCPv6, a Routing advertisement (RA) message must be sent to all nodes in the network that indicates which method to use to obtain the IPv6 address. Configure Router-advertise-daemon (radvd, install using yum install radvd) to allow the client to send DHCP request. For example:

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

- The HTTP server can be in the same server as that of the DHCP server, or can be on a different server. After the IP address is assigned from DHCP server, the router must connect to the HTTP server to download the image.

Procedure

Step 1 Create the `dhcpd.conf` file (for IPv4, IPv6 or both communication protocols), `dhcpv6.conf` file (for IPv6) or both in the `/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 router.

Step 2 Test the server once the DHCP server is running. For example, for IPv4:

- Use MAC address of the router:

  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 bootfile to the system when required.

  ```
  host ncs5500
  {
  hardware ethernet <router-mac-address>;
  if exists user-class and option user-class = "iPXE" {
   filename = "http://<httpserver-address>/<path-to-image>/ncs5500-mini-x.iso";
  }
  ```

  Ensure that the above configuration is successful.
- Use serial number of the router:

  ```
  host ncs5500
  {
  option dhcp-client-identifier "<router-serial-number>";
    filename "http://<IP-address>/<path-to-image>/ncs5500-mini-x.iso";
    fixed-address <IP-address>;
  }
  ```

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

Step 3 Restart DHCP.

```
killall dhcpd
/usr/sbin/dhcpd -f -q -4 -pf /run/dhcp-server/dhcpd.pid
-cf /etc/dhcp/dhcpd.conf ztp-mgmt &
```

Example

The example shows a sample `dhcpd.conf` file:

```
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 routers <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>";
}
```

The example shows a sample `dhcpd6.conf` file:

```
option dhcp6.name-servers <ip-address-server>;
option dhcp6.domain-search "cisco.com";
dhcpv6-lease-file-name "/var/db/dhcpd6.leases";
option dhcp6.info-refresh-time 21600;
option dhcp6.bootfile-url code 59 = string;
subnet6 <subnet> netmask <netmask> {
       range6 2001:1851:c622:1::2 2001:1851:c622:1::9;
        option dhcp6.bootfile-url "http://<address>/<path>/<image.bin>";
```

What to do next

Invoke ZTP.

Invoke ZTP

ZTP runs within the XR namespace, and within the global VPN routing/forwarding (VRF) namespace for management interfaces and line card interfaces.

Before you begin

Ensure that a DHCP server is setup. For more information, see Setup DHCP Server, on page 126.

Procedure

Edit the dhcpd.conf file to utilize the capabilities of ZTP.

The following example shows a sample DHCP server configuration including iPXE and ZTP:

```
host <host-name>
{
hardware ethernet <router-serial-number or mac-id>;
fixed-address <ip-address>;
  if exists user-class and option user-class = "iPXE" {
  # Image request, so provide ISO image
  filename "http://<ip-address>/<directory>/ncs5500-mini-x.iso";
  } else
{
  # Auto-provision request, so provide ZTP script or configuration
  filename "http://<ip-address>/<script-directory-path>/ncs5500-ztp.script";
  #filename "http://<ip-address>/<script-directory-path>/ncs5500-ztp.cfg
  }
}
```

Note

Either the ZTP .script file or the .cfg file can be provided at a time for auto-provisioning.

With this configuration, the system boots using ncs5500-mini-x.iso during installation, and then download and execute ncs5500-ztp.script when XR VM is up.

Invoke ZTP Manually

ZTP can also be invoked manually with the modified one touch provisioning approach. The process involves:

Before you begin

A configuration file can be used to specify a list of interfaces that will be brought up in XR and DHCP will be invoked on. /pkg/etc/ztp.config is a platform specific file that allows the platform to specify which if any additional interfaces will be used.

```
#
# List all the interfaces that ZTP will consider running on. ZTP will attempt
# to bring these interfaces. At which point dhclient will be able to use them.
#
# Platforms may add dynamically to this list.
#
#ZTP_DHCLIENT_INTERFACES=" \
#    Gi0_0_0_0 \
#"
...
```

Procedure

Step 1 Boot the router.

Step 2 Login manually.

Step 3 Enable interfaces.

Step 4 Invoke a new ZTP DHCP session manually using the **ztp initiate** command.

```
Router#ztp initiate
```

For example, to send DHCP requests on the GigabitEthernet interface 0/0/0/0, run the command:

```
Router#ztp initiate debug verbose interface GigabitEthernet0/0/0/0
```

ZTP will run on the management port by default unless the platform has configured otherwise. The logs will be logged in `/disk0:/ztp/ztp/log` location.

Note
To configure a 40G interface into 4 separate 10G interfaces, use the **ztp breakout nosignal-stay-in-breakout-mode** command.

Note
To enable dataport breakouts and invoke DHCP sessions on all dataport and line card interfaces that are detected, use the **ztp breakout** command.

```
Router#ztp breakout debug verbose
```

```
Router#ztp initiate dataport debug verbose
Invoke ZTP?(this may change your configuration) [confirm] [y/n]:
```

To override the prompt:

```
Router#ztp initiate noprompt
Invoke ZTP?(this may change your configuration) [confirm] [y/n]:

ZTP will now run in the background.
Please use "show logging" or look at /disk0:/ztp/ztp/log to check progress.
```

ZTP runs on the management interfaces that are `UP` by default.

Step 5 To terminate the ZTP session, use the **ztp terminate** command.

What to do next

Boot the router using iPXE.

Note While ZTP executes, intermediate configuration is created to control interface addressing and routing information. When the configuration file is downloaded, this immediate configuration is removed and downloaded configuration will be applied. But, when the script file is downloaded intermediate configuration is kept for scripts to communicate with remote hosts. Once the script is ended, the final configuration needs to be applied to the router using the **commit replace** command. This ensures that the intermediate configuration is replaced. If the **commit replace** command is not applied after the script execution, intermediate configuration will remain and the final configuration will not take effect.

Boot the Router Using iPXE

Before you use the iPXE boot, ensure that:

- DHCP server is set and is running.
- You have logged in to the System Admin console using the **admin** command.

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

```
hw-module location all bootmedia network reload
```

Example:

```
sysadmin-vm:0_RP0# hw-module location all bootmedia network reload
Wed Dec 23 15:29:57.376 UTC
Reload hardware module ? [no,yes]
```

The following example shows the output of the command:

```
iPXE 1.0.0+ (3e573) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP TFTP VLAN EFI ISO9660 NBI Menu
Trying net0...
net0: c4:72:95:a6:14:e1 using dh8900cc on PCI01:00.1 (open)
[Link:up, TX:0 TXE:0 RX:0 RXE:0]
Configuring (net0 c4:72:95:a6:14:e1)................... Ok << Talking to DHCP/PXE server to
 obtain network information
```

```
net0: 10.37.1.101/255.255.0.0 gw 10.37.1.0
net0: fe80::c672:95ff:fea6:14e1/64
net0: 2001:1800:5000:1:c672:95ff:fea6:14e1/64 gw fe80::20c:29ff:fefb:b9fe
net1: fe80::c672:95ff:fea6:14e3/64 (inaccessible)
Next server: 10.37.1.235
Filename: http://10.37.1.235/ncs5500/ncs5500-mini-x.iso


http://10.37.1.235/ ncs5500/ncs5500-mini-x.iso... 58% << Downloading file as indicated by
DHCP/PXE server to boot install image
```

Disaster Recovery Using Manual iPXE Boot

Manually booting the system using iPXE can be used to reinstall a clean system in case of a corrupt install or recover lost password. However, all the disks will be wiped out and the configuration will be removed.

Procedure

Step 1 Press **Del** or **Esc** key to enter the Boot manager.

Step 2 Use the arrow keys (up, down) to select **UEFI: Built-in EFI IPXE** to enable iPXE boot. The iPXE boot launches the auto boot.

If the standby RP is being recovered and an active RP is present, the image is pulled from the active RP and auto boot is launched. In case of a single RP, or the other RP is in BIOS or unavailable, iPXE iteratively tries to configure the available interfaces in a loop. The following message is displayed at the end of every iteration:

```
Press Ctrl-B for the iPXE command line...
```

To manually boot using iPXE, press **Ctrl-B** keys to reach the iPXE command line.

Step 3 Identify the management interface. If the management interface is connected properly and is UP, it displays `Link:up` in the following output:

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 10.x.x.y
iPXE> set net6/netmask 255.x.x.x
iPXE> set net6/gateway 10.x.x.x
```

```
iPXE>
iPXE> ifopen net6

iPXE> ping 10.x.x.z
64 bytes from 10.x.x.z: seq=1
64 bytes from 10.x.x.z: seq=2
Finished: Operation canceled (http://ipxe.org/0b072095)

iPXE> boot http://10.x.x.z/<dir-to-iso>/ncs5500-mini-x.iso-<version>_IMAGE
http://10.x.x.z/<dir-to-iso>/ncs5500-mini-x.iso-<version>_IMAGE... ok
```

Choose the net interface that shows `Link:up`. If there are multiple interfaces that show the status as UP, identify the management interface with MAC address.

iPXE also supports HTTP, TFTP and FTP. For more information, see https://ipxe.org/cmd.

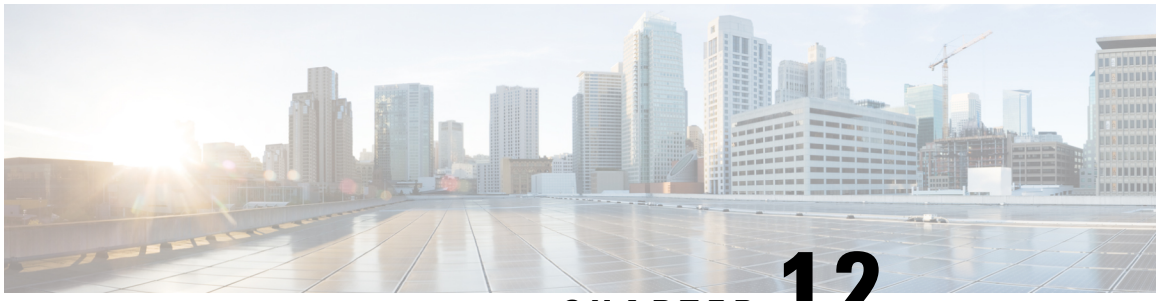
Note
Keep the standby RP in BIOS while installing the active RP.

After installing the mini ISO image, the system reboots. After successful reboot, specify the root username and password. Once you get back to the XR prompt, you can load the configuration and install remaining packages.

PART II

Setup System and Install IOS XR7 Software

- Setup Cisco NCS 5700 Series Routers with XR7 OS, on page 135
- Install Cisco IOS XR7 on NCS 5700 Series Routers, on page 151
- Workflow for Installing Cisco IOS XR7 Software, on page 155
- Additional Install Operations, on page 165



CHAPTER 12

Setup Cisco NCS 5700 Series Routers with XR7 OS

The Cisco Network Convergence System 5700 Series builds on the Cisco NCS 5500 fixed systems by combining the forwarding ASIC design with the Cisco IOS XR7 OS. The Cisco NCS 5700 series chassis is a standalone 1RU router that offers aggregation, distributed core and peering fabric, and 100G Top of Rack (ToR).

The following variants of Cisco NCS 5700 series router run on XR7 OS:

- NCS-57B1-6D24-SYS
- NCS-57B1-5DSE-SYS

XR7 OS provides significant architectural enhancements to Cisco IOS XR in these areas:

- **Modularity:** Decoupled hardware and software; disintegrated software with the flexibility to consume software packages based on requirement
- **Programmability:** Cloud scale enhancement with model-drivel APIs at all layers
- **Manageability:** Simplified software management and installation that is based on Linux tools

This document helps you set up the Cisco NCS 5700 series routers with XR7 OS. You will bring-up the router, run a health check of the system, create user profiles, and assign privileges.

- Bring-up the Router, on page 135
- Perform Preliminary Checks with Router, on page 141
- Create Users and Assign Privileges on the Router, on page 147

Bring-up the Router

Connect to the console port of the router, and power ON the router. By default, this console port connects to the XR console. If necessary, after configuration, establish subsequent connections through the management port.

The following table shows the console settings:

Table 8: Console Settings

Baud rate (in bps)	Parity	Stop bits	Data bits
115200	None	2	8

The baud rate is set by default and cannot be changed.

The router can be accessed using remote management protocols, such as SSH, Telnet, SCP and FTP. SSH is included in the software image by default, but telnet is not part of the software image. You must manually install the telnet optional package to use it.

After booting is complete, you must create a username and password. This credential is used to log on to the XR console, and get to the router prompt.

You can start or stop the console by using the following keyboard shortcuts:

- To start the console, press Ctrl + q.
- To stop the console, press Ctrl + s.

Note that by using Ctrl + s, the console output will be locked and you will need to initiate a Ctrl + q sequence to restore the console prompt.

The router completes the boot process using the pre-installed operating system (OS) image. If no image is available within the router, the router can be booted using iPXE boot or an external bootable USB drive.

Boot the Router Using Manual iPXE

Manually boot the router using iPXE if the router fails to boot when powered ON. An alternate method is to Boot from a USB device.

iPXE is a pre-boot execution environment in the network card of the management interfaces. It works at the system firmware (UEFI) level of the router. iPXE boot re-images the system, boots the router in case of a boot failure, or in the absence of a valid bootable partition. iPXE downloads the ISO image, installs the image, and finally bootstraps inside the new installation.

You need a server running HTTPS, HTTP, or TFTP. Bring-up the PXE prompt using the following steps:

Procedure

Step 1 Power ON the router.

Step 2 Press Esc or Del keys continuously (quick and repeated press and release) to pause the boot process, and get to the BIOS menu.

Step 3 Select `Boot Manager`, and then select `Built-in iPXE` option.

Step 4 When PXE boot starts reaching for a PXE server, press **Ctrl+B** keys to break into the PXE prompt.

Step 5 Add the following configuration for the router. This is required for the router to connect with the external server to download, and install the image. You can use HTTP, HTTPS or TFTP server.

Example:

```
iPXE> ifopen net0                   #Open the interface connecting outside world
iPXE> set net0/ip 10.0.0.2          #Configure the ip address of your router
```

```
iPXE> set net0/gateway 10.0.0.1      #configure the GW
iPXE> set net0/netmask 255.0.0.0     #Configure the Netmask
iPXE> ping 10.0.0.1                  #Check you can reach GW
iPXE> ping 192.0.2.0                 #check you can reach to your server running tftp or http or
https
iPXE> boot http://192.0.2.0/<directory-path>5700-x64.iso    #Copy the image on the http/https/tftp
server in any path and then point to download the image from there.
```

Note
To rectify errors while typing the command, use **Ctrl+H** keys to delete a character.

If a PXE server is configured to run a DHCP server, it assigns an IP address to the Ethernet Management interface of the router. This provides a channel to download the image that is required to re-image a router in case of a boot failure.

```
Router#reload bootmedia network location all
Proceed with reload? [confirm]
```

Note
Use the **force** option to perform an ungraceful reload of the specified location or hardware module. When **force** option is used along with the **all** location, the chassis undergoes an ungraceful reload. Use the **noprompt** option to avoid the prompt to confirm the operation. The **force** option is not recommended, and should not be used during regular operations.

Boot the Router Using USB Drive

Boot the router using USB drive if the router fails to boot when powered ON. An alternate method is to boot the router using iPXE.

Before you begin

Have access to a USB drive with a storage capacity that is between 8GB (min) and 32 GB (max). USB 2.0 and USB 3.0 are supported.

Note Use this procedure only on the active RP; the standby RP must either be powered OFF or removed from the chassis. After the active RP is installed with images from the USB drive, insert or power ON the standby RP as appropriate.

Procedure

Step 1 Copy the bootable file to a USB disk.

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.

Note
If you are unable to boot from a USB drive, remove and insert the drive again. If the drive is inserted correctly, and still fails to read from the USB drive, check the contents of the USB on another system.

This task can be completed using Windows, Linux, or MAC operating systems available on your local machine.

a) Connect the USB drive to your local machine and format it with FAT32 or MS-DOS file system using the Windows Operating System or Apple MAC Disk Utility. To check if the disk is formatted as FAT32, right click on the USB disk, and view the properties.
b) Copy the compressed boot file in .zip format from the image file to the USB drive. This .zip file can be downloaded from the Cisco Software Download center.
c) Verify that the copy operation is successful. To verify, compare the file size at source and destination. Additionally, verify the MD5 checksum value.
d) Extract the contents of the compressed boot file by unzipping it inside the USB drive. This converts the USB drive to a bootable drive.

Note
Extract the contents of the zipped file ("EFI" and "boot" directories) directly into 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.

e) Eject the USB drive from your local machine.

Step 2 Use the bootable USB drive to boot the router or upgrade its image using one of the following methods:

- **Boot menu**

  a. Insert the USB drive, and connect to the console.

  b. Power ON the router.

  c. Press Esc or Del to pause the boot process, and get to the BIOS menu.

  d. Select `Boot Manager`, and then select the `USB` option from the boot menu.

```
Cisco BIOS Setup Utility - Copyright (C) 2019 Cisco Systems, Inc

Boot Override
UEFI: Micron_M600_MTFDDAT064MBF, Partition 4
UEFI: Built-in iPXE
URFI: Built-in Shell
URFI: Built-in Grub
UEFI: USB Flash Memory1.00, Partition 1
```

  The system boots the image from the USB drive, and installs the image onto the hard disk. The router boots from the hard disk after installation.

- **XR CLI**

  Use this method if you can access the XR prompt.

  a. Insert the USB device in the RP.

  b. Access the XR prompt and run the command:

```
Router#reload bootmedia usb noprompt

Welcome to GRUB!!
Verifying (hd0,msdos1)/EFI/BOOT/grub.cfg...
(hd0,msdos1)/EFI/BOOT/grub.cfg verified using Pkcs7 signature.
Loading Kernel..
Verifying (loop)/boot/bzImage...
(loop)/boot/bzImage verified using attached signature.
Loading initrd..
Verifying (loop)/boot/initrd.img
```

Use the **force** option to perform an ungraceful reload of the specified location or hardware module. When **force** option is used along with the **all** location, the chassis undergoes an ungraceful reload. Use the **noprompt** option to avoid the prompt to confirm the operation. The **force** option is not recommended, and should not be used during regular operations.

The system boots the image from the USB and installs the image onto the hard disk. The router boots from the hard disk after installation.

Note
Execute the `install commit` command before proceeding to the next install iteration, while performing cyclic upgrade and downgrade tests.

Configure the Management Port on the Router

To use the management port for system management and remote communication, you must configure an IP address and a subnet mask for the Management Ethernet interface.

Note We recommend that you use a Virtual Private Network (VPN) routing and the forwarding (VRF) on the Management Ethernet interface.

Before you begin

- Consult your network administrator or system planner to procure IP addresses and a subnet mask for the management interface.
- Physical port Ethernet 0 on RP is the management port. Ensure that the port is connected to the management network.

Procedure

Step 1 Configure a VRF.

Example:

```
Router#conf t
Router(config)#vrf <vrf-name>
Router(config-vrf)#exit
```

Step 2 Enter interface configuration mode for the management interface of the RP.

Example:

```
Router(config)#interface mgmtEth 0/RP0/CPU0/0
```

Step 3 Assign an IP address and a subnet mask to the interface.

Example:

```
Router(config-if)#ipv4 address 10.10.10.1/8
```

Step 4 Configure the Management Ethernet interface under the VRF.

Example:

```
Router(config-if)#vrf <vrf-name>
```

Step 5 Exit the management interface configuration mode.

Example:

```
Router(config-if)#exit
```

Step 6 Place the interface in `UP` state.

Example:

```
Router(config)#no shutdown
```

Step 7 Specify the IP address of the default-gateway to configure a static route; this is used for communications with devices on other networks.

Example:

```
Router(config)#router static vrf <vrf-name> address-family ipv4 unicast 0.0.0.0/0 10.10.10.1
```

Step 8 Commit the configuration.

Example:

```
Router(config)#commit
```

Step 9 Connect to the management port to the ethernet network. With a terminal emulation program, establish a SSH or telnet connection to the management interface port using its IP address.

Synchronize Router Clock with NTP Server

Synchronize the XR clock with that of an NTP server to avoid a deviation from true time.

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 to the server. A `stratum 2` time server receives its time through NTP from a `stratum 1` time server, and so on.

Note The Cisco implementation of NTP does not support stratum 1 service.

Before you begin

Configure and connect to the management port.

Procedure

Step 1 Enter the XR configuration mode.

Example:

```
Router#configure
```

Step 2 Synchronize the console clock with the specified sever.

Example:

```
Router(config)#ntp server <NTP-source-IP-address>
```

The NTP source IP address can either be an IPv4 or an IPv6 address. For example:

IPv4:

```
Router(config)#ntp server 192.0.2.0
```

IPv6:

```
Router(config)#ntp server 2001:DB8::1
```

Note
The NTP server can also be reachable through a VRF if the Management Ethernet interface is in a VRF.

Step 3 Commit the configuration.

Example:

```
Router(config-ntp)#commit
```

Step 4 Verify that the clock is synchronised with the NTP server.

Example:

```
Router#show ntp status
Clock is synchronized, stratum 3, reference is 192.0.2.0
nominal freq is 1000000000.0000 Hz, actual freq is 1000000000.0000 Hz, precision is 2**24
reference time is E12B1B02.8BB13A2F (08:42:42.545 UTC Tue Sep 17 2019)
clock offset is -3.194 msec, root delay is 4.949 msec
root dispersion is 105.85 msec, peer dispersion is 2.84 msec
loopfilter state is 'FREQ' (Drift being measured), drift is 0.0000000000 s/s
system poll interval is 64, last update was 124 sec ago
authenticate is disabled
```

Perform Preliminary Checks with Router

After successfully logging into the console, you must perform some preliminary checks to verify the correctness of the default setup. Correct any issues that arise before proceeding with further configurations.

Verify Software Version on Router

The router is shipped with the Cisco IOS XR software pre-installed. Verify that the latest version of the software is installed. If a newer version is available, perform a system upgrade. Installing the newer version of the software provides the latest feature set on the router.

You can view the overview of the running software. This includes the following information:

- Image name and version
- User who built the image
- Time the image was built

- Build workspace
- Build host
- ISO label

Note If any modifications are made to the running software on the booted ISO, only the IOS XR version is displayed in the label field and not the label included in the ISO.

- Copyright information
- Hardware information

Display the version of the Cisco IOS XR software, and its various software components that are installed on the router.

NCS 5700 Series Routers:

```
Router#show version
Cisco IOS XR Software, Version 7.3.1 LNT
Copyright (c) 2013-2022 by Cisco Systems, Inc.
Build Information:
Built By :   xyz
Built On :   Tue June 07 19:43:44 UTC 2021
Build Host : iox-lnx-064
Workspace :  ../ncs5700/ws
Version :    7.3.1
Label :      7.3.1
cisco NCS-57B1-5DSE-SYS (D-1563N @ 2.00GHz) processor with 32GB of memory
NCS5700 uptime is 3 weeks, 1 day, 10 hours, 11 minutes
NCS5700 Fixed Scale HW Flexible Consumption Need Smart Lic
```

Verify Status of Hardware Modules on Router

Hardware modules such as fan trays, and power modules are installed on the router. The firmware on various hardware components of the router must be compatible with the Cisco IOS XR image installed. Incompatibility may cause the router to malfunction. Verify that all hardware and firmware modules are installed correctly and are operational.

Before you begin

Ensure that all required hardware modules are installed on the router.

Procedure

Step 1 View the status of the system.

Example:

NCS 5700 Series Routers:

```
Router#show platform
Node              Type                       State                  Config state
```

```
------------------------------------------------------------------------------

0/RP0/CPU0        NCS-57B1-5DSE-SYS(Active) IOS XR RUN              NSHUT

0/PM0             PSU2KW-ACPI               OPERATIONAL             NSHUT

0/PM1             PSU2KW-ACPI               OPERATIONAL             NSHUT

0/FT0             N5700-FAN                 OPERATIONAL             NSHUT

0/FT1             N5700-FAN                 OPERATIONAL             NSHUT

0/FT2             N5700-FAN                 OPERATIONAL             NSHUT

0/FT3             N5700-FAN                 OPERATIONAL             NSHUT

0/FT4             N5700-FAN                 OPERATIONAL             NSHUT

0/FT5             N5700-FAN                 OPERATIONAL             NSHUT
```

Step 2 View the list of hardware and firmware modules detected on the router.

Example:

NCS 5700 Series Routers:

```
Router#show hw-module fpd
Auto-upgrade:Enabled
Attribute codes: B golden, P protect, S secure
                   FPD Versions
                                                    ================
Location   Card type             HWver FPD device       ATR Status   Running Programd  Reload Loc
------------------------------------------------------------------------------------------------
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   IoFpga               CURRENT    0.08    0.08         0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   IoFPgaGolden     B   CURRENT            0.02         0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   Primary-BIOS      S  CURRENT    1.10    1.10         0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   StdbyFpga         S  CURRENT    0.24    0.24         0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   StdbyFpgaGolden  BS  NEED UPGD          0.00         0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   TamFw             S  CURRENT    6.05    6.05         0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   TamFwGolden      BS  NEED UPGD          0.00         0/RP0
0/PM0      PSU2KW-ACPI           0.0   PO-PrimMCU           CURRENT    1.08    1.08       NOT REQ
0/PM1      PSU2KW-ACPI           0.0   PO-PrimMCU           CURRENT   17.56   17.56       NOT REQ
```

From the result, verify that all hardware modules that are installed on the chassis are listed. If a module is not listed, it indicates that the module is malfunctioning, or is not installed properly. Remove and reinstall that hardware module.

In the preceding output, some of the significant fields are:

- FPD Device—Name of the hardware component, such as IO FPGA, IM FPGA, or BIOS

  Note
  Golden FPDs are not field upgradable.

- Status—Upgrade status of the firmware. The different states are:

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.

Status	Description
NEED UPGD	A newer firmware version is available in the installed image. We recommend that you to perform an upgrade of the firmware version.
RLOAD REQ	The upgrade is complete, and the ISO image requires a reload.
UPGD DONE	The firmware upgrade is successful.
UPGD FAIL	The firmware upgrade has failed.
UPGD PREP	The FPD firmware is preparing for upgrade.
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.

- Running—Current version of the firmware running on the FPD
- Programd—Version of the FPD programmed on the module

Step 3 If necessary, upgrade the required firmware.

Example:

```
Router#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
BIOS and IOFPGA upgrades require a power cycle of the router for the new version to take effect.

Step 4 After the modules are upgraded verify the status of the modules.

Example:

NCS 5700 Series Routers:

```
Router#show hw-module fpd
Auto-upgrade:Enabled
```

```
Attribute codes: B golden, P protect, S secure
                    FPD Versions
                                                          =================
Location   Card type             HWver FPD device       ATR Status   Running Programd  Reload Loc
-----------------------------------------------------------------------------------------------
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   IoFpga               CURRENT    0.08    0.08         0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   IoFPgaGolden     B   CURRENT    0.02                 0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   Primary-BIOS      S  CURRENT    1.10    1.10         0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   StdbyFpga         S  CURRENT    0.24    0.24         0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   StdbyFpgaGolden  BS  CURRENT            0.00         0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   TamFw             S  CURRENT    6.05    6.05         0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   TamFwGolden      BS  RLOAD REQ  0.00    0.01         0/RP0
0/PM0      PSU2KW-ACPI           0.0   PO-PrimMCU           CURRENT    1.08    1.08       NOT REQ
0/PM1      PSU2KW-ACPI           0.0   PO-PrimMCU           CURRENT   17.56   17.56       NOT REQ
```

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

Step 5 Reload the individual nodes that required an upgrade.

Example:

```
Router#reload location <node-location>
```

Step 6 Verify that all nodes that required an upgrade show an updated status of `CURRENT` with an updated FPD version.

Example:

NCS 5700 Series Routers:

```
Router#show hw-module fpd
Auto-upgrade:Enabled
Attribute codes: B golden, P protect, S secure
                    FPD Versions
                                                          =================
Location   Card type             HWver FPD device       ATR Status   Running Programd  Reload Loc
-----------------------------------------------------------------------------------------------
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   IoFpga               CURRENT    0.08    0.08         0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   IoFPgaGolden     B   CURRENT    0.02                 0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   Primary-BIOS      S  CURRENT    1.10    1.10         0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   StdbyFpga         S  CURRENT    0.24    0.24         0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   StdbyFpgaGolden  BS  CURRENT    0.00    0.01         0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   TamFw             S  CURRENT    6.05    6.05         0/RP0
0/RP0/CPU0 NCS-57B1-5DSE-SYS     0.1   TamFwGolden      BS  CURRENT    0.00    0.01         0/RP0
0/PM0      PSU2KW-ACPI           0.0   PO-PrimMCU           CURRENT    1.08    1.08       NOT REQ
0/PM1      PSU2KW-ACPI           0.0   PO-PrimMCU           CURRENT   17.56   17.56       NOT REQ
```

Verify Interface Status on the Router

After the router has booted, all available interfaces must be discovered by the system. If interfaces are not discovered, it might indicate a malfunction in the unit.

Procedure

View the interfaces discovered by the system.

Example:

```
Router#show ipv4 interfaces brief
Interface                      IP-Address      Status          Protocol Vrf-Name
-------------------------------------------------------------------------------HundredGigE0/0/0/0
             unassigned      Shutdown        Down     default
HundredGigE0/0/0/1             unassigned      Shutdown        Down     default
HundredGigE0/0/0/2             unassigned      Shutdown        Down     default
HundredGigE0/0/0/3             unassigned      Shutdown        Down     default
HundredGigE0/0/0/4             unassigned      Shutdown        Down     default
HundredGigE0/0/0/5             unassigned      Shutdown        Down     default
HundredGigE0/0/0/6             unassigned      Shutdown        Down     default
HundredGigE0/0/0/7             unassigned      Shutdown        Down     default
------------------------ <snip> --------------------------------TenGigE0/0/0/18/0
unassigned      Up              Up       default
TenGigE0/0/0/18/1              unassigned      Up              Up       default
TenGigE0/0/0/18/2              unassigned      Up              Up       default
TenGigE0/0/0/18/3              unassigned      Up              Up       default
MgmtEth0/RP0/CPU0/0            10.10.10.1      Up              Up       default
```

When a router is turned ON for the first time, all interfaces are in the `unassigned` state. Verify that the total number of interfaces displayed in the result matches with the actual number of interfaces present on the router.

Verify Node Status on Router

Each card on the router represents a node.

Procedure

Verify the operational status of the node.

Example:

NCS 5700 Series Routers:

```
Router#show platform
Node              Type                         State                    Config state
-------------------------------------------------------------------------------------

0/RP0/CPU0        NCS-57B1-5DSE-SYS(Active) IOS XR RUN               NSHUT

0/PM0             PSU2KW-ACPI               OPERATIONAL              NSHUT

0/PM1             PSU2KW-ACPI               OPERATIONAL              NSHUT

0/FT0             N5700-FAN                 OPERATIONAL              NSHUT

0/FT1             N5700-FAN                 OPERATIONAL              NSHUT

0/FT2             N5700-FAN                 OPERATIONAL              NSHUT

0/FT3             N5700-FAN                 OPERATIONAL              NSHUT

0/FT4             N5700-FAN                 OPERATIONAL              NSHUT

0/FT5             N5700-FAN                 OPERATIONAL              NSHUT
```

Displays the status of nodes present in the chassis.

Verify that the software state and the hardware state of FTs, and power modules are listed, and their state is `OPERATIONAL`. This indicates that the XR console is operational on the cards.

The platform states are described in the following table:

Card Type	State	Description
All	UNKNOWN	Error – Internal card record is not available
All	IDLE	Error – Card state is not initialized
All	DISCOVERED	Card is detected
All	POWERED_ON	Card is powered on
FC, FT, PT, PM	OPERATIONAL	Card is operating normally and is functional

Create Users and Assign Privileges on the Router

Users are 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

XR has its AAA separate from Linux. XR AAA is the primary AAA system. A user created through XR can log in directly to the EXEC prompt when connected to the router. A user created through Linux can connect to the router, but arrive at the bash prompt. The user must log in to XR explicitly in order to access the XR EXEC prompt.

Configure AAA authorization to restrict users from uncontrolled access. If AAA authorization is not configured, the command and data rules associated to the groups that are assigned to the user are bypassed. A user can have full read-write access to IOS XR configuration through Network Configuration Protocol (NETCONF), google-defined Remote Procedure Calls (gRPC), or any YANG-based agents. In order to avoid granting uncontrolled access, enable AAA authorization before setting up any configuration. To gain an understanding about AAA, and to explore the AAA services, see the *Configuring AAA Services* chapter in the *System Security Configuration Guide for Cisco NCS5500 Series Routers*.

Create a User Profile

Create new users and include the user in a user group with certain privileges. The router supports a maximum of 1024 user profiles.

In this task, you create a user, `user1`, password for this user, `pw123`, and assign the user to a group `root-lr`.

Procedure

Step 1 Enter the XR configuration mode.

Example:

```
Router#config
```

Step 2 Create a new user.

Example:

```
Router(config)#username user1
```

Step 3 Create a password for the new user.

Example:

```
Router(config-un)#password pw123
```

Step 4 Assign the user to group `root-lr`.

Example:

```
Router(config-un)#group root-lr
```

All users have `read` privileges. However, users can be assigned to `root-lr` usergroup. These users inherit the `write` privileges where users can create configurations, create new users, and so on.

Step 5 Commit the configuration.

Example:

```
Router(config-un)#commit
```

What to do next

With the router set up, you can manage your system, install software packages, and configure your network.

Create a User Group

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 router supports a maximum of 32 user groups.

In this task, you create a group name, `group1`, and assign a user, `user1` to this group.

Before you begin

Create a user profile. See Create a User Profile, on page 147.

Procedure

Step 1 Enter the XR configuration mode.

Example:

```
Router#config
```

Step 2 Create a new user group, `group1`.

Example:

```
Router#(config)#group group1
```

Step 3 Specify the name of the user, `user1` to assign to this user group.

Example:

```
Router#(config-GRP)#username user1
```

You can specify multiple user names enclosed withing double quotes. For example, users "*user1 user2 ...*".

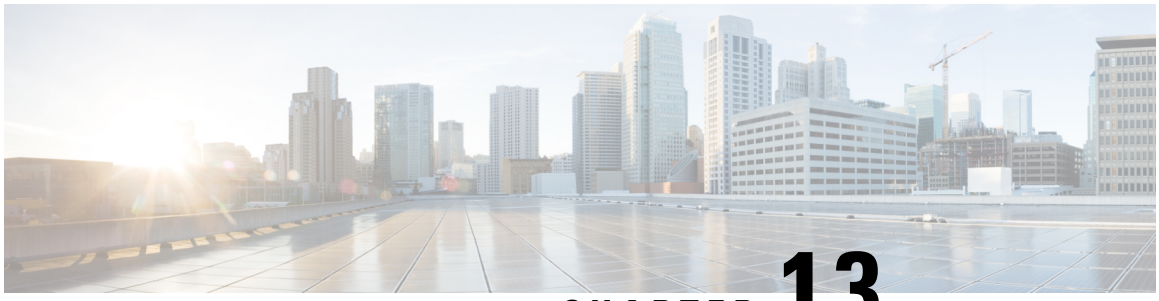
Step 4 Commit the configuration.

Example:

```
Router#commit
```

What to do next

With the router set up, you can manage your system, install software packages, and configure your network.



CHAPTER 13

Install Cisco IOS XR7 on NCS 5700 Series Routers

This section describes the concepts and procedures for upgrading or downgrading your system, installing optional packages, and obtaining bug fixes for the Cisco NCS 5700 series routers.

Cisco NCS 5700 series routers use the Cisco IOS XR7 framework. This framework refers to a set of architectural enhancements to the Cisco IOS XR software around the capabilities of modularity, simplified platform infrastructure, and programmability at various software layers.

The Cisco IOS XR software is composed of a base image (ISO) that provides the XR infrastructure, and a TAR file. The TAR file is made up of a set of packages (also called RPMs). These packages comprise mandatory and optional RPMs that can be deployed based on specific requirements. This software modularity approach provides a flexible consumption model that allows you to install a subset of IOS XR packages on devices based on your individual requirements. All critical components are modularized as packages so that you can select the features that you want to run on your router. For example, components like CDP and Telnet are modularized as packages and separated from the base image. These packages can be individually installed, upgraded or removed based on your requirements.

Cisco IOS XR7 install is Dandified Yum- or DNF-based software package manager that is used to install, update, and remove packages on the RPM-based Linux distributions. The package manager is used to automatically compute dependencies and determine the actions required to install packages.

Note For information on how to download the system upgrade procedures document, see the *About Cisco IOS XR Software Upgrade and Downgrade Guide*.

The following NCS 5700 line card variants support XR7 in Cisco IOS XR Release 7.3.1:

- NCS57B1-6D24
- NCS57B1-5DSE

- Software Deliverables and Terminologies, on page 151
- Supported Packages, on page 153

Software Deliverables and Terminologies

This section provides an understanding of the terms that are associated with installing the software.

- **Package:** A package, also known as an RPM, contains the software and metadata. A package is in `.rpm` format. A package can be mandatory or optional. Mandatory packages are part of the install image and cannot be removed. Optional packages are not required for the software to work, but can be installed to provide additional functionalities, and can be installed or removed based on requirement.
- **ISO image:** A bootable image that contains the installable files of the base operating system (OS). The image contains the IOS XR (XR7) infrastructure for fixed and distributed platforms in the form of base ISO image, mandatory RPMs. An ISO image is in `.iso` format.
- **Golden ISO (GISO):** A customizable ISO image that is built to contain preferable packages to suit diverse installation requirements. GISO can be customized to include a standard base image with the basic functional components, additional RPMs, bug fixes, and configuration files based on your requirement. GISO can also include a custom image label which can be seen using the **show version** command. From IOS XR Release 7.5.x and later, you can build your GISO image without support from Cisco. For more information, see Golden ISO images.
- **Source:** A location where packages can be installed from. The source can be a repository, local directory or a local tar file.
- **Repository:** A directory of RPMs and their metadata that a package manager uses to query the packages.
- **Active package:** A package whose software is currently running on the system.
- **Committed package:** A package that is committed and remains active following a system reload.
- **Atomic Change:** Every packaging operation is contained within an atomic change. Atomic changes may contain multiple packaging operations. During an atomic change, any changes to install IOS XR software will not be visible to the system. To make the changes visible to the system, the atomic change must be applied.
- **Top-level package:** Each block of software has a top-level package and various partition-level packages. The top-level package can be installed or upgraded directly, whereas the partition-level packages cannot be changed directly. The partition-level packages are installed or upgraded automatically as dependencies of the top-level package. The top-level package has the name format `xr-<feature>-<release>.x86_64.rpm`, whereas the dependent partition-level packages have the longer name format containing information about the partition. You can also use the standard RPM commands to check the summary or description metadata of the package, which will identify whether it is a top-level or a partition-level package.
- **Package manager:** An entity that handles the semantics to resolve dependencies in packaging operations.
- **Packaging operations:** The actions performed to change the packages that are installed on the system. The semantics are inherited from the underlying package manager. Examples of packaging operations are upgrade, downgrade, replace, add, or remove packages.
- **Synchronous action:** Synchronous action requests are supported for install actions using CLI command. Specify `synchronous` keyword in the install commands, and the prompt will only be returned when either the request has completed, `Ctrl + C` keys are pressed or a reload occurs. Pressing `Ctrl + C` keys during a synchronous action request will return the prompt to the user but will not halt the install operation. During the synchronous action request, the user is updated with the status of the request whenever it changes.
- **Transaction:** All atomic changes occur within a transaction. If the system reloads during an install transaction, the running software will be reverted to its previous state before the transaction was started. To maintain the software changes carried out during a transaction, you must commit the transaction.

- A complete install operation to modify the system's software requires three phases:
  - Packaging operation
  - **Apply:** This is required to complete an atomic change and make the software change visible to the system.
  - **Commit:** This is required to end a transaction and ensure that all software changes will still be present on router reload.

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

This action clears all configuration rollback points available. You will not be able to rollback to, or view, any commits made until the install rollback event. Any new commits made after the install rollback event will start from commit ID '1000000001'.

Supported Packages

The base image is contained within an `ISO` file. Additional optional packages (RPMs) are provided as modular software deliverables to align with diverse use cases and their deployments across the network.

Note You can create a golden ISO (GISO) with optional packages and bug fixes based on your requirement. Contact Cisco Support to build a GISO.

- ISO image containing the base install image - `ncs540l-x64-7.0.1.iso`
- Tar file containing optional RPMs - `NCS540l-iosxr-7.0.1.tar`
- ZIP file for USB boot - `ncs540l-usb_boot-7.0.1.zip`

The software deliverables can be downloaded from Cisco Software Download center.

Note All Cisco IOS XR images are signed to ensure the authenticity of the software.

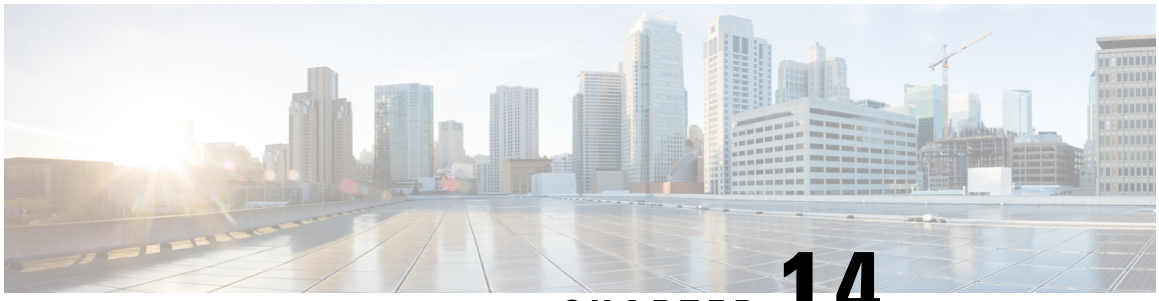
Optional Package	Included in ISO by Default
ncs5500-netflow	Yes
ncs5500-mcast	Yes
BGP	Yes

Optional Package	Included in ISO by Default
CDP	No
EIGRP	No
IPSLA	Yes
IS-IS	Yes
LLDP	Yes
MCAST	Yes
MPLS-OAM	Yes
Netflow	Yes
OSPF	Yes
Perfmgmt	Yes
RIP	No
Telnet	No
Track	Yes

Note Packages that are not included in the base ISO by default must be either added into a golden ISO, or installed separately.

SSH is part of the ISO image.

Install operation over IPv6 is not supported.



CHAPTER 14

Workflow for Installing Cisco IOS XR7 Software

The router is shipped with a pre-installed version of the Cisco IOS XR (XR7) software. When the router is powered ON for the first time, the pre-installed software starts functioning automatically. You configure the router for network capabilities. When a new version of the software is available, you can upgrade the system using these tasks:

Note For instructions to upgrade image-specific software, navigate to the CCO Software Download portal, select the product and refer to the `ncs5700-x64-<version>.docs.tar` file for the release.

- Obtain Data Models for Install Operation, on page 155
- Create a repository to install IOS XR SMUs or optional packages, on page 157
- Upgrade the System, on page 160
- Install optional packages for additional functionality, on page 163

Obtain Data Models for Install Operation

You can use YANG data models to install and upgrade XR7 software. The data models are packaged with the release image in the `/pkg/yang` directory.

The models are in the `.yang` format. Each data model can be identified as one of the following functionalities:

- `-oper` in the model name indicates an operational model. For example, `Cisco-IOS-XR-install-oper.yang` and `Cisco-IOS-XR-install-augmented-oper.yang` are operational models for the install operation.
- `-cfg` indicates a configuration model. For example, `Cisco-IOS-XR-install-cfg.yang` is a configuration model for the install operation.
- `-act` indicates an action model. For example, `Cisco-IOS-XR-install-augmented-act.yang` and `Cisco-IOS-XR-install-act.yang` are action models for the install operation.

Procedure

Step 1 View the install-related data models on the router. You can also access the supported data models to install Cisco IOS XR software from the Github repository.

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
-rw-r--r--. 1 root root 13726 Jul  2 02:04 Cisco-IOS-XR-install-oper-sub1.yang
-rw-r--r--. 1 root root  2440 Jul  2 02:04 Cisco-IOS-XR-install-augmented-oper-sub2.yang
-rw-r--r--. 1 root root 59866 Jul  2 02:04 Cisco-IOS-XR-install-augmented-oper-sub1.yang
```

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

Data Model	Description
Cisco-IOS-XR-um-install-cfg model	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 about packaging, atomic changes, and history of the install operation on the router
Cisco-IOS-XR-install-augmented-act	Action model to perform flexible install operations, including controlling the exact timing of system reloads and rolling back to a previous commit
Cisco-IOS-XR-shellutil-copy-act	Action model to copy files on the router from a source location

Step 2 Use a manageability agent like NETCONF or gRPC to connect and communicate with the router. You can send Remote Procedure Calls (RPC) requests to configure or retrieve operational data from the router. The router 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 container and leaf in the data model.

For more information about understanding the data model structure and using data models, see the *Programmability Configuration Guide for Cisco NCS 5500 Series Routers*.

Create a repository to install IOS XR SMUs or optional packages

Note If only Golden ISO (GISO) is used, you do not need to create a repository.

To install packages (RPM), code upgrades, and updates in XR7, you need a repository of RPMs for the router to download the RPMs for installation. The repository can be local to the router, or accessed remotely through FTP, HTTP, or HTTPS.

Important The repository must be created specific to each platform and release. Do not create repositories with a mix of platforms and releases.

When the repository is accessed remotely, you must provide a repository URL from where the install files are fetched. The URL contains:

- IP address of the server
- Port number of the server
- (Optional) Virtual Routing and Forwarding (VRF) name

The repository can be configured to be reachable using a non-default VRF table. If the repository is reachable through an address in a VRF, specify the name of the VRF.

The format of the repository URL is one of the following:

- FTP: `ftp://<server>[;<vrf>]/<path-to-repository>`
- HTTP: `http://<server>[;<vrf>]/<path-to-repository>`
- HTTPS: `https://<server>[;<vrf>]/<path-to-repository>`
- Local: `file:///<path-to-repository>`. The path to the repository must be under /harddisk:/ location.

For example, the URL for HTTP server is http://172.16.0.0:3333/.

Note Username and password are not supported for HTTP and FTP repositories.

Create and Configure a Local Repository

The router can serve as repository to host the RPMs. You must be a `root-lr` user with access to the router shell. Remote repository is the recommended method to access the RPMs. However, if remote repository is not your preferred option, then you can use the router as a repository to host the RPMs.

Using a local repository removes the need to setup an external server for software installation. In this method, the image files are copied directly to the router, and used to create a repository locally. However, on the downside, the files for future updates must be copied to each router individually.

This section provides the procedure for setting up a local RPM repository on the router.

Procedure

Step 1 Create a directory locally on the router's `/harddisk:`. Copy the required RPMs and ISO files (using copy or scp command) from the server to the local directory on the router.

Step 2 Access the shell of the router using `run` command and untar the RPMs.

Example:

```
Router#run
[node:~]$cd <directory-with-rpms>
[node:~]$tar -xvzf <rpm-name>.tgz
```

Step 3 Exit from the shell.

Step 4 Configure the local repository.

Example:

```
Router#config
Router(config)#install repository local-repo url file:///harddisk:/<directory-with-rpms>
Router(config)#commit
<data and time stamp> UTC: config[67543]: Configuration committed by user.
Router(config)#end
```

where, `local-repo` is the repository name, `file:///harddisk:/<directory-with-rpms>` is the local repository URL.

Step 5 Check the contents of the repository.

Example:

NCS 5700 Series:

```
Router#show install available
Trying to access repositories...
Package                Architecture          Version          Repository
--------------------  ---------------  ------------------ ---------------
xr-ncs5700-core       x86_67        7.3.1v1.0.1-1      local-repo
xr-core               x86_64        7.3.1v1.0.1-1      local-repo
```

Note
Only the top-level packages are displayed. The contents of the repository is displayed only when the configured repository is valid and the RPMs are present in the repository. It displays only the packages that are available in the repository and not part of active system.

Create and Configure an External Repository

To create an external repository, use a server that can be reached over HTTP, HTTPS or FTP. The following instructions are applicable to Linux distribution systems.

Using an external repository provides a central common repository to be used across devices. This eliminates the need to copy files for future updates to each router individually. It also serves as a single source when new RPMs (bug fixes, packages, updates) are made available. This is the recommended method to setup a repository.

Before you begin

Ensure that you have completed the following tasks:

- Set up your HTTP, HTTPS or FTP server. Ensure that the server is reachable as specified in the note above.
- Install `createrepo` utility on the Linux distribution system (if not installed already).

Procedure

Step 1 Create a directory on the server and copy all the RPMs to a directory. This directory hosts the repository and must be accessible to the HTTP, HTTPS or FTP server that the router will use to access the repository. For example, `/var/www/html`, is the directory where the repository will be created.

If the RPM files are archived (.tar format) or compressed (.tgz or .gz format), extract the files. The files hierarchically arrange in sub directories under the main directory that is used as a repository.

Step 2 Convert the directory to a repository using `createrepo` utility on the Linux server. This creates a directory named `repodata` with the metadata of all the RPMs.

Example:

```
[node]$createrepo --database /var/www/html/
Saving Primary metadata
Saving file lists metadata
Saving other metadata
Generating sqlite DBs
Sqlite DBs complete

[node]$cd /var/www/html/
[node]$ls
repodata
```

If you add new packages to the repository, change or remove packages from the repository, you must run `createrepo` command again to update the metadata. This ensures that the package manager chooses the correct packages.

Step 3 Configure the external repository.

Example:

```
Router#config
Router(config)#install repository remote-repo url http://10.194.88.104/<directory-with-rpms>
Router(config)#commit
<data and time stamp> UTC: config[67542]: Configuration committed by user 'cisco'.
Router(config)#end
```

For FTP, the repository is configured as follows:

```
Router#config
Router(config)#install repository remote-repo url ftp://10.194.88.104/<directory-with-rpms>
Router(config)#commit
<data and time stamp> UTC: config[67543]: Configuration committed by user 'cisco'.
Router(config)#end
```

where, `remote-repo` is the repository name, `http://10.194.88.104/<directory-with-rpms>` is the HTTP repository URL, and `ftp://10.194.88.104/<directory-with-rpms>` is the FTP repository URL.

Step 4 Verify connectivity to the server, and check the contents of the repository.

Example:

NCS 5700 Series:

```
Router#show install available
Trying to access repositories...
Package                Architecture          Version          Repository
--------------------  ---------------  ------------------ ---------------
xr-ncs5700-core       x86_64        7.3.1v1.0.1-1      local-repo
xr-core               x86_64        7.3.11v1.0.1-1     remote-repo
```

Note
Only the top-level packages are displayed. The contents of the repository are displayed only when the configured repository is valid and the RPMs with the updated metadata are present in the repository. It displays only the packages that are available in the repository and not part of active system.

Upgrade the System

You can upgrade the system when a newer version is available. A system upgrade replaces the existing version of the software with a newer version. By keeping the software up to date, you can ensure that the device works with the latest features and bug fixes.

You can replace the currently active software on your system with the software from a specified ISO image or GISO image. Only a minimal set of changes is performed to upgrade to the new software. Packages are not removed and reinstalled if they have the same name and version. For example, an upgrade that differs by only one package, removes and install only that one modified package.

Note The instructions in this section also apply to system downgrade.

This section shows replacing the current software version with `.iso` image. To understand the phases of install operation, see Software Deliverables and Terminologies, on page 151. For information about repositories, see Create a repository to install IOS XR SMUs or optional packages, on page 157.

Upgrade the current active version of Cisco IOS XR Software

This section shows replacing the current running software with an `.iso` image. This method must be used to change the IOS XR release version (upgrade or downgrade). It can also be used with golden ISO to keep the XR release version the same but change the installed SMUs and/or optional packages.

Upgrade the current active version

In this scenario, you replace the current software with `.iso` image, apply the changes, and commit the install operation. Committing the changes indicates the end of the current transaction. The updated software is used after the changes are applied, before the install transaction is committed.

A reboot is not always necessary. Bug fixes can be applied using a GISO that contains the current running software and the additional bugfix. These fixes may not require a reboot. Reboots are required for a system version change, if too many processes need restarting, there is configuration in the GISO to apply or if the bugfix is marked as requiring a reload.

From Cisco IOS XR Release 7.5.1, you can use the Automatic Install of Bridging Bug Fix RPMs feature to install any bridging bug fix RPMs that are prerequisite for a system upgrade or a downgrade.

You need to add the required Bridging Bug Fix RPMs into the customized ISO built using Cisco Golden ISO (GISO) build script **gisobuild.py**, by going through the steps in Golden ISO images, on page 167. Ensure that you add the required bridging bug fix RPMs into your Golden ISO. The GISO can include bridging Bug Fix RPMs for multiple releases, and installs only the specific bridging Bug Fix RPMs 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.
- The latest version has new prerequisite requirements that are not met by the earlier version.

The **install replace** command adds and applies the highest version of any bridging bugfix RPMs for the currently active XR version.

Procedure

Step 1 Copy the ISO (or GISO) image to the `/harddisk:` location on the router.

Step 2 Upgrade the system to replace the current software with the .iso image.

Example:

```
Router#install replace /harddisk:/ncs5700-x64.iso
```

Step 3 Activate the new .iso image on the router by applying the changes.

Example:

```
Router#install apply {reload | restart} [noprompt]
```

Note
For operations that do not change the XR release version (SMU or optional package install), you can control the timing of reload or process restart by using **install package replace** *iso* followed by **install apply {reload | restart } [noprompt]**.

To identify whether a reload is required or only process restart is needed, use either **show install history last transaction verbose** command or **show install request** command.

Include the keyword `noprompt` in the command to enable the system to bypass your permission to reload the router.

Applying the change gives you the flexibility to test the operation of the new software before committing the changes. If you reload the router, the router reverts the software to its previous software state.

All operations that automatically apply the new software are prohibited when an atomic change is already in progress. You must address the current atomic-change before performing this operation. To address the change, apply the current atomic-change, or cancel it with the **install package abort all-since-apply** command.

Step 4 View the install log.

Example:

```
Router#show install log detail
Fri Nov 12 09:43:49.702 UTC
2021-11-12 09:33:47 UTC    Transaction 1 started
2021-11-12 09:33:47 UTC    Atomic change 1.1 started
2021-11-12 09:33:47 UTC    Packaging operation 1.1.1 started
2021-11-12 09:33:47 UTC    Replace
```

```
2021-11-12 09:35:58 UTC    Packaging operation 1.1.1 success
2021-11-12 09:36:04 UTC    Apply by reload started
2021-11-12 09:38:48 UTC    Atomic change 1.1 successfully applied by reload
```

Step 5 Verify that the image is activated successfully.

Example:

```
Router#show install request
```

Step 6 Commit the transaction.

Example:

```
Router#install commit
```

Note
Any action requests may be run synchronously from the CLI. During this request, you are updated with the status of the request whenever it changes. The following example shows the output from a synchronous action request:

```
Router#install commit synchronous
Starting:
  install commit
Transaction 1
The install operation will continue in the background
Press Ctrl-C to return to the exec prompt. This will not cancel the install operation

Current activity: Initializing ....
Current activity: Commit transaction ......

Transaction 4: 'install commit' completed without error
```

Upgrade QDD Optical Modules Through CLI

The QDD optics firmware file needs to be copied to the router manually. Contact Cisco Support to check the QDD firmware version, IOS XR release compatibility, and to obtain the QDD optics firmware file.

Limitation: When ports share a common management interface, IOS XR serializes the firmware upgrade. Serializing and deserializing may delay the upgrade process.

Procedure

Step 1 From the router, copy the QDD firmware file to the hard disk using the following command:

Example: `scp user@10.1.1.1:/home/user/filename harddisk:/`

- When you are using VRF, use the following sample command:

```
scp user@10.1.1.1:/home/user/cl1.bin vrf MGMT harddisk:/

Tue Jan 25 02:57:22.762 UTC
Connecting to 10.1.1.1...
Password:
  Transferred 1484800 Bytes
  1484800 bytes copied in 0 sec (22161194)bytes/sec

RP/0/RP0/CPU0:8808#dir harddisk:/cl1.bin
Tue Jan 25 03:00:47.835 UTC
```

```
Directory of harddisk:/cl1.bin
35 -rw-r--r--. 1 1484800 Jan 25 02:57 dp04qsdd_dp04sfp8_161_10_01.ackit

53461500 kbytes total (42983204 kbytes free)
```

- When you are not using VRF, remove the `vrf MGMT` command:

```
scp user@10.1.1.1:/home/user/cl1.bin harddisk:/
```

Step 2 Run the following commands to upgrade the FPD for QDD optical modules:

Multiport upgrade:`upgrade optics port 0,1,2,3,4 filename /harddisk:/cl1.bin location 0/1/CPU0`

Single port upgrade: `upgrade optics port 0 filename /harddisk:/cl1.bin location 0/1/CPU0`

You can check the firmware upgrade progress using the following command: `show optics firmware upgrade port 0,1,1,2,3,4 location 0/1/CPU0`

Install optional packages for additional functionality

You can install one or more packages (RPM) that are not already present on the system. The packages are not mandatory for the software to function, but provide additional functionality. Based on your requirement, you can install or remove these optional packages. The source file can be a repository name, repository url, local filepath, or path to a tar file.

You must specify only the top-level package name that you want to install. The associated dependencies of this package, in the form of card and partition-specific packages, are included automatically. By default, the latest available version of each package is installed. You can also explicitly install a specific version of a package.

This example shows the options to install the optional package `xr-telnet-7.3.17.5.2v1.0.1-1.x86_64.rpm`.

Procedure

Step 1 Install one or more optional packages using one of the following options:

- **Option 1:** Install the package from the local directory:

  NCS 5700 Series:

  ```
  Router#install source /harddisk:/files xr-telnet-7.3.1v1.0.1-1.x86_64.rpm
  ```

  Note
  The `install source` command automatically applies the changes. Use this command to install optional packages. To upgrade existing packages, see Upgrade the System to Obtain Bug Fixes, on page 169.

- **Option 2:** Install the package from a configured remote repository:

  ```
  Router#install source install-repo xr-telnet
  ```

  Here, `install-repo` is the name of the repository. For repository configuration, see Create a repository to install IOS XR SMUs or optional packages, on page 157.

- **Option 3:** Install the package from a repository URL:

```
Router#install source http://72.16.0.0:3333/remote-repo xr-telnet
```

- **Option 4:** Add the package and apply the change. The package must be available in the repository.

  NCS 5700 Series:

```
Router#install package add xr-telnet-7.3.1v1.0.1-1.x86_64.rpm
Router#install apply {restart | reload}
```

More than one package can be installed using a single packaging operation. Use the following command:

```
Router#install source <path-to-source> <package 1> <package 2> … <package n>
```

For example,

NCS 5700 Series:

```
Router#install source /harddisk:/files xr-telnet-7.3.1v1.0.0-1.x86_64.rpm
xr-mcast-7.3.1v1.0.0-1.x86_64.rpm
```

To perform this task using data models, use the `install` RPC on the Cisco-IOS-XR-install-act data model. Here is an example with an HTTP repository:

```
<install>
   <packages>
     <packagename>pkg1</packagename>
     <packagename>pkg2</packagename>
     …
     <packagename>pkgn</packagename>
   </packages>
   <source-type>http</source-type>
   <source><path-to-source></source>
</install>
```

Step 2 Commit the operation.

Example:

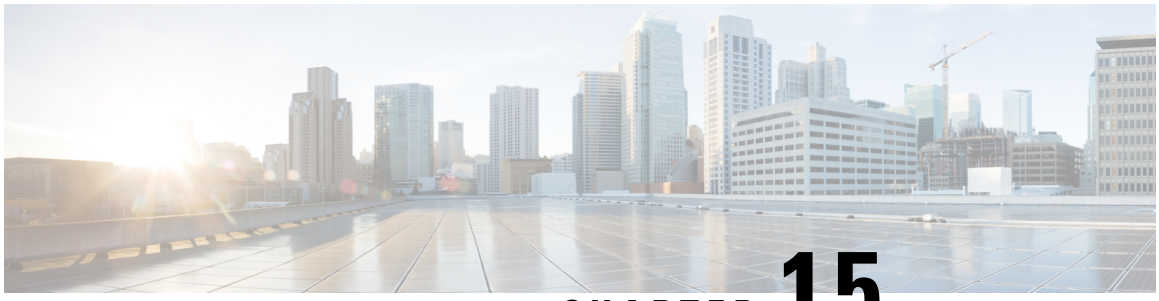
```
Router#install commit
```

Step 3 Check the status of install operation.

Example:

NCS 5700 Series:

```
Router# show install history last package
Tue Jul 14 11:09:19.748 UTC
2021-07-14 11:08:12 UTC        Packaging operation 1.1.1 started
2021-07-14 11:08:12 UTC        Add
2021-07-14 11:08:12 UTC        xr-telnet
2021-07-14 11:09:10 UTC        Packaging operation 1.1.1 success

Location 0/RP0/CPU0
   Add xr-telnet-7.3.1v1.0.0-1.x86_64
   Add xr-telnet-36cad6c174d48ffe-7.3.1v1.0.0-1.x86_64
   Add xr-telnet-ncs5700-7.3.1v1.0.0-1.x86_64
   Add xr-telnet-a1f05ad3091205a8-7.3.1v1.0.0-1.x86_64
```



CHAPTER 15

Additional Install Operations

After you upgrade your system, based on your requirement, you can perform additional install operations:

- View the Version of Installed Packages, on page 165
- Golden ISO images, on page 167
- Upgrade the System to Obtain Bug Fixes, on page 169
- Remove a bug fix from Cisco IOS XR software, on page 174
- Roll back software to a previously saved installation point , on page 174
- Caveats, on page 177

View the Version of Installed Packages

The router is shipped with a pre-installed operating system. You can view the version of the installed software and the active packages. If you have upgraded your system, installed additional packages or bug fixes, you can view the version of the committed packages.

Review the software version information:

- Package name and version
- User who built the package
- Time the package was built
- Build workspace
- Build host
- ISO label:
  - Label is present if GISO boots using PXE boot
  - Label is present if GISO is installed using the `install replace` method
  - Label reverts to default (only release version) if there is any change since the time the image with the label was installed.
  - Label is nullified and reverts to default if an RPM is added or removed on top of an existing GISO.
  - Label is repopulated if an RPM from the GISO is added or removed and a rollback operation is performed.

- Copyright information
- Hardware information

Procedure

Step 1 View the version of the Cisco IOS XR software, and its various software components that are installed on the router.

Example:

The following example shows the version information for a non-GISO image:

NCS 5500 Series:

```
Router#show version
Cisco IOS XR Software, Version 7.3.1 LNT
Copyright (c) 2013-2022 by Cisco Systems, Inc.
Build Information:
Built By     : xyz
Built On     : Tue June 07 19:43:44 UTC 2021
Build Host   : iox-lnx-064
Workspace    : ../ncs5700/ws
Version      : 7.3.1
Label        : 7.3.1

cisco NCS5700 (D-1563N @ 2.00GHz)
cisco NCS-57B1-5DSE-SYS (D-1563N @ 2.00GHz) processor with 32GB of memory
ios uptime is 3 weeks, 1 day, 10 hours, 11 minutes
NCS-57B1-5DSE-SYS Fixed Scale HW Flexible Consumption Need Smart Lic
```

The following example shows the version information for a GISO image. The customer label is appended to the `Label` field in the GISO image:

NCS 5500 Series:

```
Router#show version
Cisco IOS XR Software, Version 7.3.1 LNT
Copyright (c) 2013-2022 by Cisco Systems, Inc.
Build Information:
Built By     : xyz
Built On     : Tue June 07 19:43:44 UTC 2021
Build Host   : iox-lnx-064
Workspace    : ../ncs5700/ws
Version      : 7.3.1
Label        : 7.3.1-Customer_Label

cisco NCS5700 (D-1563N @ 2.00GHz)
cisco NCS-57B1-5DSE-SYS (D-1563N @ 2.00GHz) processor with 32GB of memory
ios uptime is 3 weeks, 1 day, 10 hours, 11 minutes
NCS-57B1-5DSE-SYS Fixed Scale HW Flexible Consumption Need Smart Lic
```

You can also use the `get` RPC on the `install.version` data model.

Step 2 View the active packages.

Example:

NCS 5700 Series:

```
Router#show install active summary
Fri Mar  5 17:37:23.205 UTC
Active Packages:    XR: 156    All: 1214
```

```
Label:              7.4.1-LABEL
Software Hash:      28dd70ef227aeca3d3fd3ecf8d1792a4f51fabb299ec7d38725869575fd9cfaf
Optional Packages                                                             Version
------------------------------------------------------------------------------------
xr-bgp                                                                    7.3.1v1.0.0-1
xr-cdp                                                                    7.3.1v1.0.0-1
xr-eigrp                                                                  7.3.1v1.0.0-1
xr-ipsla                                                                  7.3.1v1.0.0-1
xr-is-is                                                                  7.3.1v1.0.0-1
xr-lldp                                                                   7.3.1v1.0.0-1
xr-mcast                                                                  7.3.1v1.0.0-1
xr-mpls-oam                                                               7.3.1v1.0.0-1
xr-ncs5700-mcast                                                          7.3.1v1.0.0-1
xr-ncs5700-netflow                                                        7.3.1v1.0.0-1
xr-netflow                                                                7.3.1v1.0.0-1
xr-ospf                                                                   7.3.1v1.0.0-1
xr-perf-meas                                                              7.3.1v1.0.0-1
xr-perfmgmt                                                               7.3.1v1.0.0-1
xr-telnet                                                                 7.3.1v1.0.0-1
xr-track                                                                  7.3.1v1.0.0-1
xr-xcare                                                                  7.3.1v1.0.0-1
```

You can also use the `get` RPC on the `install.packages.active.node` and the `install.packages.active.node.summary` data models.

To understand the data model structure and its arguments, see the Obtain Data Models for Install Operation, on page 155 topic.

Golden ISO images

The Golden ISO (GISO) image is a custom-built IOS XR software image that

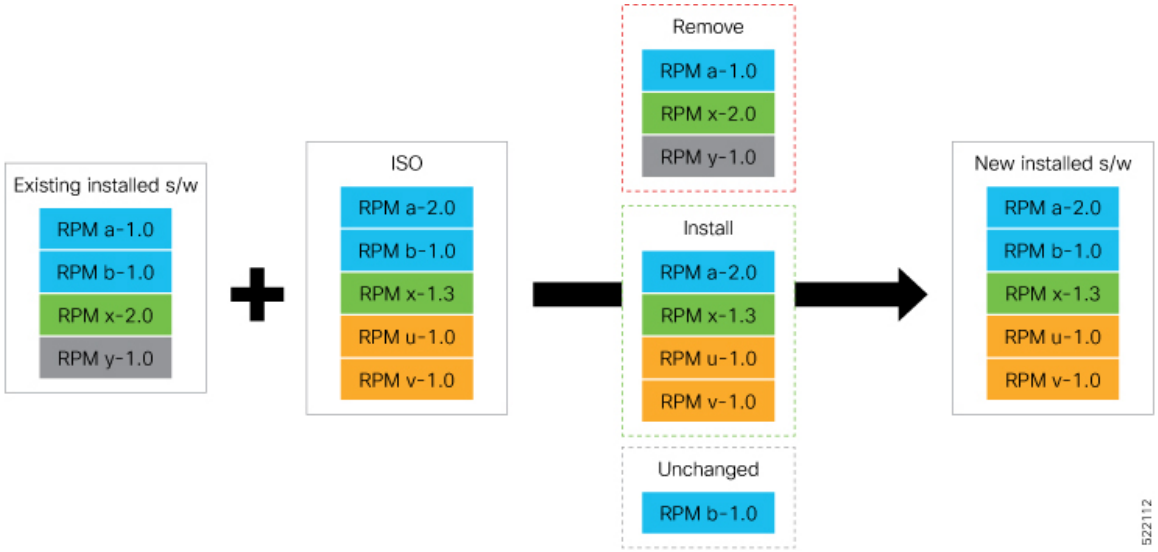
- is based on a given IOS XR version
- contains all mandatory packages
- includes optional packages and SMUs requested by the GISO image owner, and
- enables automated router boot using iPXE or system upgrade to a specified software version in a single operation.

Note When switching between Golden ISOs (GISOs) built on the same base version of IOS-XR, the **install replace** command automatically apples only the minimum required changes such as package additions, removals, and upgrades required for the transition.

Table 9: Feature History Table

Feature Name	Release Information	Description
Build a Golden ISO image using `gisobuild.py` tool	Release 7.6.1	This feature facilitates faster deployment by allowing you to build your GISO image without the need to contact Cisco support. You can now select the install files, add your RPMs, repackage them as a custom image, and install the image. In previous releases, you were required to contact Cisco to have your GISO built.

Figure 11: Golden ISO build process



Build a Golden ISO image

This section provides the options that help you build a GISO image.

Procedure

Step 1 Build a GISO image using one of the following options:

- **Option 1:** Contact Cisco Support to build the GISO image with the set of packages based on your requirement
- **Option 2:** Use the `gisobuild.py` tool to build the GISO. For information about the tool and how to use it, see https://github.com/ios-xr/gisobuild.

Step 2 Execute this command to install the GISO image built from previous step similar to installing any other ISO image.

Example:

```
Router# install replace /harddisk:/ncs540l-giso-x64.iso
```

Upgrade the System to Obtain Bug Fixes

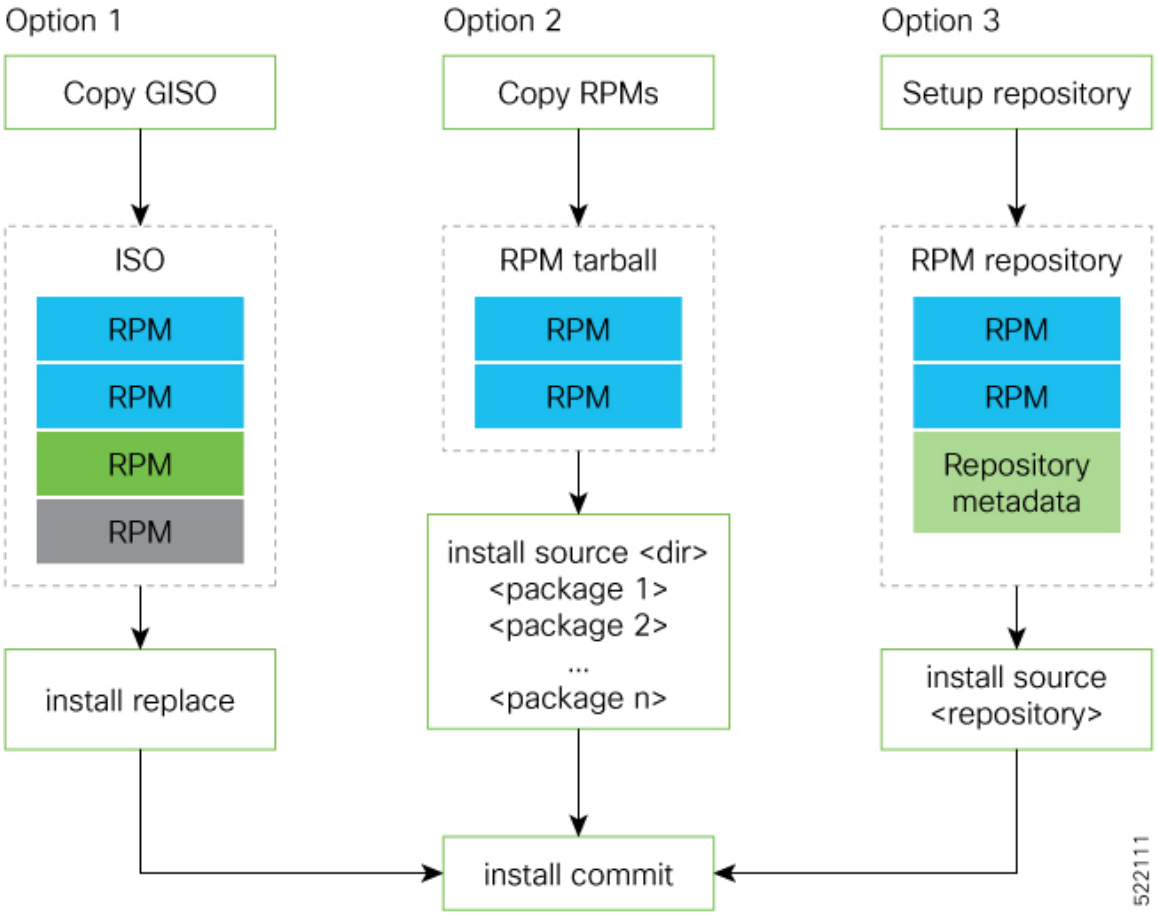
You can upgrade the system to obtain all available bug fixes or choose specific bug fix using bug ID.

Bug fixes are available as optional RPMs. The bug fixes replace packages in the base ISO without adding up to the image size. For example, even if you add 20 RPMs to the GISO, the GISO file size will remain the same as the initial ISO.

You can download the bug fix RPMs from the Cisco Software Download portal .

The `README` file provides the relevant information about the bug fix and also identifies the dependencies, if any, where other bug fix RPMs may be required for a complete fix.

The following image shows the options to install the bug fix RPMs.



The software is split into modular blocks and the package manager infrastructure computes the dependencies between the blocks. Each block of software has a top-level package and various partition-level packages. Bug

fixes that span multiple blocks may lead to creating multiple dependent packages that are built as part of earlier bug fixes.

Note We recommend that you leverage the GISO workflow. Contact Cisco Support to build a new GISO with the required bug fixes. The RPMs are present in the initrd, which is the initial RAM disk for the boot loader, and this requires that the package is signed by Cisco. You can install GISO using a single `install replace` operation. For more information about GISO, see the *Build a Golden GISO* section.

However, if you do not prefer using the GISO, here are a few alternative ways to install bug fixes:

- Create tarballs to install one or more bug fixes. For example, if you are installing bug fixes CSCxx11111, CSCyy22222 and CSCzz33333, you can use the individual tarball files and create a single tarball file.
- Use a Dandified Yum- or DNF repository to install, update, or remove relevant bug fixes. See, Create a repository to install IOS XR SMUs or optional packages, on page 157.

Note Use the RPM repository to harvest the benefits of package manager. The package manager queries the available packages, and downloads only those packages and their dependencies that are needed for installation.

Procedure

Step 1 View the list of available bug fixes.

Example:

NCS 5700 Series:

```
Router#show install fixes available
Bug Id            Packages                       Repository
-----------    ----------------------------  ----------------
CSCxx12345     xr-5700-core-7.3.1v1.0.1-1      <repository-name>

               xr-core-7.3.1v1.0.1-1           <repository-name>
```

Step 2 Install the bug fix or package using one of the following options:

- Install the package where the bug fix is applied.

  NCS 5700 Series:

  ```
  Router#install package upgrade xr-5700-core-7.3.1v1.0.1-1 xr-core-7.0.1v1.0.1-1
  Packaging operation 1.1.1 started - xr-5700-core-7.3.1v1.0.1-1 xr-core-7.0.1v1.0.1-1
  ```

  This task can also be accomplished using Cisco-IOS-XR-install-augmented-oper data model.

  Apply the changes.

  ```
  Router#install apply [reload | restart]
  ```

  Note
  To identify if you must reload or restart the system while applying the changes, use one of these two methods:

  - History of last transaction

```
Router#show install history last transaction verbose
2019-09-11 17:01:46 UTC    Transaction 3 started
2019-09-11 17:01:46 UTC      Atomic change 3.1 started
2019-09-11 17:01:46 UTC      Packaging operation 3.1.1 started
2019-09-11 17:16:46 UTC    Transaction 3 complete

Least impactful apply method: process restart
```

The command also displays the information about the changes to files and processes because of the install operation, and the package operations carried out on each node.

- Show install request

```
Router:#show install request
Wed Sep 14 02:53:21.525 PDT

User request: install package abort latest
Operation ID: 2.1.2
State:        Failure since 2022-09-14 02:48:15 UTC-07:00
Disk space check failed on nodes: 0/0/CPU0. Query 'show install history id 2.1.2 errors' for
 more details and next steps

Current activity:    Await user input
Time started:        2022-09-14 02:48:20 UTC-07:00

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

NCS 5700 Series:

```
Router#show install request
User request: install package upgrade xr-5700-core-7.3.1v1.0.1-1
Operation ID: 2.1.1
State: Success

Current activity: Await user input
Time started:

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

Here, both `install apply restart` and `install apply reload` options are available. In this case, use `install apply restart` command because the impact on the system is the least. But when only an `install apply reload` option is available, then reload is the only option to apply the change.

- Install the optional package. Changes are applied automatically.

Attention
Automatic change may trigger a reload of the router depending on the package that is installed.

```
Router#install source /harddisk:/files xr-<platform>-core-7.3.1v1.0.1-1
```

Note
Packages can also be installed using the package name.

```
Router#install source /harddisk:/files  xr-<platform>-core
```

- This task can also be performed using YANG data models. Use `install` RPC on the Cisco-IOS-XR-install-act data model. Here is an example usage with an HTTP repository:

```
<install>
   <packages>
     <packagename>pkg1</packagename>
   </packages>
   <source-type>http</source-type>
   <source><path-to-source></source>
</install>
```

Step 3 View the state of the packaging operation.

Example:

```
Router#show install request
User request: install package upgrade xr-<platform>-core-<version> xr-core-<version>
Operation ID: 2.1.2
State:        In progress since

Current activity:    Initiate operation
Next activity:       Begin transaction
Time started:        2019-06-25 07:41:06

No per-location information.
```

Step 4 View the log to ensure that the installation is successful.

Example:

```
Router#show install log
2019-06-25 07:41:06 UTC    Transaction 1 started
2019-06-25 07:45:08 UTC    Upgrade (Success)
2019-06-25 07:45:08 UTC    xr-<platform>-core-<version>
2019-06-25 07:45:08 UTC    xr-core-<version>
2019-06-25 07:57:02 UTC    Atomic change 1.1 successfully applied by reload
```

Step 5 View the history of the install operation.

Example:

```
Router#show install history table
  Transaction             Atomic Change           Packaging Operations
---------------------  -------------------------  ---------------------------------
Id   Status           Id  Method   Status      Id  Operation   Inputs Status
---------------------  -------------------------  ---------------------------------
 1   In progress      1   Reload   Success      1   Upgrade        1  Success
```

The command can also be used to view more details if there is a failed operation.

Use **show install history id <operation-id>** command to filter the history of install information by ID. IDs are of the form `<transaction id>.<atomic id>.<packaging id>`.

```
Router#show install history id ?
  WORD Specify an operation ID (e.g. 1, 1.2, 1.2.3)
```

Use **show install history last** command to view the last packaging operation, atomic change, or transaction.

```
Router#show install history last ?
     atomic-change  Show the last atomic change
     package        Show the last packaging operation
     transaction    Show the last transaction
```

Step 6 After the operation is complete, verify that the packages `xr-5700-core-7.3.1v1.0.1-1` and `xr-core-7.3.1v1.0.1-1and xr-core-7.5.2v1.0.1-1` are installed and active.

Example:

Example:

NCS 5700 Series:

```
Router#show install active summary
Fri Mar  5 17:37:23.205 UTC
Active Packages:    XR: 156    All: 1214
Label:              7.4.1-LABEL
Software Hash:      28dd70ef227aeca3d3fd3ecf8d1792a4f51fabb299ec7d38725869575fd9cfaf
Optional Packages                                                          Version
-------------------------------------------------------------------------------
xr-bgp                                                          7.3.1v1.0.0-1
xr-cdp                                                          7.3.1v1.0.0-1
xr-eigrp                                                        7.3.1v1.0.0-1
xr-ipsla                                                        7.3.1v1.0.0-1
xr-is-is                                                        7.3.1v1.0.0-1
xr-lldp                                                         7.3.1v1.0.0-1
xr-mcast                                                        7.3.1v1.0.0-1
xr-mpls-oam                                                     7.3.1v1.0.0-1
xr-ncs5700-mcast                                                7.3.1v1.0.0-1
xr-ncs5700-netflow                                              7.3.1v1.0.0-1
xr-netflow                                                      7.3.1v1.0.0-1
xr-ospf                                                         7.3.1v1.0.0-1
xr-perf-meas                                                    7.3.1v1.0.0-1
xr-perfmgmt                                                     7.3.1v1.0.0-1
xr-telnet                                                       7.3.1v1.0.0-1
xr-track                                                        7.3.1v1.0.0-1
xr-xcare                                                        7.3.1v1.0.0-1
```

Example:

```
Router#show install active summary
```

The version has changed. The version `1.0.1-1` indicates that the bug fix is installed.

This task can also be accomplished using data models. Use the `get` RPC for `install.fixes.active` operation using Cisco-IOS-XR-install-augmented-oper data model.

Step 7 Commit the changes for the changes to persist after a reload operation.

Example:

```
Router#install commit
```

Step 8 View the list of bug IDs for which fixes are committed.

Example:

```
Router#show install fixes committed
```

This task can also be accomplished using data models. Use the `get` RPC for `install.fixes.committed` operation using Cisco-IOS-XR-install-augmented-oper data model.

Step 9 View the list of active bug fix RPMs.

Example:

```
Router#show install fixes active
```

This task can also be accomplished using data models. Use the `get` RPC for `install.fixes.active` operation using Cisco-IOS-XR-install-augmented-oper data model.

Remove a bug fix from Cisco IOS XR software

You can remove a bug fix that is causing issues or is no longer needed.

Before you begin

Warning Bug fixes that are prerequisites for other fixes can only be removed if the dependent bug fixes are removed.

Procedure

Use one of the following options to remove a bug fix from your IOS XR software version:

- If the bug fix to be removed was the most recently installed, and the device supports the **install rollback** feature, roll back using rollback ID
  1. Identify the rollback ID for the bug fix you want to remove.
  2. Use the rollback process to return your software to a previous stable version using the rollback ID.
- Build a custom Golden ISO (GISO) image
  1. Use the `gisobuild.py` tool to create a custom GISO image that excludes the unwanted bug fix and includes any required fixes.
  2. Install the custom GISO image to update your system.

The bug fix is removed from your system using the selected method.

Roll back software to a previously saved installation point

You can roll your system software back to a previous version. This could be used to discard an ongoing install operation, or to undo an install operation that has already been committed. After each commit operation, the system saves a record of the committed software packages. Each record is a restoration point, and is assigned a unique ID. This ID is known as a rollback ID. You can use the rollback ID to roll back the software to a

restoration point associated with this ID. Up to 900MB of space is allowed for rollback points, instead of a specific number of rollback points.

Table 10: Feature History Table

Feature Name	Release Information	Feature Description
Roll back to a previously saved installation point	Release 7.4.1	A maximum of 900MB of disk space is used to store all rollback points. You can delete older rollback points to make space for newer or smaller rollback points such as, SMUs. This feature introduces the following commands: • **install commit description**—sets the description for a rollback ID • **install label id** *id* **description** *description*—adds a new description or overwrites the description of the specified rollback ID. The following command is updated to include the keyword, **oldest** *n*: • **clear install rollback oldest** *n*—deletes the specified number of rollback points

Note

- Use rollback ID 0 to roll back to the software that was present after the system booted for the first time.
- If you commit an install transaction using the **install commit** command, the GISO ZTP configuration is saved along with the rest of the software changes. This means that if you use the **install rollback** or **install package rollback** command to revert the software to the state of a previous transaction, the GISO ZTP configuration is also reverted to its previous state. To undo this install operation, as well as the change in GISO ZTP configuration, use the **install package abort** command. If there is no GISO ZTP configuration saved at the end of the install transaction, the existing GISO ZTP configuration is reverted to the previous state when the rollback operation of that transaction is applied.

Roll back Cisco IOS XR software using a rollback ID

This procedure helps you rollback your software using a rollback ID.

Procedure

Step 1 View the list of available rollback IDs.

Example:

```
Router# show install rollback list-ids
```

Step 2 Explore the main packages that can be installed if you roll the software back to the specific rollback ID.

Example:

```
Router# show install rollback id <id>
```

Step 3 View the relative changes that are made to the currently installed software if it is rolled back to a rollback ID.

Example:

```
Router# show install rollback id <id> changes
```

To perform these tasks using data models, use the `get` RPC on the Cisco-IOS-XR-install-augmented-oper data model.

```
<rpc>
  <get>
   <filter type="subtree">
      <install xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-install-augmented-oper">
      <rollback/>
     </install>
   </filter>
  </get>
</rpc>
```

Step 4 Roll back to the software associated with the specific rollback ID.

Example:

```
Router# install rollback <id> [commit]
```

If you want to apply the change and roll back to the associated rollback ID, commit the change. You can also include the keyword `noprompt` in the command to enable the system to bypass your permission to reload the router.

Attention
This roll back operation installs the previous software and also applies the change automatically. This may reload the router depending on the package that is rolled back.

Alternatively, use the **install package rollback** command to only roll back the package but not apply the changes. You can check whether the router will reload or restart if you apply the change using the **show install history last transaction verbose** command or **show install request** command. Based on the command output, you can take the appropriate action using the **install apply reload | restart** command to either reload or restart the system. Use the **install commi**t command to commit the transaction.

To perform this task using data models, use the `install-rollback` RPC on the Cisco-IOS-XR-install-augmented-oper data model.

```
<rpc>
 <install-rollback xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-install-augmented-act">
  <commit>true</commit>
   <transaction-id>0</transaction-id>
```

```
 </install-rollback>
</rpc>
```

To understand the data model structure and its arguments, see the Access the Install-Related Data Model.

Step 5 Commit the operation.

Example:

```
Router#install commit
```

Caveats

Some IOS XR releases have changes to the upgrade procedure due to new features and enhancements in these releases. The information in this section summarises the key changes. To view supported upgrade paths and caveats for any release upgrade path, use the **show install upgrade-matrix** commands.

Release 7.10.1

Table 11: Upgrade caveats

From	To	Bridge SMUs required	caveats
7.5.2	7.10.1 and later	Yes	1*, 2*
7.6.1	7.10.1 and later	Yes	1*, 2*
7.6.2	7.10.1 and later	None	1*
7.7.1	7.10.1 and later	None	1*
7.7.2	7.10.1 and later	None	1*
7.8.1	7.10.1 and later	None	1*
7.8.2	7.10.1 and later	None	1*
7.9.1	7.10.1 and later	None	1*
7.9.2	7.10.1 and later	None	1*

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

2*: Ensure that a reload bridging SMU (CSCwd71524) is installed.

Note CSCwd71524:

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

These downgrade caveats are applicable for Release 7.10.1 and later.

Table 12: Downgrade caveats

From	To	Target SMUs required	Caveats
7.10.1 and later	7.5.2	Yes	***, A*, B*
7.10.1 and later	7.6.1	Yes	***, A*, B*
7.10.1 and later	7.6.2	Yes	***, A*, B*
7.10.1 and later	7.7.1	Yes	***, A*, B*
7.10.1 and later	7.7.2	Yes	***, A*, B*
7.10.1 and later	7.8.1	Yes	***, A*, B*
7.10.1 and later	7.8.2	Yes	***, A*, B*
7.10.1 and later	7.9.1	Yes	***
7.10.1 and later	7.9.2	Yes	***

***

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

- You must regenerate crypto keys and certificates after a downgrade.
- PXE recovery is required if the image downgrading isn’t bootable.

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

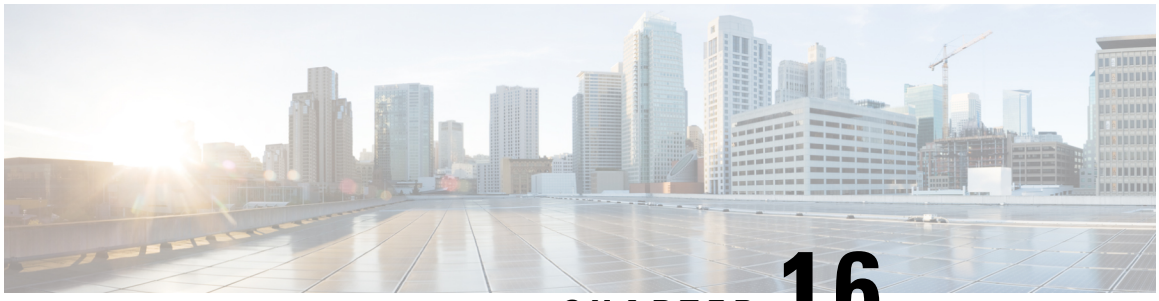
B*: You must recover the router through PXE if a power cycle occurs during the downgrade.

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

PART III

System Hardware Profiles

- NCS5700 MDB Profile, on page 183



CHAPTER 16

NCS5700 MDB Profile

- Hardware MDB Profiles for Layer 2 and 3 Services, on page 183
- Configure Hardware MDB Profiles for Layer 2 Services, on page 186

Hardware MDB Profiles for Layer 2 and 3 Services

Table 13: Feature History Table

Feature Name	Release	Description
Hardware MDB profiles for Layer 2 and Layer 3 services overlay scale increased to 32000	Release 25.2.1	Introduced in this release on: NCS 5700 fixed port routers, NCS 5700 line cards [Mode: Compatibility; Native] You can now configure hardware MDB profiles for Layer 2 and Layer 3 services together, over SRv6 underlay, enabling you to set overlay scales to 32000 in terms of the number of routers and sessions. This release adds the following hardware MDB profiles to the **hw-module profile mdb** command: • **l2max-srv6** • **l2max-se-srv6** • **l3max-srv6** • **l3max-se-srv6**

Feature Name	Release	Description
Hardware MDB Profiles for Layer 2 Services	Release 7.6.1	You can now configure hardware modular database (MDB) profiles for Layer 2 services, enabling you to achieve higher scales in terms of the number of routers and sessions. In earlier releases, you could configure hardware MDB profiles for Layer 3 services. This release adds the following hardware MDB profiles to the **hw-module profile mdb** command: • **l2max** • **l2max-se**

Hardware MDB profile is used to modify router resources for the specific needs during the router boot up time. You can configure the hardware MDB profile or you can view the default profile. The hardware MDB profiles are applicable for base and scale line cards of fixed and modular Cisco NCS 5700 Series Routers.

Hardware MDB profile feature was introduced in Cisco IOS XR software Release 7.4.1 with **l3max** and **l3max-se** configuration profile options.

Starting from Cisco IOS XR software Release 7.6.1, it supports **l3max**, **l3max-se**, **l2max**, and **l2max-se** profiles, and provides high scale layer 2 and layer 3 services. Router with base line cards supports **l2max** and **l3max** profiles, and routers with scale line cards support **l2max-se** and **l3max-se** profiles.

All features that are supported on **l3max** or **l3max-se** hardware profile are also supported on **l2max** or **l2max-se** hardware profile and the features that are supported on **l2max** or **l2max-se** hardware profile are also supported on **l3max** or **l3max-se** hardware profile respectively.

Use the **hw-module profile mdb** command in the Global Configuration mode to configure the hardware MDB profile options.

Modular routers boots up in the Cisco NCS 5700 Series compatibility mode by default. You can configure the router to boot up in native mode and with the desired MDB profile.

Following tables provides the base and scale profiles of MDB for Cisco NCS 5700 Series Routers and Line Cards.

Table 14: MDB Hardware Module Profile for Fixed System of Cisco NCS 5700 Series Routers

Base Profile	Scale Profile
• Default profile for Cisco NCS 5700 Series Base Router is **l3max**. • Use **hw-module profile mdb l2max** command to configure **l2max** profile.	• Default profile for Cisco NCS 5700 Series Scale Router is **l3max-se**. • Use **hw-module profile mdb l2max-se** command to configure **l2max-se** profile.

Table 15: MDB Hardware Module Profile for Modular System of Cisco NCS 5700 Series Routers and Line Cards

Type	Base Profile	Scale Profile
Description	Configure NCS 5700 MDB profile for the system which has the following line cards: • Cisco NCS 5700 Series base line cards alone. • Cisco NCS 5700 Series base and Cisco NCS 5700 Series scale line cards. • Cisco NCS 5700 Series scale line cards alone. **Note** Recommended the configuration of **l3max-se** and **l2max-se** only when Cisco NCS 5700 Series scale line cards are present.	Configure MDB scale profile for the system which has Cisco NCS 5700 Series scale line cards.
NCS5700 MDB Profile	• Ensure that the system has either base or scale Cisco NCS 5700 Series line cards. • Command to configure Cisco NCS 5700 Series native mode is **hw-module profile npu native-mode-enable**. • Command to configure **l3max** MDB profile is **hw-module profile mdb l3max**. System loads with native mode and **l3max** MDB profile in a single reload. • Command to configure **l2max** MDB profile is **hw-module profile mdb l2max**. System loads with native mode and **l2max** MDB profile in a single reload. **Note** Even if the system has only scale line cards, the default profile is **l3max**. Other profiles can be configured as specified in the Configuration Example section.	• Ensure that the system has Cisco NCS 5700 Series scale line cards. • Command to configure Cisco NCS 5700 Series native mode is **hw-module profile npu native-mode-enable**. • Command to configure **l3max-se** MDB profile is **hw-module profile mdb l3max-se**. System loads with native mode and **l3max-se** MDB profile in a single reload. • Command to configure **l2max-se** MDB profile is **hw-module profile mdb l2max-se**. System loads with native mode and **l2max-se** MDB profile in a single reload.

Note

- You must reload the router to activate the newly configured MDB profile.
- The default profile for a NCS-57C3-MOD-SYS (base) router is **l3max**. As per the requirements, you can configure it to **l2max** profile.
- The default profile for a NCS-57C3-MOD-SE-S (scale) router is **l3max-se**. As per the requirements, you can configure it to **l2max-se** profile. Even though base profiles can be configured on SE systems, it is not recommended to use low scale profile.

Restrictions

The following hardware doesn't support hardware MDB profiles.

- Cisco NCS 5500 Series line cards
- Cisco NCS 5700 Series line cards operating in compatible mode.

Configure Hardware MDB Profiles for Layer 2 Services

Router supports configuration of hardware MDB profiles for layer 2 services in the following combination of line cards:

- Cisco NCS 5700 Series base line cards alone.
- Cisco NCS 5700 Series base and Cisco NCS 5700 Series scale line cards.
- Cisco NCS 5700 Series scale line cards alone.

Prerequisites:

- Ensure that the system has either base or scale Cisco NCS 5700 Series line cards.
- Use **hw-module profile npu native-mode-enable** command to configure Cisco NCS 5700 Series in native mode.

Configuration Example

This example shows how to configure the hardware MDB profile as **l3max** in the Cisco NCS 5700 Series base line cards.

```
Router# configure
Router(config)# hw-module profile mdb l3max
Router(config)# commit
```

Verification

```
Router# show hw-module profile mdb-scale
MDB scale profile: l3max
```

Configuration Example

This example shows how to configure the hardware MDB profile as **l2max** in the Cisco NCS 5700 Series base line cards.

```
Router# configure
Router(config)# hw-module profile mdb l2max
Router(config)# commit
```

Verification

```
Router# show hw-module profile mdb-scale
MDB scale profile: l2max
```

Configuration Example

This example shows how to configure the hardware MDB profile as **l3max-se** profile in the Cisco NCS 5700 Series scale line cards.

```
Router# configure
Router(config)# hw-module profile mdb l3max-se
Router(config)# commit
```

Verification

```
Router# show hw-module profile mdb-scale
MDB scale profile: l3max-se
```

Configuration Example

This example shows how to configure the hardware MDB profile as **l2max-se** profile in the Cisco NCS 5700 Series scale line cards.

```
Router# configure
Router(config)# hw-module profile mdb l2max-se
Router(config)# commit
```

Verification

```
Router# show hw-module profile mdb-scale
MDB scale profile: l2max-se
```