System Setup and Software Installation Guide for Cisco NCS 560 Series Routers, IOS XR Release 26.1.x, 26.2.x

First Published: 2026-02-28

Last Modified: 2026-06-09

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)

© 2026 Cisco Systems, Inc. All rights reserved.

CONTENTS

CHAPTER 1 **Cisco NCS 560-4 Product Overview** 1

Command Modes 1

CHAPTER 2 **Bring-up the Router** 3

Boot the Router 3

Setup Root User Credentials 4

Access the System Admin Console 5

Configure the Management Port 6

Perform Clock Synchronization with NTP Server 8

CHAPTER 3 **Provision Network Devices using Zero Touch Provisioning** 9

Learn about Zero Touch Provisioning 9

Zero Touch Provisioning on a Fresh Boot of a Router 10

Fresh Boot Using Removable Storage Device 10

Fresh Boot Using DHCP 13

Build your configuration file 14

Create User Script 15

ZTP Shell Utilities 15

ZTP Helper Python Library 17

Limitations of temporary ZTP files 21

Set Up DHCP Server 22

Authentication on Data Ports 25

Invoke ZTP Manually 25

Configure ZTP BootScript 27

Customize ZTP Initialization File 28

ZTP file download behavior with libcurl 29

CHAPTER 4 **ZTP over Layer 2** 31
Prerequisites 33
Customizing the ztp.ini File 33
Prerequisites 34
Customizing the ztp.ini File 35

CHAPTER 5 **Deploy Router Using Bootz** 37
Supported Bootz Versions 38
Components used in the Bootz Process 38
Onboard Devices Using Bootz Workflow 39
Obtain Ownership Voucher 40
Build Bootstrapping Data 41
Provision Bootz Using DHCP Server 41
Bootz Workflow for Standby RP 46
Overview 46
Prerequisites 47
Restrictions 47
Use Cases 47
How the Router Obtains and Processes the OV Information 48

CHAPTER 6 **Perform Preliminary Checks** 51
Verify Status of Hardware Modules 51
Verify Node Status 52
Verify Environmental Parameters 53
Verify Software Version 54
Verify Firmware Version 54
Verify Interface Status 57

CHAPTER 7 **Create User Profiles and Assign Privileges** 59
Create a User Profile in System Admin VM 60
Create a User Group in System Admin VM 62
Create Command Rules 63
Create Data Rules 66

Change Disaster-recovery Username and Password 68

CHAPTER 8 **Perform System Upgrade and Install Feature Packages** 71

Upgrading the System 71

View supported software upgrade or downgrade versions 73

Optional packages 73

Limitations of optional packages 74

Commands and YANG data models supported using optional RPMs 75

Upgrading Features 76

Workflow for Install Process 77

Install Packages 78

Install Prepared Packages 83

Uninstall Packages 85

CHAPTER 9 **In Service Software Upgrade** 89

Overview 89

Restrictions and Usage Guidelines 90

Pre-installation Tasks 91

ISSU - Single-step Installation 95

ISSU - Multi-step Installation 98

Recovering from a Failed ISSU Operation 100

Installing Packages Using ISSU: Related Commands 100

CHAPTER 10 **Upgrading Field-Programmable Device** 103

Prerequisites for FPD Image Upgrades 103

Overview of FPD Image Upgrade Support 103

FPD upgrade service 104

Determining Upgrade Requirement 104

Manual FPD Upgrade 105

How to Upgrade FPD Images 105

Configuration Examples for FPD Image Upgrade 106

Auto FPD Upgrade 109

Limitations and Usage Guidelines 109

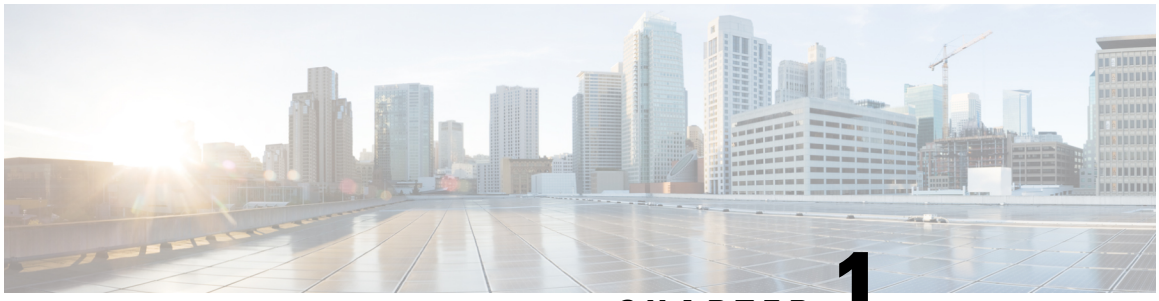
Configuring Auto FPD During System Upgrade 111

Configuring Auto FPD During ISSU 115
Automatic FPD Upgrade for PSU 124
Upgrade Failure 125

CHAPTER 11 **Manage Automatic Dependency** 127
Update RPMs and SMUs 127
Upgrade Base Software Version 128
Downgrade an RPM 129

CHAPTER 12 **Customize Installation using Golden ISO** 131
Limitations 131
Customize Installation using Golden ISO 132
Limitations 132
Golden ISO Workflow 132
Build Golden ISO Using Script 133
Install Golden ISO 135
Install Replace with Golden ISO 137

CHAPTER 13 **Disaster Recovery** 141
Boot using USB Drive 141
Create a Bootable USB Drive Using Compressed Boot File 141
Boot the Router Using iPXE 142
Zero Touch Provisioning 142
Setup DHCP Server 143
Invoke ZTP 145
Invoke ZTP Manually 145
Boot the Router Using iPXE 146
Disaster Recovery Using Manual iPXE Boot 147



CHAPTER 1

Cisco NCS 560-4 Product Overview

The Cisco NCS 560-4 Router is a four-rack unit (4-RU), fully-redundant, centralized forwarding system that has:,

- two router processor (RSP) slots
- six interface module (IM) slots
- aggregate backplane capacity of 1.8 Tbps, with 25 Gbps-capable SerDes for all IM slots

  support for (2+1) power supplies capable of delivering approximately 1.5 KW power to the chassis

  support for extended temperature based on route processor configuration

For more information on the Cisco NCS 560-4 router, see the *Cisco NCS 560-4 Router Hardware Installation Guide*.

The Cisco NCS 560-4 router supports the following route processors:

- N560-RSP4—a medium-scale route processor
- N560-RSP4-E—a high-performace router processor with an aggregate switching capacity of 800 Gbps.

Note The above route processors cannot be used together in the same router.

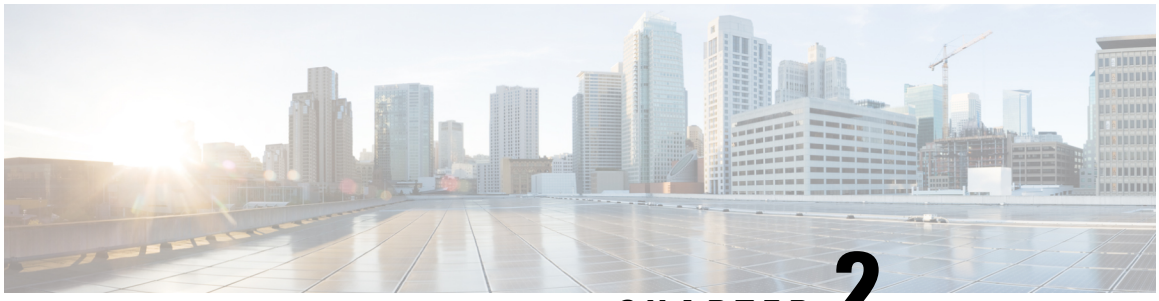
See the *Cisco N560-RSP4 and Cisco N560-RSP4-E Route Processor Hardware Installation Guide* for more information.

- Command Modes, on page 1

Command Modes

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

Command Mode	Description
XR EXEC mode (XR VM execution mode)	Run commands on the XR VM to display the operational state of the router. Example: `RP/0/RP0/CPU0:routerRP0/CPU0:ios#`
XR Config mode (XR VM configuration mode)	Perform security, routing, and other XR feature configurations on the XR VM. Example: `RP/0/RP0/CPU0:routerRP0/CPU0:ios#`**`configure`** `RP/0/RP0/CPU0:router(config)#`
System Admin EXEC mode (System Admin execution mode)	Run commands on the System Admin 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#`
System Admin Config mode (System Admin configuration mode)	Run configuration commands on the System Admin VM to manage and operate the hardware modules of the entire chassis. Example: `RP/0/RP0/CPU0:routerRP0/CPU0:ios#`**`admin`** `sysadmin-vm:0_RP0#`**`config`** `sysadmin-vm:0_RP0(config)#`



CHAPTER 2

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.

- Boot the Router, on page 3
- Setup Root User Credentials, on page 4
- Access the System Admin Console, on page 5
- Configure the Management Port, on page 6
- Perform Clock Synchronization with NTP Server, on page 8

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 4.

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/
!
```

Example

```
Enter root-system username: admin
Enter secret:
Enter secret again:
Use the 'configure' command to modify this configuration.
User Access Verification

Username: admin
Password:

RP/0/RP0/CPU0:ios# show run username
Sun May 30 14:20:42.311 UTC
username admin
 group root-lr
 group cisco-support
 secret 10
$6$RS5kn1r/ww.DDn1.$eDFxhqTEYa6hqTs3MODQt11mBp4cMgdQqt.syC/J831QIl1yJT9vd2W8zEHfBKz4.z4FyImRdzwvKTqAMuyBA0

!
```

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 **admin**

Example:

The following example shows the command output :

```
RP/0/RP0/CPU0:routerRP0/CPU0:ios#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
```

Step 3 (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 on RP is the management port. Ensure that the port is connected to management network.

Procedure

Step 1 **configure**

Example:

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

Enters mode.

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

Example:

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

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

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

Example:

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

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

Step 4 **no shutdown**

Example:

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

Places the interface in an "up" state.

Step 5 **exit**

Example:

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

Exits the Management interface configuration mode.

Repeat the above steps for the redundant route processor.

Step 6 **ipv4 virtual address** *ipv4 virtual address subnet-mask*

Example:

```
RP/0/RP0/CPU0:ios(config)#ipv4 virtual address 1.70.31.160 255.255.0.0
```

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

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

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

end —Prompts user to take one of these actions:

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

What to do next

Connect 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
```

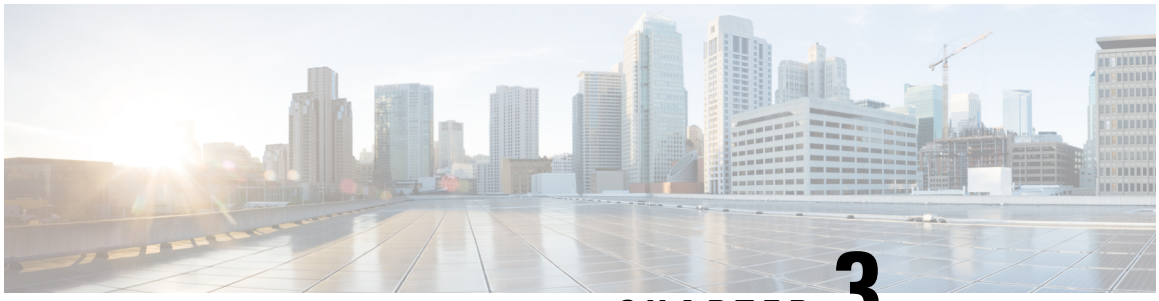
Enters mode.

Step 2 **ntp server** *server_address*

Example:

```
RP/0/RP0/CPU0:routerRP0/CPU0:ios(config)#ntp server 64.90.182.55
```

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



CHAPTER 3

Provision Network Devices using Zero Touch Provisioning

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 phyton.

- Learn about Zero Touch Provisioning, on page 9
- Zero Touch Provisioning on a Fresh Boot of a Router, on page 10
- Build your configuration file, on page 14
- Limitations of temporary ZTP files, on page 21
- Set Up DHCP Server, on page 22
- Invoke ZTP Manually, on page 25
- Configure ZTP BootScript, on page 27
- Customize ZTP Initialization File, on page 28
- ZTP file download behavior with libcurl, on page 29

Learn about Zero Touch Provisioning

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

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.

ZTP provides multiple options, such as:

- Automatically apply specific configuration in a large-scale environment.
- Download and install specific IOS XR image.
- Install specific application package or third party applications automatically.
- Deploy containers without manual intervention.

- Upgrade or downgrade software versions effortlessly on thousands of network devices at a time

Benefits of Using ZTP

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

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

  By automating repeated tasks, ZTP allows network administrators to concentrate on more important stuff.
- ZTP process helps you to quickly restore service. Rather than troubleshooting an issue by hand, you can reset a system to well-known working status.

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 10
- **Manual Invocation:** Use this method when you want to forcefully initiate ZTP on a fully configured device. See Invoke ZTP Manually, on page 25.

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.

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, ZTP looks for the provisioning files in the following order:

   a. `<usb-root>/xr-config/<chassis-serial>/ztp_actions.json`

   b. `<usb-root>/xr-config/ztp_actions.json`

   c. `<usb-root>/xr-config/<chassis-serial>/router-cfg`

   d. `<usb-root>/xr-config/router-cfg`

4. 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.

5. If a `ztp_actions.json` file is found, ZTP uses only the configuration path specified inside that JSON file.

   Sample generic configuration:

   ```
   "global": {
       "actions": {
         "config": "xr-config/common/router-cfg"
       }
          }
   ```

6. For device-specific configuration, ZTP looks for the `router-cfg` file inside the folder matching the device's chassis serial number within the `xr-config` folder and applies this device-specific configuration. For example, `/USB-path/xr-config/FOC2102R1D0/router-cfg`.

   Device-specific configuration:

   ```
   "FOC2102R1D0": {
       "actions": {
         "image-install": {
           "boot-image": {
   ```

```
            "os-version": "7.9.1",
            "path": "xr-config/FOC2102R1D0/8000-x64-7.9.1.iso"
          }
        }
      }
    }
```

For information about the JSON keys used above, see JSON keys and options in ztp_actions file.

7. If the device-specific configuration is not found, ZTP looks for the generic `router-cfg` file directly inside the `xr-config` folder and applies this generic configuration.
8. If the config path inside `ztp_actions.json` is missing or the file is absent, ZTP retries and does not fall back to `router-cfg`.
9. The device applies the configuration.
10. The network device is now up and running.

JSON keys and options in ztp_actions.json file

The supported JSON keys and options in `ztp_actions.json` include:

- Top-level keys:
  - `global`: Generic or default action block used when no device-specific block matches.
  - `<chassis-serial>`: Device-specific action block keyed by chassis serial number (e.g., `FOC2102R1D0`), which takes priority over `global`.
  - `actions`: Container for actual ZTP actions.
- Action keys under `actions`:
  - `config`: String path to the XR config file to apply. Required for normal config provisioning. If absent, ZTP retries.
  - `image-install`: Requests image upgrade/install before config apply.
    - `boot-image` sub-container includes:
      - `os-version`: Requested image version string.
      - `path`: Path to ISO/GISO image on USB.
  - `enable-secure-mode`: Boolean (`true` or `false`). When `true`, ZTP enables secure mode and restarts ZTP to continue in Secure ZTP flow. Image or config actions in the same file are ignored. If unsupported, ZTP moves to the next fetcher. If enabling secure mode fails, ZTP exits to avoid undefined state. Default is `false` if absent.

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 14.

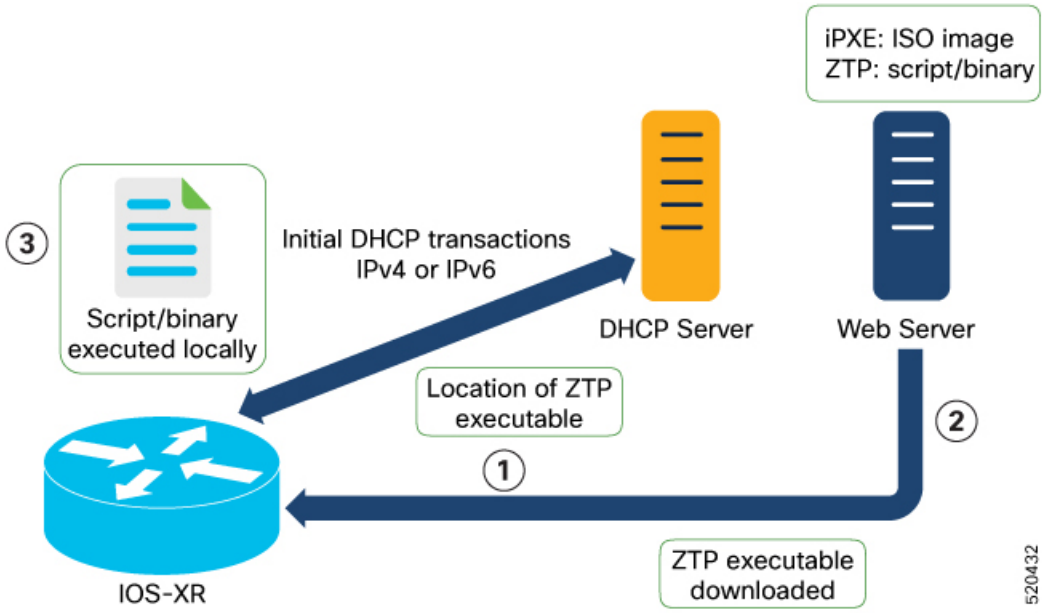
Note When you use a USB flash drive as a source for ZTP, you can't use the script file for provisioning. The script file isn't supported for USB fetcher.

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.

Build your configuration file

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

The configuration file content starts with *!! IOS XR* 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.

Limitations of temporary ZTP files

During the ZTP (Zero Touch Provisioning) process, temporary files such as */disk0:/ztp/xr_config/interface_list* may remain on disk if the ZTP process does not exit gracefully.

Note For Cisco IOS XR software versions prior to 7.5.1, ZTP used the */disk0:/ztp/xr_config/xr_config_to_revert* file for similar tracking.

```
RP/0/RP0/CPU0:ios#run
[node0_RP0_CPU0:~]$ls /disk0\:/ztp/xr_config/
current  interface_list  xr_config_to_revert
```

```
[node0_RP0_CPU0:~]$rm /disk0\:/ztp/xr_config/interface_list
[node0_RP0_CPU0:~]$rm /disk0\:/ztp/xr_config/xr_config_to_revert
```

The ZTP process may exit abruptly under the following conditions:

- The ZTP process crashes.
- The ZTP process is manually killed.
- A reload is triggered from a ZTP script.
- The ZTP process is terminated from a script.
- A device reload or RP switchover occurs while ZTP is running.
- An RP is shut down during the ZTP process and reinserted later.

Before performing a manual upgrade, ensure that these temporary ZTP files (*/disk0:/ztp/xr_config/interface_list* and */disk0:/ztp/xr_config/xr_config_to_revert*) are deleted if they exist on disk. This prevents ZTP from bringing down configured interfaces on the next boot, even when a local username is already configured, before ZTP exits.

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";
   }
}
```

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";
}
```

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 `dhcpcd`. Hence, ZTP can run on all interfaces irrespective of their availability.

Note `dhclient` has been migrated to `dhcpcd` in Cisco IOS XR Release 25.3.1 as `dhclient` has reached its end of service.

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
```

Configure ZTP BootScript

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

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

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

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

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

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

```
#!/bin/bash
exec &> /dev/console # send logs to console
source /pkg/bin/ztp_helper.sh

# If we want to only run one time:
xrcmd "show running" | grep -q myhostname
if [[ $? -eq 0 ]]; then
    echo Already configured
fi

# Set the hostname
cat >/tmp/config <<%%
!! XR config example
hostname myhostname
%%
xrapply /tmp/config

#
# Force an invoke of ZTP again. If there was a username normally it would not run. This
forces it.
# Kill off ztp if it is running already and suppress errors to the console when ztp runs
below and
```

```
# cleans up xrcmd that invokes it. ztp will continue to run however.
#
xrcmd "ztp terminate noprompt" 2>/dev/null
xrcmd "ztp initiate noprompt" 2>/dev/null
```

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.

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]
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.
```

Prior to Cisco IOS XR Release 7.5.1, the ztp disable command creates a `ztp.ini` file in the `/disk0:/ztp` folder. Starting with Cisco IOS XR Release 7.5.1 and later, the ztp.ini file is created in the `/disk0:/ztp/internal` folder. This file tracks whether ZTP is disabled, ensuring persistence of the disabled state during upgrades and downgrades.

However, the disabled state of ZTP is not persistent when reimaging is performed between pre-7.5.1 and post-7.5.1 images. This means reimaging across these release boundaries requires reconfiguration to disable ZTP again.

If ZTP was disabled:

- Before upgrading from a pre-7.5.1 image, copy `/disk0:/ztp/ztp.ini` to `/disk0:/ztp/internal`.
- Before downgrading to a pre-7.5.1 image, copy `/disk0:/ztp/internal/ztp.ini` to `/disk0:/ztp`.

ZTP file download behavior with libcurl

Cisco IOS XR ZTP uses the `libcurl` library to download provisioning files such as scripts and software images in releases 24.1.1, 24.2.1, 24.2.2, and 24.2.21. The ZTP process retrieves the download URL from DHCP options and passes it directly as a string to libcurl, which initiates the file transfer.

Key behaviors and considerations for ZTP file downloads with `libcurl` include:

- If the download URL uses HTTPS, `libcurl` automatically attempts the HTTPS connection. ZTP allows downloads from HTTPS servers without needing explicit certificate verification configuration, so HTTPS downloads succeed without extra flags or options.
- Administrators sometimes add the `-k` option to a curl command to allow insecure HTTPS connections. However, ZTP does not interpret `-k` as an option; it treats it as part of the DHCP-provided URL string.

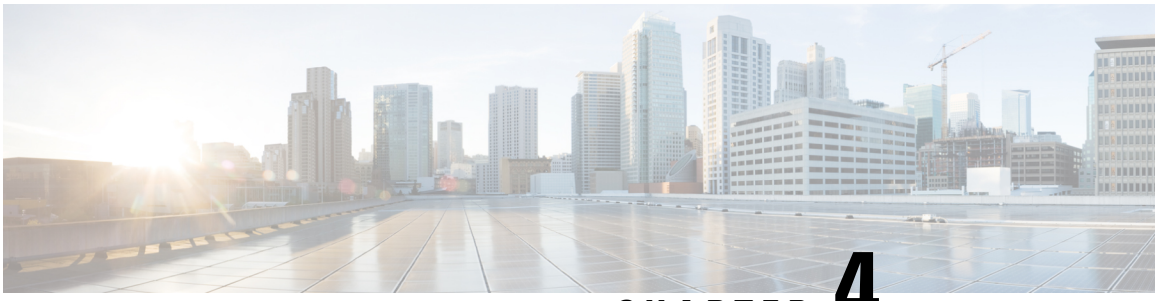
  For example, using a DHCP-provided value like *https://server.example.com/ztp.script -k* causes ZTP to interpret `-k` as part of the path, likely resulting in a 404 error or download failure.
- Do not include command-line options such as `-k` in DHCP-provided ZTP URLs in releases 24.1.1, 24.2.1, 24.2.2, and 24.2.21. Use only a valid file URL, such as `https://server.example.com/ztp.script`.

- If needed, you can disable libcurl file downloads in the ZTP configuration file (`ztp.ini`) in releases 24.1.1, 24.2.1, 24.2.2, and 24.2.21 by adding:

  ```
  [Http]
  libcurl_http_get: False
  ```

  When `libcurl` is disabled, ZTP uses the alternative download method as defined by the framework or falls back to the Linux curl utility.

Following these guidelines ensures that ZTP reliably downloads files using libcurl without errors caused by invalid options in the URL.



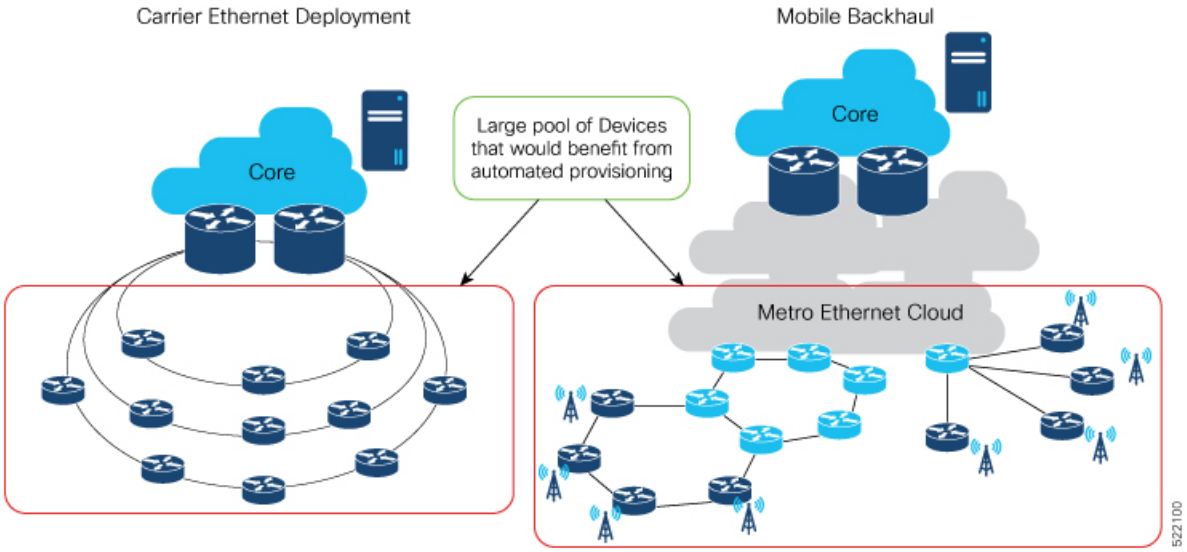
CHAPTER 4

ZTP over Layer 2

When a device boots up without any configuration, it is not possible to configure the VLAN to tag the DHCP request packets. Because ZTP is the first active process that runs before any router configuration is present or before any manual intervention, ZTP can either:

- Monitor a set of control packets to receive the VLAN IDs, or
- Be preconfigured with a set of VLAN IDs

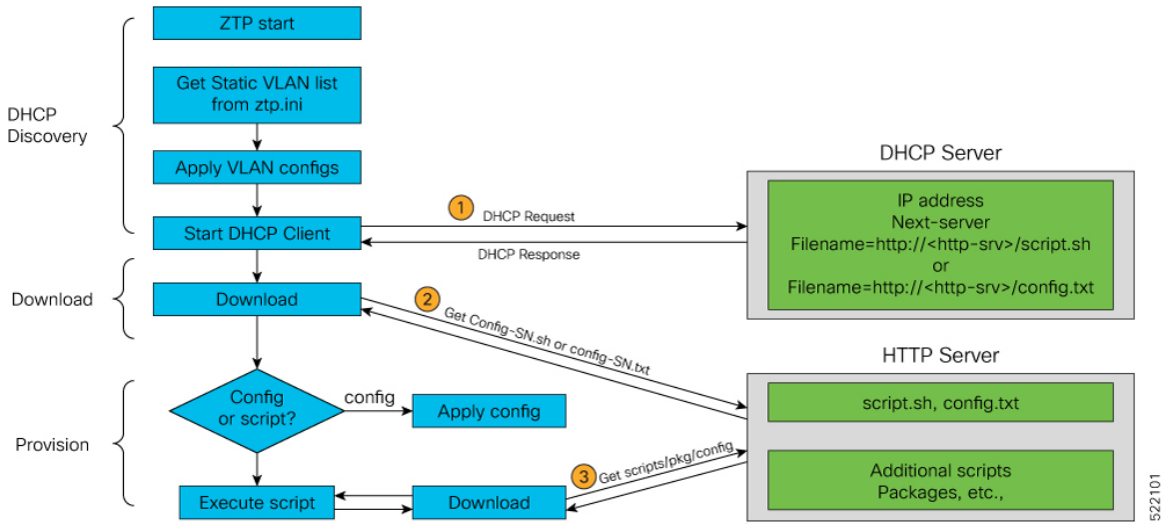
Consider the following sample network:



ZTP obtains VLAN details in one of the following ways:

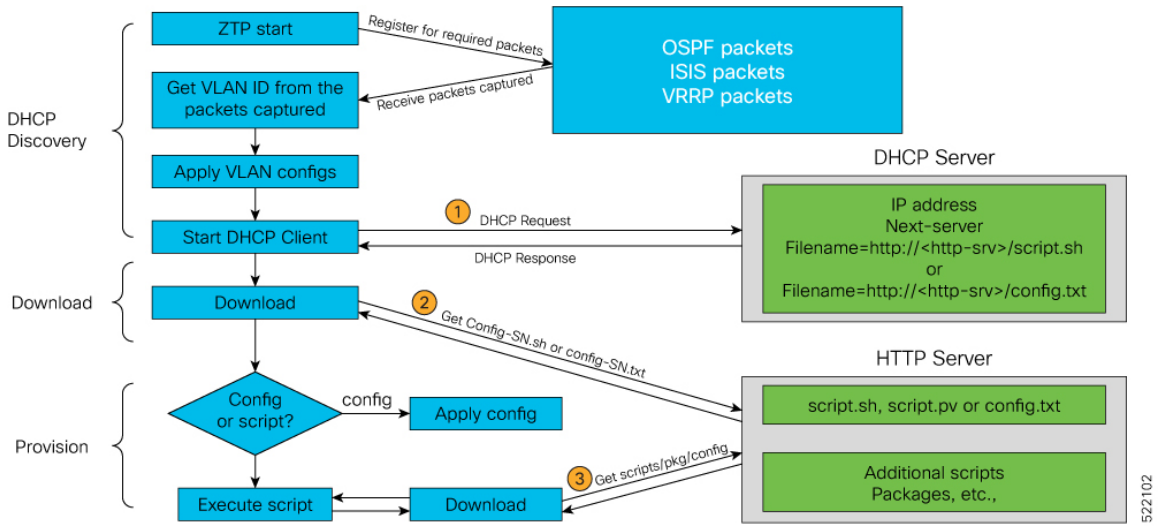
1. Statically, by reading the options mentioned in ztp.ini file

Figure 1: Getting VLAN Details From ztp.ini File



2. Dynamically, by snooping the packets in the netwok

Figure 2: Getting VLAN Details by Packet Snooping



Note Dynamic VLAN workflow with snooping is supported only on a fresh boot or reload of the router.

- Prerequisites, on page 33
- Customizing the ztp.ini File, on page 33
- Prerequisites, on page 34
- Customizing the ztp.ini File, on page 35

Prerequisites

1. Only the following Layer 2 encapsulation protocols are supported:
   a. DOT1Q
      1. Either inner VLAN ID
      2. Or only an outer VLAN ID
   b. DOT1AD
      1. An outer VLAN ID and an inner VLAN ID.

         For more information, see the section *Configuring Layer 2 Encapsulation*.
2. ZTP processes packets only from the following protocols to detect the VLAN IDs:
   a. OSPF
   b. ISIS
   c. VRRP

Customizing the ztp.ini File

The ZTP configuration file is called the `ztp.ini`. This file defines a default sequential flow for ZTP.

To modify the ZTP configurable options, create a copy of the `ztp.ini` file in any location, for example, `/disk0:/ztp/` directory.

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

To include VLAN in the *ztp.ini* file, customize the `Fetcher Priority` attribute.

`Fetcher Priority` defines which port ZTP should use to get the provisioning details. By default, each port has a fetcher priority defined in the *ztp.ini* file. You can modify the default priority of the fetcher. Allowed range is from 0 to 9.

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

By default, the USB port has the higher priority.

In this example, ZTP first sends IPv4 DHCP request on all the management ports. If there is a failure, then ZTP sends IPv6 DHCP request on all the management ports. Similarly, the same order is followed on all the data ports as well as for VLAN ports.

```
# Copyright (c) 2018-2020 by cisco Systems, Inc.
# All rights reserved.
```

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

[Fetcher Priority]
usb:    0
Mgmt4:  1
Mgmt6:  2
DPort4: 3
DPort6: 4
Vlan4: 6
Vlan6: 7

[Vlan Options]          <--Vlan options can be set only if the vlan fetcher priority is
set.
snoop_packets: False    <-- a 'False' value sets up static VLAN; a 'True' value sets up
dynamic OSPF, ISIS, or VRRP packet snooping in the network.
vlan_type: dot1q
vlan_ids: 650
```

The other configurable options are:

- `ZTP`: 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.
- `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
  ```

Prerequisites

1. Only the following Layer 2 encapsulation protocols are supported:
   a. DOT1Q
      1. Either inner VLAN ID
      2. Or only an outer VLAN ID
   b. DOT1AD
      1. An outer VLAN ID and an inner VLAN ID.

         For more information, see the section *Configuring Layer 2 Encapsulation*.
2. ZTP processes packets only from the following protocols to detect the VLAN IDs:
   a. OSPF
   b. ISIS
   c. VRRP

Customizing the ztp.ini File

The ZTP configuration file is called the `ztp.ini`. This file defines a default sequential flow for ZTP.

To modify the ZTP configurable options, create a copy of the `ztp.ini` file in any location, for example, `/disk0:/ztp/` directory.

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

To include VLAN in the *ztp.ini* file, customize the `Fetcher Priority` attribute.

`Fetcher Priority` defines which port ZTP should use to get the provisioning details. By default, each port has a fetcher priority defined in the *ztp.ini* file. You can modify the default priority of the fetcher. Allowed range is from 0 to 9.

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

By default, the USB port has the higher priority.

In this example, ZTP first sends IPv4 DHCP request on all the management ports. If there is a failure, then ZTP sends IPv6 DHCP request on all the management ports. Similarly, the same order is followed on all the data ports as well as for VLAN ports.

```
# Copyright (c) 2018-2020 by cisco Systems, Inc.
# All rights reserved.

[Startup]
start:         True
retry_forever: False

[Fetcher Priority]
usb:    0
Mgmt4:  1
Mgmt6:  2
DPort4: 3
DPort6: 4
Vlan4: 6
Vlan6: 7

[Vlan Options]          <--Vlan options can be set only if the vlan fetcher priority is
set.
snoop_packets: False    <-- a 'False' value sets up static VLAN; a 'True' value sets up
dynamic OSPF, ISIS, or VRRP packet snooping in the network.
vlan_type: dot1q
vlan_ids: 650
```

The other configurable options are:

- `ZTP`: 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.
- `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
```

CHAPTER 5

Deploy Router Using Bootz

With the Bootz process, you can securely and seamlessly provision network devices accurately within minutes and without any manual intervention.

Table 1: Feature History Table

Feature	Release Information	Feature Description
Provisioning Using Bootz Process	Release 7.11.1	This feature allows devices in the network to establish a secure connection with the remote Bootz server and authenticate information using a three-step validation process. This process involves validating the network device, the Bootz server, and the onboarding information thereby mitigating security risks and preventing malicious actions during remote provisioning.

Unlike the Secure ZTP process, which relies on vendor-specific definitions for bootstrapping a device, the Bootz process offers a specification that outlines data elements in a vendor-agnostic manner. It also details the necessary operations at turn-up time, integrating them into the boot process.

Also, the bootstrap request in the Bootz process includes the unique identifier or serial number for each node as opposed to the Secure ZTP process where the bootstrap request does not include serial numbers. The Bootz server returns the signed onboarding information with ownership voucher and owner certificate for the requested serial number of the device.

- Supported Bootz Versions, on page 38
- Components used in the Bootz Process, on page 38
- Onboard Devices Using Bootz Workflow, on page 39
- Obtain Ownership Voucher, on page 40
- Build Bootstrapping Data, on page 41
- Provision Bootz Using DHCP Server, on page 41

Supported Bootz Versions

This table provides the Bootz versions supported in each release. The Bootz Bootstrap server must be compatible with the respective Bootz version.

Table 2: Bootz Versions

Release	Version with File Path
Release 24.4.1	`openconfig/bootz v0.3.1`
Release 24.3.1	`openconfig/bootz v0.3.1`
Release 24.2.1	`openconfig/bootz v0.1.1-0.20231106050618-8d6e2559f803`
Release 24.1.1	`openconfig/bootz v0.0.0-20230809153947-e6bc0be82dd8`
Release 7.11.1	`openconfig/bootz v0.1.0`

Components used in the Bootz Process

These components are part of the Bootz process.

- **Onboarding Device (Router):** A router is a Cisco device that you want to provision and connect to your network. Bootz is supported only on platforms that have *Hardware TAM*[1] support.
- **DHCP Server:** The DHCP server provides the URL where the Bootz process can access the bootstrapping information.
- **MASA Server:** You can generate and store the ownership voucher in the MASA server. The MASA server sends the ownership voucher to the Bootz server so that the Bootz process validates the device and establishes device ownership.
- **Bootz Bootstrap Server:** A Bootz Bootstrap server is any gRPC server used as a Bootz bootstrapping data source. For example, Google Proto. The Bootz Bootstrap server is complaint with Openconfig Bootz standards.

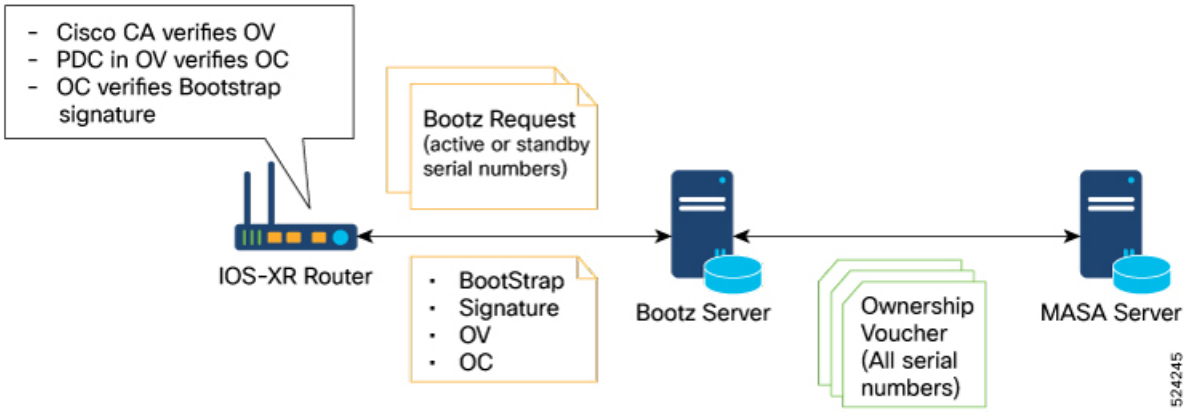
Note Bootz only supports a single name-server. As a result, when the DHCP server has more than one server address configured, Bootz fails to apply the server configuration.

The Bootz server contains these artifacts:

- **Cisco IOS XR software images:** You can download Cisco images, SMU, and patches from the Cisco Support & Downloads page.

[1] A secure storage device that stores the customer certificates and Cisco's internal secure data like trust anchors, SUDI certificates, secure flags, and other security information.

- **Bootstrapping Data:** It is a collection of data that you have created and uploaded to the Bootz server. The router obtains this data from the Bootz server during the provisioning process.



Onboard Devices Using Bootz Workflow

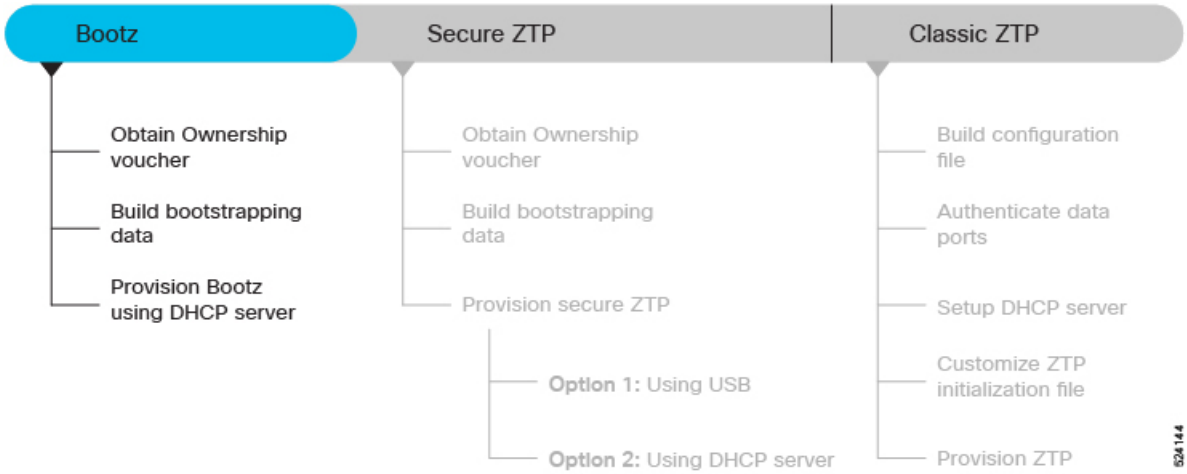
The Cisco IOS XR software supports Bootz provisioning capabilities. The Bootz process uses the Google Remote Procedure Call (gRPC) protocol for fetching information from a remote server.

The Bootz workflow performs these validations to onboard the remote devices securely.

1. **Router Validation**: The Bootz server authenticates the router before providing the bootstrapping data.
2. **Server Validation**: The router in turn validates the Bootz server and ensures that the onboarding is performed for the correct network. Once it is validated, the Bootz server sends the bootstrapping data (for example, a YANG data model) or artifact to the router.
3. **Artifact Validation**: The router validates the bootstrapping data or artifacts received from the Bootz server.

This figure provides the Bootz workflow and the processes involved in the workflow. The sections that follow describe these processes in detail.

Figure 3: Bootz Workflow



Obtain Ownership Voucher

The ownership voucher is used to identify the owner of the device by verifying the owner certificate stored in the device.

How to obtain Ownership Voucher

These steps help you obtain the ownership voucher from Cisco:

1. Contact Cisco Support.
2. Provide these information in your request to Cisco.
   - **Pinned Domain certificate (PDC):** PDC is an X.509 v3 certificate structure that uses Distinguished Encoding Rules (DER). The router uses this certificate to trust a public key infrastructure for verifying 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.
   - Purchase order details with the serial numbers of the routers.

   Sample Request:

```
{
   "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"
}
```

3. Cisco generates the ownership voucher in .vcj format (Example: DCA213140YX.vcj) and sends the voucher in response to your request.

Build Bootstrapping Data

Steps to build the bootstrapping data:

1. Create and upload the bootstrapping data to the gRPC server or Bootz bootstrap server.
2. The router sends a bootstrap request with these artifacts to the Bootz server.
   - Serial number of the control card or line card
   - Software image to download and install
   - Bootloader Password for the device
   - Certificate used to validate the bootstrap server
   - Bootstrap server configuration information such as server credentials, path information, authentication information, and certificates

   For the request message format, see the Bootstrap Request Message.
3. The Bootz server returns the listed bootstrapping data in its response to the router. The router receives these data during the provisioning process.
   - **Signed Bootstrap Response:** Each bootstrap response contains the onboarding information for:

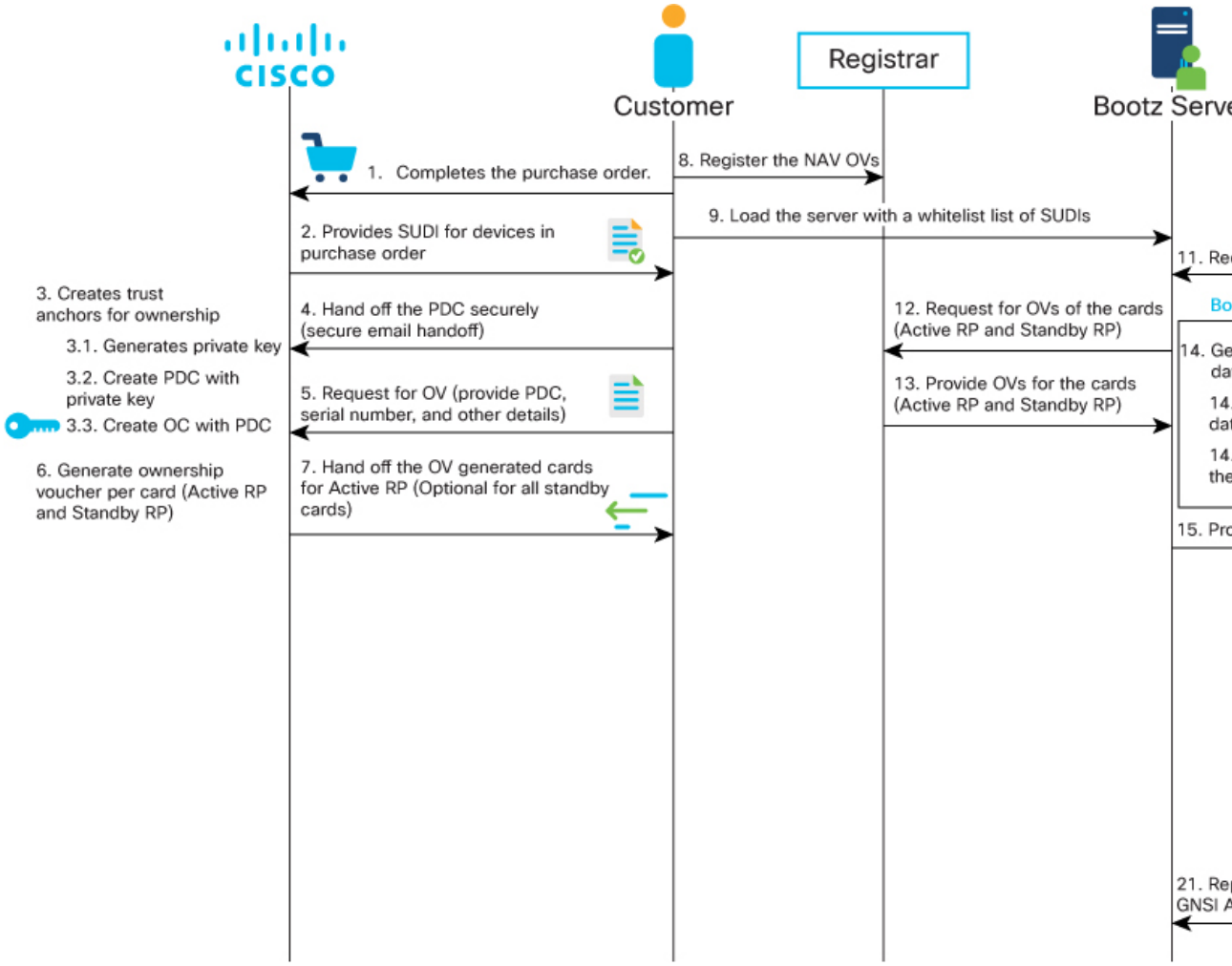
     For the response message format, see the Bootstrap Response Message for a single card.
   - **Owner Certificate:** The owner certificate is installed on the router with your organization's public key. The router uses this public key in the owner certificate to verify the signature in the signed bootstrap response artifact.
   - **Ownership Voucher:** The ownership voucher is used to identify the device owner by verifying the owner certificate stored in the device. Cisco generates and supplies the ownership voucher in response to your request containing the PDC and device serial numbers. For more information, see How to obtain Ownership Voucher.
4. When the router obtains the onboarding information from the Bootz server, the router reports the bootstrapping progress to the Bootz server using the API calls.

Provision Bootz Using DHCP Server

When you boot the device, the Bootz process initiates automatically on a device without prior configuration. During the process, the router receives the details of the configuration file from the DHCP server.

This figure illustrates the end-to-end sequence of the Bootz process:

Figure 4: End-to-end sequence of the Bootz process



Before you begin

As part of the initial setup for secure ZTP, the network administrator:

- Ensures to enable secure ZTP on the router using the **ztp secure-mode enable** command and reload the router.
- Contacts Cisco Support and follows the steps in How to obtain Ownership Voucher to obtain a voucher from Cisco.

Procedure

Step 1 Upload the listed bootstrapping data to the Bootz server. Refer to your vendor documentation as the upload procedure may vary from server to server.

- Cisco IOS XR software images

Note
Download Cisco images, SMU, and patches from the Cisco Support & Downloads page.

- Serial numbers of the routers to be onboarded
- Owner certificates
- Pinned Domain Certificate (PDC)
- Ownership vouchers

Step 2 Set up the DHCP server to provide the redirect URL to the router:

Before triggering the secure ZTP process, configure the DHCP server so that it provides the location of the IOS-XR image to the router. For information about how to configure the DHCP server, see your DHCP server documentation.

Configure these parameters in the DHCP server:

- `option-code`: Use one of these DHCP SZTP redirect option parameters in the `option-code` setting.
  - OPTION_V4_SZTP_REDIRECT (143): DHCP v4 code for IPv4.
  - OPTION_V6_SZTP_REDIRECT (136): DHCP v6 code for IPv6.
- `option-length`: Provide the option length in octets.
- `bootstrap-servers`: A list of servers. The onboarding device contact these servers for the bootstrapping data.

  `"bootz://<ip-address-or-hostname>[:<port>]<endpoint>"`

Example: `option dhcp6.bootstrap-servers code 136 = text;`

Step 3 Power on the router.

This procedure provides the high-level workflow of the Bootz process:

a. 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.

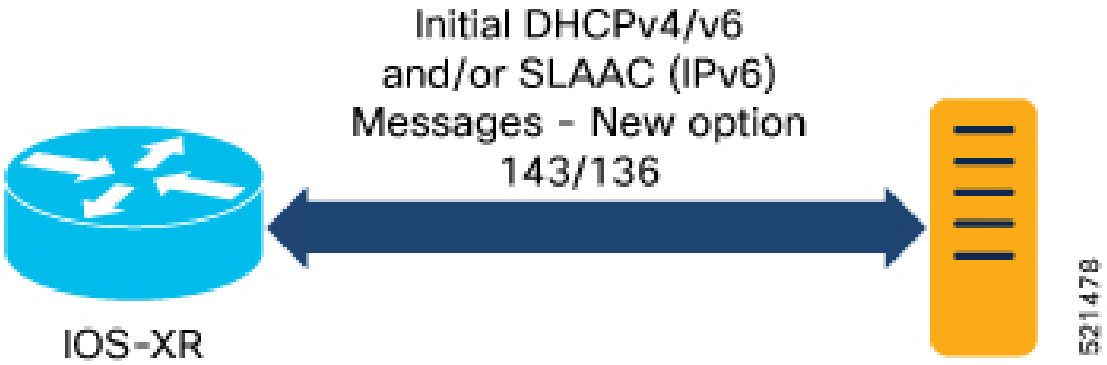
Note
When `secure-ztp mode` is enabled, the ZTP process accepts only the `secure-redirect-URL` and ignores the presence of the boot file name option from the DHCP response.

b. **DHCP discovery**:

1. The router initiates a DHCP request to the DHCP server.
2. The DHCP server responds with a DHCPv4 143 address option (for IPv4 addressing) or a DHCPv6 136 option (for IPv6 addressing).

Note
URLs to access bootstrap servers for further configuration are listed in options 136 and 143.

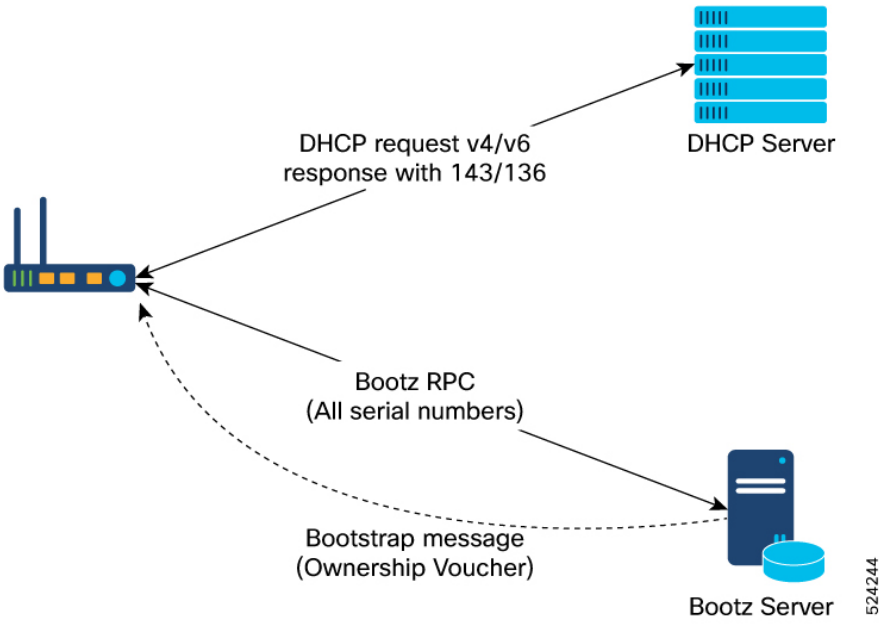
Figure 5: DHCP discovery



c. **Router and Bootz server validation**:

1. After receiving the URL from the DHCP server, the router initiates a gRPC connection to the Bootz server. The Bootz server IP address is obtained from the DHCP response.
2. The Bootz server authenticates the router before it provides the bootstrapping data.
3. After the Bootz server authenticates the router or the onboarding device, the router validates the Bootz server to ensure that the onboarding is performed for the correct network.

   After validating the Bootz server, the router sends the serial number for each control card or line card and other artifacts in its bootstrap request.
4. After its validation, the Bootz server sends the required artifacts along with the bootstrap response data to the router or the onboarding device.



d. **Ownership Voucher verification**:

The router receives the bootstrap response data that contains owner certificate, ownership voucher for each serial number, and the details of the image upgrade, if any.

Bootstrap response data includes the following:

- Image path
- Image version
- Trust anchor
- Boot configuration
- GNSI artifacts

These artifacts come from the Bootz server as a bootstrap response gRPC message. The router verifies the ownership voucher by validating its signature to one of its preconfigured trust anchors and downloads the image. When the router obtains the onboarding information, it reports the bootstrapping progress to the Bootz server.

e. **Artifact Validation**:

The router validates the artifacts received from the Bootz server as follows:

1. The device extracts the `pinned-domain-cert` node, an X.509 certificate from the ownership voucher to verify the owner certificate.
2. The device authenticates the owner certificate by performing the X.509 certificate path verification process on the trusted certificate.
3. Finally, the device verifies whether the artifact is signed by the validated owner certificate.

f. **Provision the device**:

1. The device first processes the boot image information.
2. Executes the script and then onboards the artifacts received from the Bootz server.

g. After the onboarding process is completed, the network device is operational.

Bootz Workflow for Standby RP

Table 3: Feature History Table

Feature	Release Information	Feature Description
Bootz Workflow for Standby RP	Release 24.3.1	This feature enables the Bootz workflow to achieve full-system onboarding for devices with both active and standby Route Processors (RPs). In earlier releases, the Bootz workflow only supported onboarding for devices with an active RP. With this enhancement, the Bootz workflow can now detect faulty or tampered standby cards that are inserted dynamically during or after the active RP Bootz process. It does this by verifying the ownership voucher (OV) of the other cards during the initial Bootz process for the active RP. Faulty cards can be shut down to prevent security threats during remote provisioning, ensuring smooth network operation. This feature allows Bootz workflow to validate the standby RP as part of the active RP Bootz process. This feature introduces the **ztp bootz-server** command. This feature modifies the `Cisco-IOS-XR-ztp-cfg.yang`. (see GitHub, YANG Data Models Navigator)

Overview

The Bootz workflow now processes the ownership voucher (OV) for multiple control cards or line cards detected on the standby RP, either before the Bootz process for an active RP starts, during its progress, or after its completion.

With this new feature, the Bootz workflow:

- Allows dynamic insertion or replacement of standby control cards and line cards.
- Processes the ownership voucher (OV) bundle that the Bootz server sends to the router for the standby control cards or line cards.
- Performs the ownership verification of all standby control cards and line cards.

This feature allows you to configure the ZTP Bootz server with the **ztp bootz-server** command to store the server and vendor information received during the initial Bootz process for the active RP.

The router uses this configuration to communicate with the Bootz server and obtain the OV bundle (.tar file) for dynamically inserted or replaced standby control cards or line cards.

Prerequisites

- Configure the Bootz server to return the bootstrap data response message for all the serial numbers of the cards on the device with either the OV bundle or individual ownership voucher for each card.
- Include the **ztp bootz-server** configuration in the server's onboarding information or the vendor configuration information. This configuration is received from the Bootz server during the initial GetBootstrapDataRequest exchange for the active RP.
- Ensure that the routers running the Bootz client can process the OV bundle.

Restrictions

The dynamic Bootz workflow for the standby RP is triggered only if these conditions are met:

- Secure ZTP is enabled on the device.
- The Bootz process for the active RP is completed or not in progress.
- The Bootz server configuration from the initial Bootz process for active RP is available for dynamically inserted standby cards.

Use Cases

These use cases describe different scenarios where the standby RP cards are detected and the OV information is processed accordingly.

Use Case 1 - Standby Card Detected Before Bootz Process for Active RP Starts

If both the active RP and standby RP are detected during the initial boot process before the router communicates with the Bootz server:

- The router sends a bootstrap request to the Bootz server, including the serial numbers for both the active RP and standby RP.
- The Bootz server responds with the OV information for both the active RP and standby RP in its bootstrap response.
- If the response message for the initial bootstrap data request (for active RP) does not include an OV bundle for the standby RP, an additional bootstrap data request is triggered to fetch the OV information for the standby RP.

If there is no OV bundle in the bootstrap response for the initial bootstrap data request (for active RP),

Use Case 2 - Standby Card Detected During Bootz Process for Active RP

If the standby card is detected while the Bootz process for the active RP is in progress, the Bootz process for the standby RP is automatically triggered after the active RP's Bootz process completes.

In this scenario, the router uses the server information received during the initial Bootz process for the active RP, which you have configured using the **ztp bootz-server** command, to:

- Communicate with the Bootz server.
- Send the serial numbers for the dynamically inserted cards in its bootstrap request for standby RP to the Bootz server.
- Obtain the OV bundle (.tar file) from the Bootz server and process the OV for each card with a matching serial number. For more information about the OV bundle, see How the router obtains and processes the OV information.

The Bootz server, in turn, sends the OV information for the standby RP in its bootstrap response.

Use Case 3 - Standby Card Detected After Bootz Process for Active RP Completes

If the standby RP is detected after the initial Bootz process for active RP is completed, the Bootz process is triggered again automatically. The Bootz process is re-triggered if one of the following events occur:

- When a new card is inserted.
- When an existing card is replaced with another card.

Store the server information obtained during the initial Bootz process for active RP. As the secure ZTP workflow is not re-triggered for dynamically inserted standby control cards or line cards, the router uses this stored server information to communicate with the Bootz server and obtain the ownership vouchers for the newly inserted standby control cards or line cards.

Configuration to store server information obtained from the active RP Bootz process:

```
RP/0/RP0/CPU0:ios# config
RP/0/RP0/CPU0:ios(config)#ztp bootz-server ip 1.1.1.1 port 5000 trust-anchor
/misc/disk1/ta.cert
RP/0/RP0/CPU0:ios(config)# commit
```

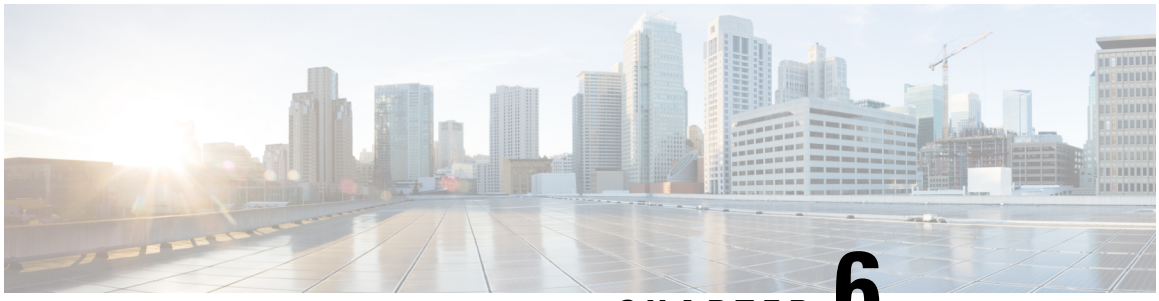
Once the dynamic Bootz workflow for standby RP is triggered, the router communicates with the Bootz server using the server and vendor configuration information specified in the **ztp bootz-server** command. The Bootz server then sends the OV information for the dynamically inserted standby cards in its bootstrap response.

How the Router Obtains and Processes the OV Information

The router uses one of these methods to process the OV information that it has obtained from the Bootz server.

- **If the Bootz server is configured to send an OV bundle:**
  1. The Bootz server sends the OV bundle as a single tar file in the bootstrap response to the router for the RPs.
  2. The router running the Bootz client processes the tar file to verify individual ownership voucher for each serial number.
  3. The router loads the owner certificate and the ownership voucher on each card with a serial number matching the serial number included in the ownership voucher.
- **If the Bootz server is configured to send individual ownership vouchers rather than an OV bundle, the router:**

1. Communicates with the Bootz server using the server information from the bootstrap response for active RP.
2. Sends a new bootstrap request for the standby RP
3. Obtains the ownership voucher information for the standby RP.



CHAPTER 6

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 Status of Hardware Modules, on page 51
- Verify Node Status, on page 52
- Verify Environmental Parameters, on page 53
- Verify Software Version, on page 54
- Verify Firmware Version, on page 54
- Verify Interface Status, on page 57

Verify Status of Hardware Modules

Hardware modules include RPs, 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.

Procedure

show platform

Example:

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

Displays the list of hardware modules detected on the router.

```
Mon Jul  29 10:59:22.163 UTC+00:00
Location  Card Type              HW State      SW State      Config State
------------------------------------------------------------------------------
0/RP0     N560-RSP4-E            OPERATIONAL   OPERATIONAL   NSHUT
0/FT0     A907-FAN-E             OPERATIONAL   N/A           NSHUT
0/PM2     A900-PWR1200-A         OPERATIONAL   N/A           NSHUT
```

Verify Node Status

Each card on the router represents a node. The operational status of the node is verified using the **show platform** command. This command is to be executed independently from both XR and System Admin mode CLIs.

Procedure

Step 1 **show platform**

Example:

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

The **show platform** command when executed from the XR EXEC mode displays the status of XR console running on various RPs and LCs.

Verify that all RPs are listed and their state is OPERATIONAL. This indicates that the XR console is operational on the cards.

Step 2 **admin**

Example:

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

Enters mode.

Step 3 **show platform**

Example:

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

The **show platform** command when executed from the System Admin EXEC mode displays the status of all hardware units like cards (RPs, IMs) and hardware modules (fan trays) on the router.

This is an example for single-chassis system:

```
sysadmin-vm:0_RP0# show platform
Thu Mar  28 08:19:08.640 UTC+00:00
Location  Card Type              HW State      SW State      Config State
------------------------------------------------------------------------------
0/0       NCS4200-1T16G-PS       OPERATIONAL   N/A           NSHUT
0/1       NCS4200-1T16G-PS       OPERATIONAL   N/A           NSHUT
0/2       NCS4200-1T16G-PS       OPERATIONAL   N/A           NSHUT
0/3       NCS4200-1T16G-PS       OPERATIONAL   N/A           NSHUT
0/4       A900-IMA8Z             OPERATIONAL   N/A           NSHUT
0/5       A900-IMA8Z             OPERATIONAL   N/A           NSHUT
0/7       N560-IMA1W             OPERATIONAL   N/A           NSHUT
0/9       N560-IMA2C             OPERATIONAL   N/A           NSHUT
0/10      A900-IMA8Z             OPERATIONAL   N/A           NSHUT
0/11      A900-IMA8Z             OPERATIONAL   N/A           NSHUT
0/12      NCS4200-1T16G-PS       OPERATIONAL   N/A           NSHUT
0/13      NCS4200-1T16G-PS       OPERATIONAL   N/A           NSHUT
0/14      NCS4200-1T16G-PS       OPERATIONAL   N/A           NSHUT
0/15      NCS4200-1T16G-PS       OPERATIONAL   N/A           NSHUT
0/RP0     N560-RSP4-E            OPERATIONAL   OPERATIONAL   NSHUT
0/RP1     N560-RSP4-E            OPERATIONAL   OPERATIONAL   NSHUT
```

```
0/FT0     N560-FAN-H              OPERATIONAL   N/A           NSHUT
0/PM0     A900-PWR1200-A          OPERATIONAL   N/A           NSHUT
0/PM2     A900-PWR1200-A          OPERATIONAL   N/A           NSHUT

sysadmin-vm:0_RP0#
```

Verify that all cards installed on the router are displayed in the result. The software state of LCs/IMs and RPs and the hardware state of FTs and power modulesshould be "OPERATIONAL". Various hardware and software states are listed here.

Hardware states:

- OPERATIONAL—Card is operating normally and is fully functional
- POWERED_ON—Power is on and the card is booting up
- FAILED—Card is powered on but has experienced some internal failure
- PRESENT—Card is in the shutdown state
- OFFLINE—User has changed the card state to OFFLINE. The card is accessible for diagnostics

Software states:

- OPERATIONAL—Software is operating normally and is fully functional
- SW_INACTIVE—Software is not completely operational
- FAILED—Software is operational but the card has experienced some internal failure

Verify Environmental Parameters

The following commands display the environmental parameters. Execute these commands independently from both XR and System Admin mode commands.

Procedure

Step 1 **show environment temperatures**

Example:

```
sysadmin-vm:0_RP0# show environment temperatures
Mon Jul  29 11:12:24.828 UTC+00:00
================================================================================
Location  TEMPERATURE                Value   Crit Major Minor Minor Major  Crit
          Sensor                   (deg C)   (Lo) (Lo)  (Lo)  (Hi)  (Hi)   (Hi)
--------------------------------------------------------------------------------
0/RP0
          QMX Die Temp                55    -40   -30   -20   100   108   112
          Inlet                       34    -40   -30   -20    70    75    85
          FPGA Die                    60    -40   -30   -20    95    98   102
          Outlet                      53    -40   -30   -20    85    90    95
          Humidity                    21    -40   -30   -20    85    95    98
0/FT0
          Fan Inlet                   37    -10    -5     0   100   110   120
```

```
0/PM2
          Inlet Temperature              38    -40   -30   -20    95   100   105
          Outlet Temperature             42    -40   -30   -20    75    80    85
sysadmin-vm:0_RP0#
```

Step 2 **show environment fan**

Example:

```
sysadmin-vm:0_RP0# show environment fan
Mon Jul  29 11:13:30.258 UTC+00:00
================================================================================
                          Fan speed (rpm)
Location   FRU Type           FAN_0   FAN_1   FAN_2   FAN_3   FAN_4   FAN_5
                                                FAN_6   FAN_7   FAN_8   FAN_9   FAN_10  FAN_11

                                                  FAN_12  FAN_13  FAN_14  FAN_15
--------------------------------------------------------------------------------
0/FT0      A907-FAN-E         10298   10369   10288   10351   10330   10373
                                                10351   10252   10341   10348   10273   10316

                                                  13215   13321   16189   16304
```

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.

Procedure

Execute the **show version** command to display the version of Cisco IOS XR software and various software components installed on the router.

Example:

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 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.

Procedure

show hw-module fpd

Example:

```
RP/0/RP0/CPU0:N560_SYSPSV#show hw-module fpd
Wed Mar 13 22:35:40.387 IST
                                                                  FPD Versions
                                                                  ================
Location   Card type        HWver FPD device       ATR Status   Running Programd
-----------------------------------------------------------------------------
0/0        NCS4200-1T16G-PS  0.0   IMFPGA              CURRENT    1.76    1.76
0/1        NCS4200-1T16G-PS  0.0   IMFPGA              CURRENT    1.76    1.76
0/2        NCS4200-1T16G-PS  0.0   IMFPGA              CURRENT    1.76    1.76
0/3        NCS4200-1T16G-PS  0.0   IMFPGA              CURRENT    1.76    1.76
0/4        A900-IMA8Z        0.0   IMFPGA              CURRENT   17.02   17.02
0/5        A900-IMA8Z        0.0   IMFPGA              CURRENT   17.02   17.02
0/7        N560-IMA2C        0.0   IMFPGA              CURRENT    3.04    3.04
0/9        N560-IMA2C        0.0   IMFPGA              CURRENT    3.04    3.04
0/10       A900-IMA8Z        0.0   IMFPGA              CURRENT   17.02   17.02
0/11       A900-IMA8Z        0.0   IMFPGA              CURRENT   17.02   17.02
0/12       NCS4200-1T16G-PS  0.0   IMFPGA              CURRENT    1.76    1.76
0/13       NCS4200-1T16G-PS  0.0   IMFPGA              CURRENT    1.76    1.76
0/14       NCS4200-1T16G-PS  0.0   IMFPGA              CURRENT    1.76    1.76
0/15       NCS4200-1T16G-PS  0.0   IMFPGA              CURRENT    1.76    1.76
0/RP0      N560-RSP4-E       0.0   IOFPGA              CURRENT    0.53    0.53
0/RP0      N560-RSP4-E       0.0   PRIMARY-BIOS        CURRENT    0.14    0.14
0/RP1      N560-RSP4-E       0.0   IOFPGA              CURRENT    0.53    0.53
0/RP1      N560-RSP4-E       0.0   PRIMARY-BIOS        CURRENT    0.14    0.14
0/FT0      N560-FAN-H        0.256 PSOC                CURRENT    2.01    2.01
0/PM0      A900-PWR1200-A    0.0   PriMCU              CURRENT  0.00    0.00
0/PM0      A900-PWR1200-A    0.0   SecMCU              CURRENT  0.00    0.00
0/PM2      A900-PWR1200-A    0.0   PriMCU              CURRENT  0.00    0.00
0/PM2      A900-PWR1200-A    0.0   SecMCU              CURRENT  0.00    0.00
```

Effective Cisco IOS XR Release 7.2.1, the N560-1MA1W interface module is supported on the routers.

```
RP/0/RP1/CPU0:ios#show hw-module fpd
 Tue Jun 23 16:10:04.026 IST
                                                                  FPD Versions
                                                                  ================
Location   Card type             HWver FPD device       ATR Status   Running Programd
-----------------------------------------------------------------------------
0/0        A900-IMA8CS1Z-M       0.0   IMFPGA              CURRENT    1.95    1.95
0/1        A900-IMA8CS1Z-M       0.0   IMFPGA              CURRENT    1.95    1.95
0/2        A900-IMA8CS1Z-M       0.0   IMFPGA              CURRENT    1.95    1.95
0/7        N560-IMA1W            66.32 CFP2-DE-DCO         CURRENT   38.27397 38.27397
0/7        N560-IMA1W            0.0   IMFPGA              CURRENT    1.16    1.16
0/9        N560-IMA2C            0.0   IMFPGA              CURRENT    4.80    4.80
0/10       A900-IMA8Z            0.0   IMFPGA              CURRENT   17.05   17.05
0/11       A900-IMA8Z            0.0   IMFPGA              CURRENT   17.05   17.05
0/RP0      N560-RSP4-E           0.0   ADM                 CURRENT    1.05    1.05
0/RP0      N560-RSP4-E           0.0   IOFPGA              CURRENT    0.56    0.56
0/RP0      N560-RSP4-E           0.0   PRIMARY-BIOS        CURRENT    0.16    0.16
0/RP0      N560-RSP4-E           0.0   SATA                CURRENT  1.30    1.30
0/RP1      N560-RSP4-E           0.0   ADM                 CURRENT    1.05    1.05
0/RP1      N560-RSP4-E           0.0   IOFPGA              CURRENT    0.56    0.56
0/RP1      N560-RSP4-E           0.0   PRIMARY-BIOS        CURRENT    0.16    0.16
0/RP1      N560-RSP4-E           0.0   SATA                CURRENT  1.30    1.30
0/FT0      N560-FAN-H            1.0   PSOC                CURRENT    2.02    2.02
RP/0/RP1/CPU0:ios#
```

Note
Ensure that the CFP2-DCO firmware version is also compatible with Cisco IOS XR Release 7.2.1.

Note
To upgrade firmware on CFP2-DCO, controller optics (R/S/I/P) must be shut down.

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 IO FPGA, IM FPGA, and BIOS.
- 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.
- Programmd—Version of the FPD programmed on the module.

What to do next

- Upgrade the required firmware by using the **upgrade hw-module location all fpd** command in the EXEC mode. For the FPD upgrade to take effect, the router needs a power cycle.

Note BIOS and IOFPGA upgrades require power cycle of the router for the new version to take effect.

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.

Procedure

show ipv4 interface summary

Example:

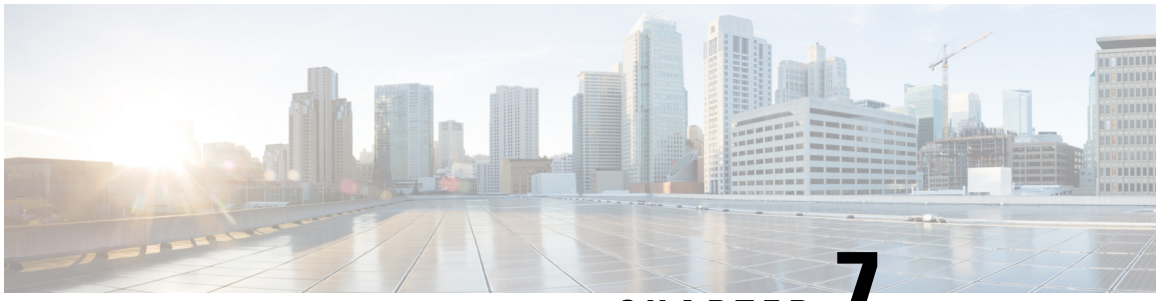
```
RP/0/RP0/CPU0:router#show ipv4 interface summary
```

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.

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 7

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.

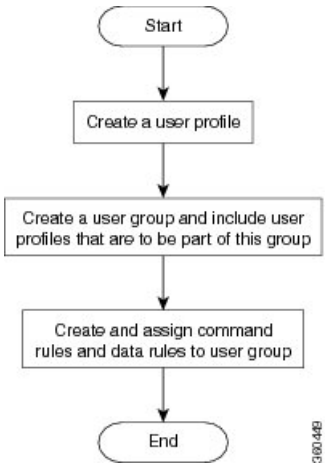
Note If any user on XR is deleted, the local database checks whether there is a first user on System Admin VM.

- 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 6: 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 a User Profile in System Admin VM, on page 60
- Create a User Group in System Admin VM, on page 62
- Create Command Rules, on page 63
- Create Data Rules, on page 66
- Change Disaster-recovery Username and Password, on page 68

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 modeXR 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_RP0sysadmin-vm:0_RP0#config
```

Enters System Admin Config System Admin Configmode.

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 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 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 62.

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 66.

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 62.

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.

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.

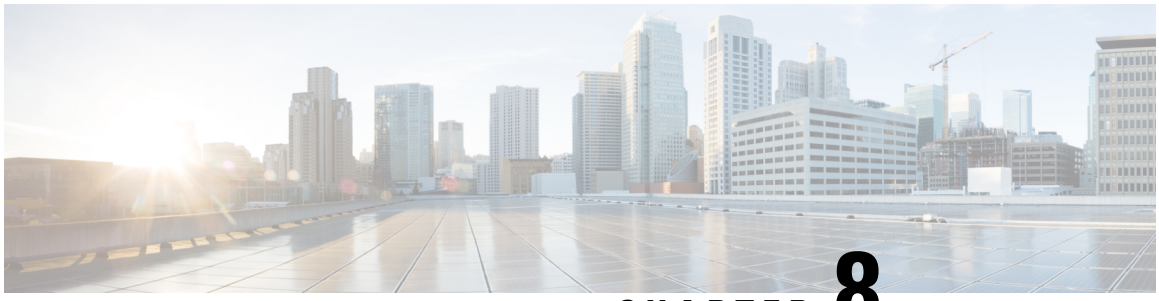
When you need to make use of the disaster recovery username, you need to enter it as *username***@localhost**.

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.



CHAPTER 8

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 71
- View supported software upgrade or downgrade versions, on page 73
- Optional packages, on page 73
- Upgrading Features, on page 76
- Workflow for Install Process, on page 77
- Install Packages, on page 78
- Install Prepared Packages, on page 83
- Uninstall Packages, on page 85

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 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 *ncs560-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 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'll not be able to roll back to, or view, any commits made until the install rollback event. Any new commits made after the install rollback event starts from commit ID '1000000001'.

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 For a successful upgrade from Cisco IOS XR Release 7.2.2 to Cisco IOS XR Release 24.2.21, perform one of the following actions:

- Upgrade to the interim version, say, 7.3.2 and then to 24.2.21.
- Upgrade using USB boot.

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 4: 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.

Optional packages

Cisco IOS XR software images are composed of a base image (ISO) that includes mandatory packages and optional packages (RPMs). Optional RPMs are software packages included in or available for the IOS XR image that provide additional features beyond the mandatory base functionality.

These optional RPMs can be included in the ISO image or provided separately for installation as needed. Optional RPMs enable specific features such as routing protocols such as OSPF, RIP, ISIS, and so on, security features such as k9sec for 802.1X, multicast, telemetry, and others.

Some optional RPMs are included in the base ISO image, while others are available as separate downloadable packages. Operators can add, upgrade, or remove optional RPMs using IOS XR software management commands, allowing flexible customization of the software features on the device.

Table 5: Feature History Table

Feature Name	Release Information	Description
Deprecated insecure protocols in optional RPMs	Release 26.2.1	This feature improves platform security and reduces the threat surface of the Cisco Networking products by restricting insecure protocols such as Telnet, FTP, TFTP, and RCP (deprecated in Cisco IOS XR Release 25.4.1) from Cisco IOS XR base package. In this release, Telnet is added to the optional RPM, `Telnet.rpm`, and FTP, TFTP, and RCP are added to the optional RPM, `IP-Insecure-Apps.rpm .` We recommend you to migrate to secure alternatives such as SSH, SCP, SFTP as Telnet, FTP, TFTP, and RCP protocols arenot supported anymore in the Cisco IOS XR base package.

Starting with Cisco IOS XR Release 26.2.1, the insecure protocols such as Telnet, FTP, TFTP, and RCP are no longer part of Cisco IOS XR base packages. Instead, install the new optional RPMs `Telnet.rpm` to use Telnet features and `IP-Insecure-Apps.rpm` to use FTP and TFTP features.

You must install these optional RPMs only if it is obsolutely required as deprecation warnings continue to be generated once every 30 days as long as Telnet, FTP, TFTP, and RCP configurations are used. The support for these optional RPMs will be completely removed from Cisco IOS XR in future releases such that Telnet, FTP, TFTP, and RCP will no longer be supported in Cisco IOS XR.

To avoid system failures or configuration inconsistencies, we strongly recommend you to migrate to secure protocols such as SSHv2, SFTP, and SCP.

Note Telnet to console is still supported and not impacted by this feature enhancement.

Limitations of optional packages

To improve system security, the absence of optional packages such as `Telnet.rpm` and `IP-Insecure-Apps.rpm` disables certain network management features and restricts insecure protocols.

Without the optional RPMs, `Telnet.rpm` and `IP-Insecure-Apps.rpm`:

- Telnet configuration and execution commands cannot be used.
- Telnet to Management ports is not supported.
- Configuration and execution commands using FTP, TFTP, and RCP are rejected.

- Copy, install, load, or any operations using FTP or TFTP will fail and report errors.
- Systems upgraded with existing insecure configurations may raise configuration inconsistency alarms.

  You must clear inconsistencies before committing any new configuration.

Commands and YANG data models supported using optional RPMs

In Cisco IOS XR Release 26.2.1, you can continue to use the commands and YANG data models for the deprecated Telnet, FTP, and TFTP protocols if you have installed the new `Telnet.rpm` and `IP-Insecure-Apps.rpm` respectively.

The following tables provide the supported configuration and execution commands and the YANG data models for Telnet, FTP, and TFTP in optional RPMs.

- Supported Telnet configuration and execution commands in optional RPMs
- Supported FTP and TFTP configuration and execution commands in optional RPMs
- Supported Telnet, TFTP, and FTP YANG models in optional RPMs

Table 6: Supported FTP and TFTP commands in optional RPMs

FTP config commands	TFTP config commands	FTP exec commands	TFTP exec commands
ip ftp client anonymous-password <>	**ip tftp client dscp** <>	**copy harddisk: ftp:**	**copy harddisk: tftp:**
ip ftp client passive	**ip tftp client dscp** <>	**copy ftp: harddisk:**	**copy tftp: harddisk:**
ip ftp client password *<WORD/clear/encrypted>*	**ip tftp client retries** <>	**copy running-config ftp:**	**copy running-config tftp:**
ip ftp client source-interface <>	**ip tftp client source-interface**<>	**copy ftp: running-config**	**copy tftp: running-config**
ip ftp client username <>	**ip tftp client timeout** <>	**copy xml-schema ftp:**	**copy xml-schema tftp:**
ip ftp client vrf <>	**ip tftp client vrf** *<WORD>*	**install source <ftp-path>**	**install source <tftp-path>**
ftp client anonymous-password <>	**ip tftp server homedir** <>	**install add source <ftp-path>**	**install add source <tftp-path>**
ftp client passive	**tftp client dscp** <>	**install replace <ftp-path>**	**install replace <tftp-path>**
ftp client password *<WORD/clear/encrypted>*	**tftp client retries** <>		
ftp client source-interface <>	**tftp client source-interface** <>		

FTP config commands	TFTP config commands	FTP exec commands	TFTP exec commands
ftp client username <>	**tftp client timeout** <>		
load ftp	**tftp client vrf** <*WORD*>		
	tftp ipv4 server homedir <>		
	tftp ipv6 server homedir <>		
	tftp vrf <WORD> ipv4 server homedir <>		
	tftp vrf <*WORD*>**ipv6 server homedir** <>		
	load tftp		

Table 7: Supported Telnet, TFTP, and FTP YANG models in optional RPMs

Telnet YANG models	FTP and TFTP YANG models	FTP and TFTP YANG models for Install
Cisco-IOS-XR-ipv4-telnet-cfg	Cisco-IOS-XR-um-ftp-tftp-cfg	Cisco-IOS-XR-spirit-install-act.yang
Cisco-IOS-XR-ipv4-telnet-mgmt-cfg	Cisco-IOS-XR-ip-tcp-cfg	
Cisco-IOS-XR-um-telnet-cfg	Cisco-IOS-XR-shellutil-copy-act	
Cisco-IOS-XR-openconfig-system-terminal-ext		

Upgrading Features

Upgrading features is the process of deploying new features and software patches on the router. Feature upgrade is done by installing package files, termed simply, packages. Software patch installation is done 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 mandatory RPM which is a part of the base software version and hence cannot be removed. Optional RPMs such as OSPF can be added and removed as required.

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

- ncs560-mpls-<*package-versison*>-<*release-number*>.x86_64.rpm

- ncs560-isis-<*package-versison*>-<*release-number*>.x86_64.rpm
- ncs560-mcast-<*package-versison*>-<*release-number*>.x86_64.rpm
- ncs560-mgbl-<*package-versison*>-<*release-number*>.x86_64.rpm
- ncs560-bgp-<*package-versison*>-<*release-number*>.x86_64.rpm
- ncs560-ospf-<*package-versison*>-<*release-number*>.x86_64.rpm
- ncs560-mpls-te-rsvp-<*package-versison*>-<*release-number*>.x86_64.rpm
- ncs560-li-<*package-versison*>-<*release-number*>.x86_64.rpm
- ncs560-eigrp-<*package-versison*>-<*release-number*>.x86_64.rpm
- ncs560-k9sec-<*package-versison*>-<*release-number*>.x86_64.rpm

Package and SMU installation is performed using **install** commands. For more information about the install process, see the *Install Packages* section.

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.

For more information on upgrading the system and the RPMs, see *Cisco IOS XR Flexible Packaging Configuration Guide*.

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, you should activate corresponding third-party base package.

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

Note All Cisco RPMs have the platform name in the file name. For example, **ncs560**-sysadmin.

Workflow for Install Process

The workflow for installation and uninstallation processes is depicted in this flowchart.

For installing a package, see Install Packages, on page 78. For uninstalling a package, see Uninstall Packages, on page 85.

Install Packages

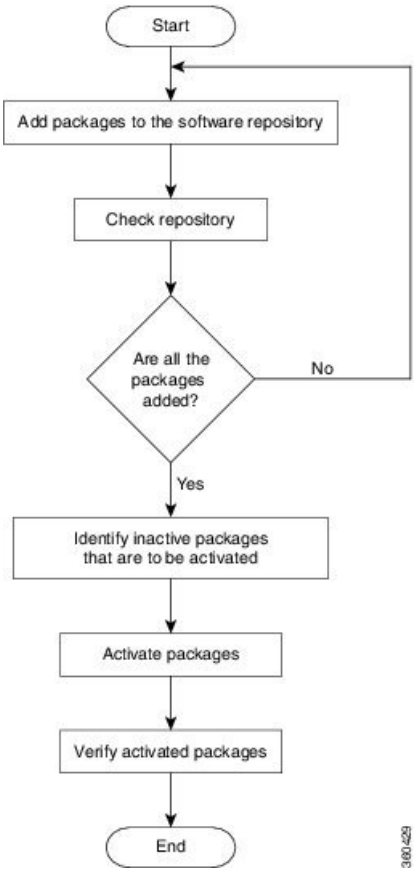
Complete this task to upgrade the system or install a patch. The system upgrade is done using an ISO image file, while the patch installation is done using packages and SMUs. This task is also used to install *.rpm* files. The *.rpm* file contains multiple packages and SMUs that are merged into a single file. The packaging format defines one RPM per component, without dependency on the card type.

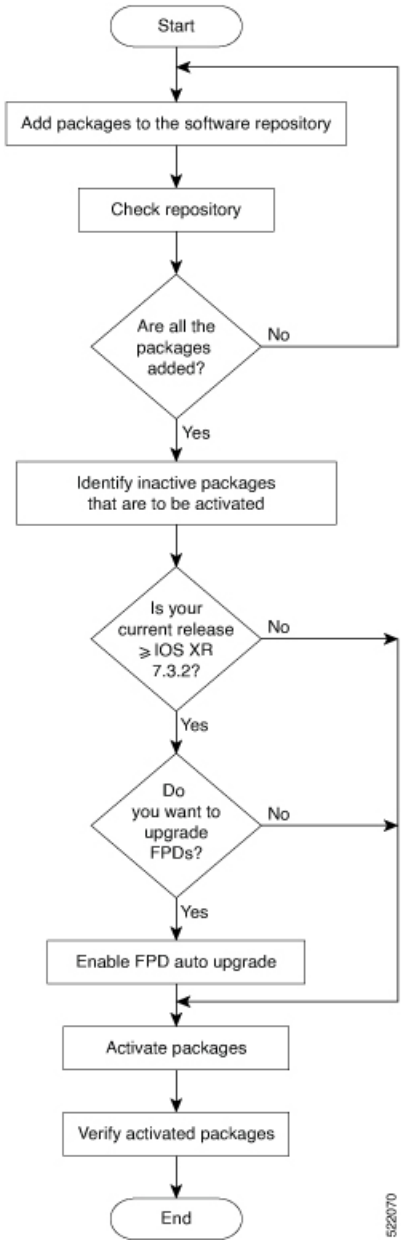
Note

- The System Admin package and XR package can be executed using **install** commands in the System Admin EXEC and XR EXEC mode. All **install** commands are applicable in both these modes.
- Install operation over IPv6 is not supported.

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

Figure 7: Installing Packages Workflow





Before you begin

Note Effective Cisco IOS XR Release 7.3.2, in the XR VM, use the **fpd auto-upgrade enable** command to automatically upgrade the FPDs. For more information, see the *Auto FPD Upgrade* section in the *Upgrading Field-Programmable Device* chapter.

- Review the Install the Latest FPD on the Cisco NCS560 Routers TechNote.

- Configure and connect to the management port. The installable file is accessed through the management port. For details about configuring the management port, see Configure the Management Port.
- Copy the package to be installed either on the router's hard disk or on a network server to which the router has access.

Procedure

Step 1 Execute one of these:

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

Example:

```
RP/0/RP0/CPU0:router# install add source /harddisk:/ncs560-mpls-1.0.0.0-r60023I.x86_64.rpm
 ncs560-mgbl-2.0.0.0-r60023I.x86_64.rpm
RP0
RP/0/RP0/CPU0:router# install add source
/harddisk:/ncs560-mpls-te-rsvp-1.0.0.0-<release-number>.x86_64.rpm
ncs560-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/
RP/0/RP0/CPU0:router# install add source
/harddisk:/ncs560-mpls-1.0.0.0-<release-number>.x86_64.rpm
ncs560-mgbl-2.0.0.0-<release-number>.x86_64.rpm
RP/0/RP0/CPU0:router# install add source
/harddisk:/ncs560-mpls-te-rsvp-1.0.0.0-<release-number>.x86_64.rpm
ncs560-mgbl-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 and 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 as soon as possible.

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
Thu Mar 28 13:29:03.219 IST

The install add operation 36 is 30% complete
RP/0/RP0/CPU0:router#
```

(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
```

For system administration packages, the remaining steps must be performed from the System Admin EXEC mode. Use the **admin** command to enter the System Aadmin EXEC mode.

Step 3 **show install repository**

Example:

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

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 the following:

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

Example:

```
RP/0/RP0/CPU0:router# install activate ncs560-mcast-1.0.0.0-<release-number>.x86_64.rpm
ncs560-mpls-1.0.0.0-<release-number>.x86_64.rpm
```

The *operation_id* is that of the **install add** operation. This command can also be run from System 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. The **install activate** command runs in the background, and the EXEC prompt is returned.

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 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. Activation of some SMUs require a manual reloading 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 executing the **install activate** command. 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.

Step 7 **install commit**

Example:

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

Commits the XR newly active software. To commit both XR and System Admin software, use **install commit system**.

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.

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 Aadmin EXEC mode. The progress of FPD upgrade process can be monitored using the **show hw-module fpd** command in the System Aadmin EXEC mode. Reload the router after the FPD upgrade is completed.

Note Starting Cisco IOS XR Release 7.3.2, if you have enabled the FPD auto-upgrade feature, you do not need to use the **upgrade hw-module location all fpd all** command to upgrade FPDs.

Also, the router automatically reloads after an FPD auto upgrade.

Use the **show hw-module fpd** command to verify the FPD versions.

For more information, see the *Auto FPD Upgrade* section in the *Upgrading Field-Programmable Device* chapter.

- 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, on page 85.

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

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 78.

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

Thu Mar 28 13:29:03.219 IST

The install add operation 36 is 30% complete
RP/0/RP0/CPU0:ios#
```

(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:

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.

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.

Note Starting Cisco IOS XR Release 7.3.2, if you have enabled the FPD auto-upgrade feature, you do not need to use the **upgrade hw-module location all fpd all** command to upgrade FPDs.

Also, the router automatically reloads after an FPD auto upgrade.

Use the **show hw-module fpd** command to verify the FPD versions.

- 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.

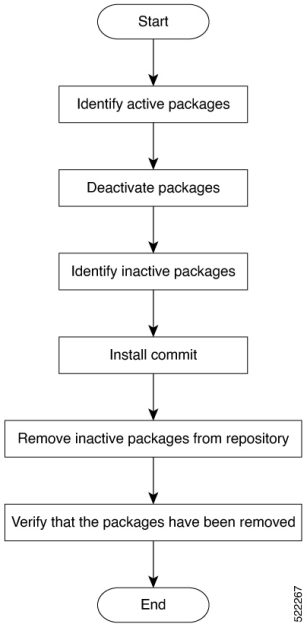
Uninstall Packages

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 8: 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.

Step 2 Execute one of these:

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

Example:

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:

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**

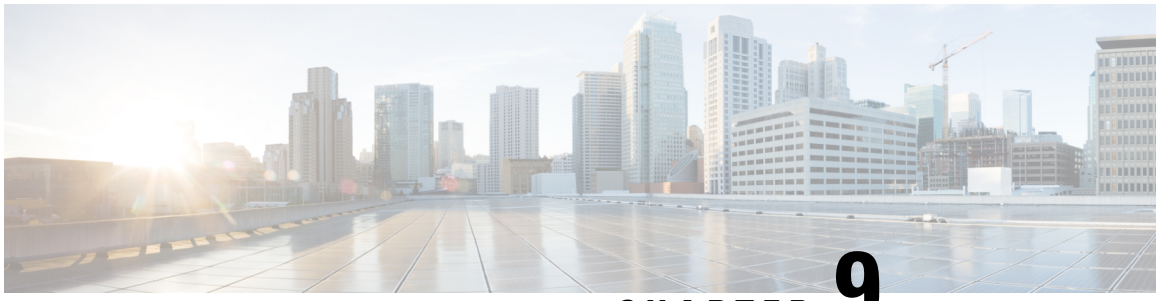
Example:

```
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 9

In Service Software Upgrade

This document contains the following topics:

- Overview, on page 89
- Restrictions and Usage Guidelines, on page 90
- Pre-installation Tasks, on page 91
- ISSU - Single-step Installation, on page 95
- ISSU - Multi-step Installation, on page 98
- Recovering from a Failed ISSU Operation, on page 100
- Installing Packages Using ISSU: Related Commands, on page 100

Overview

In-Service Software Upgrade (ISSU) provides the ability to upgrade the IOS XR 64-bit version on the routers with minimal disruption on the control plane and forwarding plane. ISSU supports upgrading a Cisco IOS XR 64-bit image from a lower to a higher version. ISSU supports Zero Topology Loss (ZTL) and Minimum Packet Loss (MPL). Packet loss is for less than 50 milliseconds unless specified otherwise in the *Release Notes*.

SMUs containing software fixes also can be installed using ISSU. See the corresponding SMU Readme for more information on installing the fixes.

You can perform ISSU installation in a single-step or as multi-step processs. However, you must perform the pre-installation tasks before executing ISSU. During the pre-installation tasks and ISSU execution, V1 refers to the image currently running on the router and V2 refers to the upgraded image.

ISSU execution contains the following phases:

- Prepare phase: The installable files are pre-checked and loaded on the router before activation. This phase is optional.
- Activate phase: The new image (V2) is downloaded to all nodes in the router replacing the old image (V1). This phase can be either run in consequent phases like Load, Run, and Cleanup or by using a one-shot Activate phase.
- Commit phase: In this phase, the ISSU installation is committed (complete with V2) on all nodes.

ISSU supports upgrading the System Admin VM and XR VM individually. Using ISSU, the System Admin VM and XR VM can also be upgraded sequentially. The upgrade sequence is *System Admin ISSU* followed by *IOS XR ISSU*.

Note You cannot upgrade both VMs simultaneously through the ISSU processs.

Note Committing the upgrade from XR VM commits both, the System Admin and XR software. However, committing the upgrade from System Admin VM commits only the System Admin software.

Note When RP1 is the active RP and System Admin VM ISSU is triggered, there is an additional VM switch over compared to performing System Admin VM ISSU from RP0. This is an expected behavior.

Restrictions and Usage Guidelines

Caution When performing an ISSU from Cisco IOS XR Release 7.1.2 to Cisco IOS XR Release 7.6.2 image, during the ISSU upgrade load phase, the process may be aborted due to multiple processes that don't declare on time.

In the XR-7.1.2 image 40G optics isn't supported in the 100G port, so during ISSU load phase V1 in 7.1.2 image (RP0/RP1) maintained the same 100G port but in V2 (RP0/RP1) in 7.6.2, image changed to 40G port because 40G optics was inserted. So you must remove the 40G optics while performing an ISSU operation from XR-7.1.2 to XR-7.6.2 image.

Note When performing an ISSU from Cisco IOS XR Release 7.0.2 to Cisco IOS XR Release 7.3.2 with SR-TE enabled, ensure that you configure RSVP on the router *before* the upgrade. The RSVP configuration ensures that the ISSU operation doesn't fail.

You can remove the RSVP configuration after the upgrade is complete.

Note When performing an ISSU from Cisco IOS XR Release 7.0.2 to Cisco IOS XR Release 7.3.2 with SR-TE enabled, ensure that you configure RSVP on the router *before* the upgrade. The RSVP configuration ensures that the ISSU operation doesn't fail.

You can remove the RSVP configuration after the upgrade is complete.

Note ISSU isn't supported on this router in Cisco IOS XR Release 7.5.1.

Note Before performing the ISSU-SMU deactivation, ensure that you consider the dependencies on the relevant SMUs already present on the router. This consideration avoids conflicts between components during ISSU operation.

- ISSU is supported only from one Extended Maintenance Release (EMR) to another. For more information on types of releases, see Guidelines for Cisco IOS XR Software.
- ISSU-SMUs support activation and deactivation.
- ISSU isn't supported when there's a kernel change.
- ISSU isn't supported with a reload SMU. For more information on SMUs, see IOS XR Software Maintenance Updates (SMUs).

Pre-installation Tasks

Procedure

Step 1 Configure NTP on the XR VM. When NTP is configured, the System Admin VM automatically syncronizes with NTP running on the RP. If the NTP server is not available, configure clock on both, the XR VM and the System Admin VM in configuration mode. Ensure that your clock is set to the correct location and timezone.

Example:

```
RP/0/RP0/CPU0:ios# show ntp associations

   address         ref clock         st  when  poll reach  delay  offset    disp
*~202.153.144.25   10.64.58.51       2   159  1024  377   14.80   0.001   0.019
* sys_peer, # selected, + candidate, - outlayer, x falseticker, ~ configured

RP/0/RP0/CPU0:ios# admin show ntp associations

     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 192.0.4.4       .INIT.          16 u    -  256    0    0.000    0.000   0.000
*192.0.0.4       202.153.144.25   3 u   62  256  377    0.218   -0.034   0.291

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

Step 2 Using the **show redundancy summary** command, ensure that the dual RP systems are synchronized and they are in active and standby roles respectively. The RP status should either be *Final Band* or *Running*.

Example:

```
RP/0/RP0/CPU0:ios# show redundancy summary
    Active Node    Standby Node
    -----------    ------------
     0/RP0/CPU0      0/RP1/CPU0 (Node Ready, NSR:Ready)

RP/0/RP0/CPU0:ios# show platform vm
Node name       Node type       Partner name    SW status       IP address
--------------- --------------- --------------- --------------- ---------------
```

```
0/RP1/CPU0      RP (STANDBY)    0/RP0/CPU0      FINAL Band      192.0.4.4
0/RP0/CPU0      RP (ACTIVE)     0/RP1/CPU0      FINAL Band      192.0.0.4

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

Step 3 Ensure that the firmware on the RP and interface modules (IMs) is upgraded to the latest version. You can upgrade the router cards in a single step by using the **upgrade hw-module location all fpd all** command. Use the **show hw-module location fpd** command to verify the firmware versions.

Example:

```
RP/0/RP0/CPU0:ios# show hw-module fpd
Sun Apr  5 12:47:21.347 CEST
                                                                FPD Versions
                                                                ================
Location   Card type              HWver FPD device       ATR Status   Running Programd
-----------------------------------------------------------------------------------
0/1        A900-IMA8CS1Z-M       0.0   IMFPGA               CURRENT    1.95    1.95
0/5        A900-IMA8Z            0.0   IMFPGA               CURRENT   17.05   17.05
0/9        N560-IMA2C            0.0   IMFPGA               CURRENT    4.80    4.80
0/10       A900-IMA8Z            0.0   IMFPGA               CURRENT   17.05   17.05
0/RP0      N560-RSP4-E           0.0   ADM                  CURRENT    1.05    1.05
0/RP0      N560-RSP4-E           0.0   IOFPGA               CURRENT    0.56    0.56
0/RP0      N560-RSP4-E           0.0   PRIMARY-BIOS         CURRENT    0.16    0.16
0/RP0      N560-RSP4-E           0.0   SATA                 CURRENT    1.30    1.30
0/RP1      N560-RSP4-E           0.0   ADM                  CURRENT    1.05    1.05
0/RP1      N560-RSP4-E           0.0   IOFPGA               CURRENT    0.56    0.56
0/RP1      N560-RSP4-E           0.0   PRIMARY-BIOS         CURRENT    0.16    0.16
0/RP1      N560-RSP4-E           0.0   SATA                 CURRENT    1.30    1.30
0/FT0      N560-FAN-H            1.0   PSOC                 CURRENT    2.02    2.02

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

Step 4 Check the disk storage space on both, System Admin VM and XR VM and ensure that sufficient disk space is available. Remove files such as, show-tech, cores, kernel dumps, manually created text, log, debug information and so on. This example shows verifying the disk storage space for System Admin VM and XR VM on both RPs. You must also verify the disk space on both RPs. If required, verify the disk storage on line cards using the **show media location** command.

Example:

```
RP/0/RP0/CPU0:ios# show media  location all

Media Information for node0_RP1_CPU0.


-----------------------------------------------
Partition                   Size     Used  Percent    Avail
rootfs:                     3.9G     2.3G      63%     1.4G
harddisk:                   5.6G     2.6G      49%     2.8G
log:                        459M     134M      32%     291M
config:                     459M      10M       3%     415M
disk0:                      2.0G      46M       3%     1.8G
-----------------------------------------------------
rootfs: = root file system (read-only)
log: = system log files (read-only)
config: = configuration storage (read-only)

Media Information for node0_RP0_CPU0.


-----------------------------------------------
Partition                   Size     Used  Percent    Avail
```

```
rootfs:                       3.9G     2.3G      63%     1.4G
harddisk:                     5.6G     3.6G      68%     1.8G
log:                          459M     167M      40%     259M
config:                       459M      11M       3%     414M
disk0:                        2.0G      81M       5%     1.8G
-----------------------------------------------------
rootfs: = root file system (read-only)
log: = system log files (read-only)
config: = configuration storage (read-only)
RP/0/RP0/CPU0:ios#ad show media  location all

************************************************************
Location : 0/RP1
************************************************************
-----------------------------------------------------------------------
Partition                     Size     Used  Percent    Avail
-----------------------------------------------------------------------
rootfs:                       2.4G     616M      28%     1.7G
harddisk:                     5.6G     2.8G      53%     2.5G
log:                          459M     119M      28%     306M
config:                       459M     4.1M       1%     421M
disk0:                       1011M     1.7M       1%     940M
install:                      5.6G     2.2G      41%     3.1G
install:/tmp                  5.6G     2.2G      41%     3.1G
install:/cache                5.6G     2.2G      41%     3.1G
rootfs:/install/tmp           5.6G     2.2G      41%     3.1G
-----------------------------------------------------------------------
rootfs:  = root file system (read-only)
log:     = system log files (read-only)
config:  = configuration storage (read-only)
install: = install repository (read-only)

************************************************************
Location : 0/RP0
************************************************************
-----------------------------------------------------------------------
Partition                     Size     Used  Percent    Avail
-----------------------------------------------------------------------
rootfs:                       2.4G     616M      28%     1.7G
harddisk:                     5.6G     2.8G      53%     2.5G
log:                          459M     128M      31%     297M
config:                       459M     4.0M       1%     421M
disk0:                       1011M     1.9M       1%     940M
install:                      5.6G     2.2G      41%     3.1G
install:/tmp                  5.6G     2.2G      41%     3.1G
install:/cache                5.6G     2.2G      41%     3.1G
rootfs:/install/tmp           5.6G     2.2G      41%     3.1G
-----------------------------------------------------------------------
rootfs:  = root file system (read-only)
log:     = system log files (read-only)
config:  = configuration storage (read-only)
install: = install repository (read-only)

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

Step 5 Populate the repository with RPMs and SMUs. Pick and install individual RPMs and SMUs, one by one, or make a tarball and install it, or break it down with multiple tarballs.

Example:

```
RP/0/RP0/CPU0:ios# install add source harddisk:/ 702_ncs560_V2.tar

Fri Apr  3 11:02:01.208 UTC
Apr 03 11:02:03 Install operation 7 started by sanity:
```

```
 install add source harddisk:/ 702_ncs560_V2.tar
Apr 03 11:02:10 Install operation will continue in the background
RP/0/RP0/CPU0:tb2-ncs560-pe1#

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

Fri Apr  3 11:02:41.567 UTC
User sanity, Op Id 7
install add
702_ncs560_V2.tar
The install add operation 7 is 60% complete
Add validate request sent to sysadmin

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

Step 6 Use the **show install repository command** to validate that packages, images, or SMUs are populated properly in the router's repository. There should be a one to one relationship between V1 and V2 images and SMUs. For example, if you install a SMU on V1, you also need the corresponding V2 version in the repository to execute ISSU.

Example:

```
RP/0/RP0/CPU0:ios# show install repository | in mini

ncs560-mini-x-6.6.3  <— V1 iso image currently running

ncs560-mini-x-7.0.2  <— V2 iso image to upgrade

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

If you are using the **install activate issu ncs560-mini-x-7.0.2** command in the *sysadmin* prompt, the system automatically extracts the sysadmin and host ISO files and continues with the sysadmin ISSU operation. You must extract the XR IOS file separately for the XR ISSU to take place.

Step 7 Depending on the version of the image, extract the ISO image in System Admin VM or XR VM.

Example:

For XR VM:

```
RP/0/RP0/CPU0:ios# install extract ncs560-mini-x-7.0.2

Wed Apr  1  23:07:10.119 UTC+00:00
RP/0/RP0/CPU0:ios#

RP/0/RP0/CPU0:ios# show install repository | in xr
ncs560-xr-6.6.3
ncs560-xr-7.0.2   extracted V2 image
RP/0/RP0/CPU0:ios#
```

For System Admin VM:

```
sysadmin-vm:0_RP0# install extract ncs560-mini-x-7.0.2

Wed Apr  1  23:07:10.119 UTC+00:00
result Wed Apr  1 23:07:11 2020 Install operation 1 (install extract) started by user
'sanity' will continue asynchronously.

sysadmin-vm:0_RP0#

sysadmin-vm:0_RP0# show install repository all | in sys|host
ncs560-sysadmin-6.6.3
```

```
host-6.6.3
ncs560-sysadmin-7.0.2   extracted V2 image
host-7.0.2              extracted V2 image


sysadmin-vm:0_RP0#
```

ISSU - Single-step Installation

To perform ISSU on the router, ISSU must be performed for both, System Admin VM and XR VM in the following order—first, System Admin VM (under the admin prompt) and then XR VM (under the IOS-XR prompt).

This section shows how to perform ISSU on the router with the IOS XR 64-bit image in a single step. You can either upgrade the system or install a patch in a single step. The system upgrade is done using an ISO image file, while the patch installation is done using packages and SMUs.

Note Depending on whether you are installing a System Admin package or a XR package, execute these commands in the System Admin EXEC mode or XR EXEC mode respectively

Before you begin

Note Effective Cisco IOS XR Release 7.3.2, in the XR VM, use the **fpd auto-upgrade enable** command to automatically upgrade the FPD. For more information, see the *Auto FPD Upgrade* section in the *Upgrading Field-Programmable Device* chapter.

- Copy the package to be installed either on the router's hard disk or on a network server to which the router has access.
- Ensure that the dual route processor (RP) system with standby is in "is ready" state.

Procedure

Step 1 Use the **install activate issu** command to activate the ISSU installation in XR VM or System Admin VM in a single step.

Example:

For System Admin VM:

```
sysadmin-vm:0_RP0# install activate issu ncs560-sysadmin-release-version
host-release-version
```

For XR VM:

```
RP/0/RP0/CPU0:ios# install activate issu ncs560-xr-release-version
package-name-release-version
```

Step 2 Use the **install commit** command to commit the newly-active software.

Example:

For System Admin VM:

```
sysadmin-vm:0_RP0# install commit
```

For XR VM:

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

Example

This example shows how to perform System Admin VM upgrade using ISSU and how to verify the installation using the show commands.

```
sysadmin-vm:0_RP0# show install repository all | in sys|host
ncs560-sysadmin-6.6.3
host-6.6.3
ncs560-sysadmin-7.0.2
host-7.0.2


sysadmin-vm:0_RP0:# install activate issu ncs560-sysadmin-7.0.2 host-7.0.2

This install operation will result in admin VMs reload
 Do you want to proceed [yes/no]: yes
 Proceeding with operation
 result Wed Oct 31 21:12:21 2018 Install operation 2 (install prepare and activate issu)
 started by user 'root' will continue asynchronously.

sysadmin-vm:0_RP0:#

!# Monitoring the progress of the installation.
!# The installation may take up to 30 minutes.

sysadmin-vm:0_RP0:# show install request
User root, Op Id 2
 install prepare issu
 host-7.0.2
 This operation is 40% complete
 Waiting for agents to complete host prepare ..

sysadmin-vm:0_RP0:#


sysadmin-vm:0_RP0:# show install request
User root, Op Id 2
 install activate issu
 ISSU stage Phase1
 ncs560-sysadmin-7.0.2
   Node 0/RP0 [RP] : 90% of current state is completed
   Node 0/RP1 [RP] : 90% of current state is completed

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

```
!# Message after successful completion. Admin VM will reload after this message. . There
should be no packet drop.
0/RP0/ADMIN0:Oct 31 21:27:53.260 : inst_mgr[5019]: %INFRA-INSTMGR-2-OPERATION_SUCCESS :
Install operation 2 completed successfully

!# Verifying the active package

sysadmin-vm:0_RP1# show install active summary
Active Packages: 1
        ncs560-sysadmin-7.0.2 version=7.0.2 [Boot image]

!# Verifies the image previously committed
sysadmin-vm:0_RP1# show install commit summary
Committed Packages: 1
        ncs560-sysadmin-6.6.3 version=6.6.3 [Boot image]

!# Commits the latest image

sysadmin-vm:0_RP1# install commit


result Wed Oct 31 21:32:58 2018 Install operation 3 (install commit) started by user 'root'
 will continue asynchronously.
sysadmin-vm:0_RP1#

0/RP1/ADMIN0:Oct 31 21:33:02.061 : inst_mgr[6913]:%INFRA-INSTMGR-2-OPERATION_SUCCESS :
Install operation 3 completed successfully
Wed Oct 31 21:33:02 2018 Install operation 3 completed successfully.
```

This example shows how to perform XR VM upgrade using ISSU and verify the installation using the show commands.

```
!# Verify the active packages
RP/0/RP0/CPU0:ios# show install active summary
Active Packages: 1
        ncs560-xr-6.6.3 version=6.6.3 [Boot image]
        !# Performing ISSU Installation

RP/0/RP0/CPU0:ios# install activate issu ncs560-xr-7.0.2

RP/0/RP0/CPU0:ios# install activate issu ncs560-xr-7.0.2
Oct 31 21:48:14 Install operation 10 started by root:
install activate issu ncs560-xr-7.0.2
Oct 31 21:48:14 Package list:
Oct 31 21:48:14 ncs560-xr-7.0.2

This install operation will start the issu, continue?
[yes/no]:[yes] yes Oct 31 21:49:13 Install operation will continue in the background
RP/0/RP0/CPU0:ios#

RP/0/RP1/CPU0:ios# show issu summary
 Fri Apr  3 12:33:48.324 UTC
INSTALL Operation ID    : Operation 10 Started at Fri Apr  3 11:56:11 2020
ISSU Progress           : 100.0%
Total ISSU Time         : 00:17:28
ISSU Type               : IMAGE(V1-6.6.3/V2-7.0.2)

Phase           Start-Time      End-Time                State
--------------------------------------------------------------------------------
Prepare         11:49:40        11:55:26                Completed
Load            11:56:11        12:06:42                Completed
```

```
Run             12:30:37        12:30:48                Completed
Cleanup         12:32:46        12:33:46                Completed
--------------------------------------------------------------------------------
Current Status          : ISSU Orchestration Successfully Completed

Setup Information       : Single Chassis
ISSU Ready/Not Ready    : 0 / 0

Node ISSU readiness per rack per slot
Key: Ready - 'Y', Not ready - 'N', Primary node - '*', Complete - '-'

Rack 0   RP0   RP1
```

What to do next

Note Starting Cisco IOS XR Release 7.3.2, if you have enabled the FPD auto-upgrade feature, you do not need to use the **upgrade hw-module location all fpd all** command to upgrade FPDs.

Use the **show hw-module fpd** command to verify the FPD versions.

For more information, see the *Auto FPD Upgrade* section in the *Upgrading Field-Programmable Device* chapter.

ISSU - Multi-step Installation

To perform ISSU on the router, ISSU must be performed for both, System Admin VM and XR VM in the following order—first, System Admin VM (under the admin prompt) and then XR VM (under the IOS-XR prompt).

This section shows how to perform ISSU on the router with IOS XR 64-bit in multiple steps.

Note Depending on whether you are installing a System Admin package or a XR package, execute these commands in the System Admin EXEC mode or XR EXEC mode respectively.

Note You should update the System Admin VM first and then update the XR VM. IOS XR 64-bit ISSU fails if the System Admin VM is not updated first.

Before you begin

Note Effective Cisco IOS XR Release 7.3.2, in the XR VM, use the **fpd auto-upgrade enable** command to automatically upgrade the FPD. For more information, see the *Auto FPD Upgrade* section in the *Upgrading Field-Programmable Device* chapter.

- Copy the package to be installed either on the router's hard disk or on a network server to which the router has access.
- Ensure that the dual route processor (RP) system with standby is in "is ready" state.
- Before starting any operation that triggers reload or switchover, monitor your system install health by running the **show install health** command in System Admin mode. By running this command, you can verify that show commands such as **show install repository** display the correct output.

Procedure

Step 1 (Optional) Prepare the installable files by using the **install prepare issu** *package_name* command. During the prepare phase, pre-activation checks are performed and the components of the installable files are loaded on to the router setup.

Example:

For XR VM:

```
RP/0/RP0/CPU0:ios# install prepare issu ncs560-xr-release-version
```

Step 2 Use the **install activate issu load** command to start the *Load* phase.

Example:

For XR VM:

```
RP/0/RP0/CPU0:ios# install activate issu load ncs560-xr-release-version
```

This step downloads the new image (V2) to all nodes in the router. The new image is checked for compatibility to ensure that the router can be upgraded. At the start of the Load phase, the router configuration mode is locked, and you cannot perform any configuration on the router until ISSU completes the phase. At the end of this stage, all standby nodes run V2 and all active nodes (including all line cards) still run the original software images (V1).

Step 3 Use the **install activate issu run** command to start the *Run* phase.

Example:

For XR VM:

```
RP/0/RP0/CPU0:ios# install activate issu run
```

This phase starts the version switch from V1 to V2. All the packages that have been prepared are activated to make the package configurations active on the router.

Step 4 Use the **install activate issu cleanup** command to start the *Cleanup* phase.

Example:

For XR VM:

```
RP/0/RP0/CPU0:ios# install activate issu cleanup
```

This phase initiates shutdown of VMs with previous versions after running the activation. The Cleanup phase concludes the ISSU process and the new software runs on all nodes in the system.

Step 5 Use the **install commit** command to commit the newly-active software.

Example:

For XR VM:

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

Recovering from a Failed ISSU Operation

While performing the ISSU operation in IOS XR, the operation can abort due to multiple reasons, such as aa software issue. In such cases, you must perform a cleanup operation to restore the system to the stable state.

Abort an ISSU operation manually after the *Prepare* phase

Procedure

Step 1 To abort and clean up an ISSU operation manually after the *Prepare* phase, use the following command in the XR VM:

```
RP/0/RP0/CPU0:ios# install prepare clean
```

Step 2 To abort the ISSU operation manually after the *Load* phase, use the following command in the XR VM:

```
RP/0/RP0/CPU0:ios# install activate issu abort
```

To restore the IOS XR version on the standby RP, use the following command in the XR VM:

```
RP/0/RP0/CPU0:ios# install activate issu abort cleanup
```

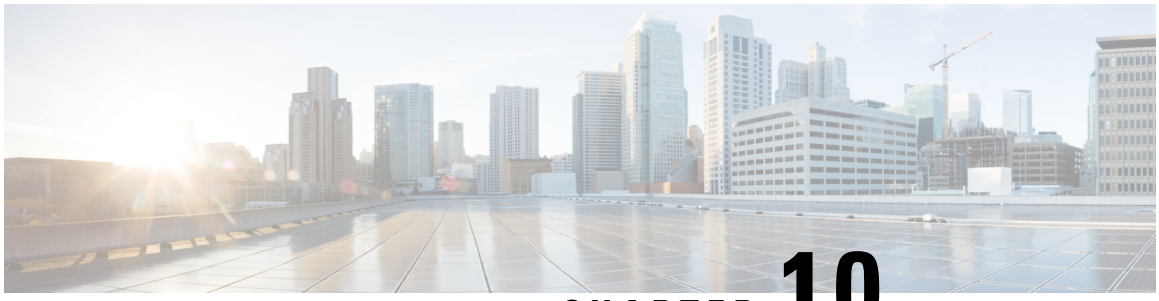
Note
The install activate issu abort cleanup command must be run only from current active RP.

Note
You cannot abort the ISSU operation after the *Run* phase.

Installing Packages Using ISSU: Related Commands

Command	Purpose
show issu	Displays the state or status of the ISSU operation. Effective with Cisco IOS XR Release 7.0.2, this command is also supported for System Admin VM ISSU.
admin show issu	Displays the status of the ISSU operation.
show install request	Displays the progress of the ISSU installation.

Command	Purpose
admin show install request	Displays details about the progress of the install operation.
show install active	Displays the active packages on the system.
install prepare clean	Clears the existing prepared image. If there is a failure in the Prepare phase, run this command to clear the prepared image.
show issu milestone summary	Displays information about the ISSU milestones.



CHAPTER 10

Upgrading Field-Programmable Device

An FPD is a field programmable logic device which contains non-volatile, re-programmable memory to define its internal wiring and functionality. The contents of this non-volatile memory are called the FPD image or FPD firmware. Over the lifespan of an FPD, FPD firmware images may need upgrades for bug fixes or functionality improvements. These upgrades are performed in the field with minimum system impact.

- Prerequisites for FPD Image Upgrades, on page 103
- Overview of FPD Image Upgrade Support, on page 103
- FPD upgrade service, on page 104

Prerequisites for FPD Image Upgrades

- You must install the FPD pie before you install the SMUs or Service Packs. If you install the SMU or Service Packs before the FPD pie, the FPDs on the line card may not upgrade. In such cases, you must remove the SMUs and Service Packs and reload the router.
- When upgrading ASR 9006 router from a 32-bit IOS-XR to 64-bit IOS-XR, the corresponding RSPs and line cards must have ROMMON version 18.33 or later and a compatible FPD version (18.30 or later). If upgrading from a 32-bit IOS XR release, an intermediate upgrade (for example, IOS XR 7.1.2) may be required to update the line card ROMMON and FPD before proceeding to newer 64-bit IOS XR releases.

Overview of FPD Image Upgrade Support

An FPD image is used to upgrade the software on an FPD.

FPD versions must be compatible with the Cisco IOS XR software that is running on the router; if an incompatibility exists between an FPD version and the Cisco IOS XR software, the device with the FPGA may not operate properly until the incompatibility is resolved.

Note Downgrade of FPDs is not recommended.

Note FPD auto upgrade is not supported on this router.

FPD upgrade service

The main tasks of the FPD upgrade service are:

- Check FPD image version to decide if a specific firmware image needs an upgrade or not.
- Automatic FPD Image Upgrade (if enabled).
- Manual FPD Image Upgrade using the **upgrade hw-module fpd** command.
- Invoke the appropriate device driver with a name of the new image to load.

An FPD image package is used to upgrade FPD images. The **install activate** command is used to place the FPD binary files into the expected location on the boot devices.

Supported Upgrade Methods

Method	Remarks
Manual Upgrade	Upgrade using CLI, force upgrade supported.
Auto Upgrade	Upgrade using install SMU activation or during image upgrade. User can enable/disable auto upgrade feature.

Important FPD auto upgrade is not supported on this router in releases before Cisco IOS XR Release 7.3.2.

Determining Upgrade Requirement

Use the **show hw-module fpd** command to determine if an FPD upgrade is required. Check for NEED UPGD in the Status column.

Example

```
Router: #show hw - module fpd

Wed Dec 14 07:08:08.424 UTC

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

0/0        NC55-18H18F           1.0   MIFPGA               NEED UPGD   7.01      7.01

0/0        NC55-18H18F           1.0   Bootloader           CURRENT     1.14      1.14
0/0        NC55-18H18F           1.0   IOFPGA               CURRENT     0.07      0.07
0/0        NC55-18H18F           1.0   SATA-M600-MCT        CURRENT     0.23      0.23
```

Use the **show fpd package** command to find out which FPGAs are supported with your current software release and minimum hardware requirements for each module.

Manual FPD Upgrade

Manual FPD upgrade is performed using the **upgrade hw-module fpd** command. All cards or all FPGA in a card can be upgraded. If reload is required to activate FPD, the upgrade should be complete. Interface module (IMs) and RSPs cannot be reloaded during the process of the FPD upgrade.

FPD upgrade is transaction-based:

- Each fpd upgrade CLI execution is one transaction.
- Only one transaction is allowed at any given time.
- One transaction may include one or many FPD upgrades

The **force** option can be used to forcibly upgrade the FPD (regardless of whether it is required or not). It triggers all FPDs to be upgraded or downgraded. The **force** option can also be used to downgrade or upgrade the FPGAs even after the version check. However, the **force** option must be used cautiously and only to recover a component from a failed upgrade.

Note

- Sometimes, FPDs can have primary and backup images.
- Force FPD upgrade with **upgrade hw-module location all fpd all force** command affects forwarding over BVI interface. You must reload involved locations to recover.
- The use of the **force** option when performing an FPD upgrade is not recommended except under explicit direction from Cisco engineering or TAC for a one-time purpose only.
- FPD upgrade should be performed in Admin mode only.
- A new FPD upgrade should be issued only when previous FPD upgrades have been completed on the same FPD with the following syslog message:

```
RP/0/RP0/CPU0:May 10 10:11:44.414 UTC: fpd-serv[205]: %INFRA-FPD_Manager-1-UPGRADE_ALERT
 : FPD Upgrade Completed (use "show hw-module fpd" to check upgrade status)
```

How to Upgrade FPD Images

- Migrate the software to a later Cisco IOS XR software release.
- Swap IMs or RSPs from a system running a different Cisco IOS XR software release.
- Insert a new IM or RSP.

In the event of an FPD incompatibility with your card, you might receive the following error message:

```
LC/0/0/CPU0:Jul 5 03:00:18.929 UTC: optics_driver[220]: %L2-OPTICS-3-BAD_FPGA_IMAGE :
Detected bad MI FPGA image programmed in MI FPGA SPI flash in 0/0/CPU0 location: Failed to
 validate meta data CRC
LC/0/0/CPU0:Jul 5 03:00:19.019 UTC: optics_driver[220]: %L2-OPTICS-3-BACKUP_FPGA_LOADED :
Detected Backup FPGA image running on 0/0/CPU0 - primary image corrupted (@0x8c = 0x44)
RP/0/RP0/CPU0:Jul 5 03:00:48.987 UTC: fpd-serv[301]: %PKT_INFRA-FM-3-FAULT_MAJOR : ALARM_MAJOR
 :FPD-NEED-UPGRADE :DECLARE :0/0:
```

Upgrades to the Cisco IOS XR software might result in an FPD incompatibility. Ensure that you perform the FPD upgrade procedure and resolve all incompatibilities, for the cards to function properly.

Note The use of the **force** option when performing a FPD upgrade is not recommended except under explicit direction from Cisco engineering or TAC for a one-time purpose only.

Before you begin

- Use the Admin console to run upgrade commands instead of the common console. Also, ensure that you reload from the Admin console and not from the common console.
- The FPD upgrade procedure is performed while the card is online. At the end of the procedure the card must be reloaded before the FPD upgrade is complete. To reload the card, you can use the **hw-module location <location> reload** command in Admin mode, during the next maintenance window. The upgrade procedure is not complete until the card is reloaded.
- During the FPD upgrade, you *must not* do the following:
  - Reload, perform an online insertion and removal (OIR) of a line card (LC), or power down the chassis. Doing so may cause the node to enter an unusable state.
  - Press **Ctrl-C** if the console appears to hang without any output. Doing so may abort the upgrade.
- If you are not sure whether a card requires an FPD upgrade, you can install the card and use the **show hw-module fpd** command to determine if the FPD image on the card is compatible with the currently running Cisco IOS XR software release.

Configuration Examples for FPD Image Upgrade

The following examples indicates the use of commands associated with the FPD image upgrade procedure.

show fpd package Command Output: Example

Use the **show fpd package** command in System Admin EXEC mode to find out which IMs and RSPs are supported with your current Cisco IOS XR software release, which FPD image package you need for each IM or RSP, and what the minimum hardware requirements are for each module. If multiple FPD images are available for your card, they are listed as Subtype fpga2, fpga3, and so on.

Note The FPD name used in the FPD Description column of the output of the `show fpd package` command includes the last ten characters of DCO-PID. Depending on the slot and port numbers, the FPD name is appended with DCO_0, DCO_1, or DCO_2. For example, the FPD names for `CFP2-WDM-D-1HL` in port 0 and port 1 are `-WDM-D-1HL_DCO_0` and `WDM-D-1HL_DCO_1` respectively.

The following example shows a sample output from the **show fpd package** command:

```
sysadmin_vm:0_RP0# show fpd package
Fri Oct 16 07:20:50.021 UTC

================================ ================================================
                                            Field Programmable Device Package
```

Card Type	FPD Description	Req Reload	SW Ver	Min Req SW Ver	Min Req Board Ver
A900-IMA8CS1Z-CC	IMFPGA	YES	1.96	1.96	0.0
A900-IMA8CS1Z-M	IMFPGA	YES	1.96	1.96	0.0
A900-IMA8Z	IMFPGA	YES	17.05	17.05	0.0
A900-IMA8Z-CC	IMFPGA	YES	17.05	17.05	0.0
A900-IMA8Z-L	IMFPGA	YES	1.45	1.45	0.0
A900-PWR1200-A	DCA-PriMCU(A)	NO	0.11	0.11	0.0
	DCA-SecMCU(A)	NO	1.04	1.04	0.0
A900-PWR1200-D	LIT-PriMCU(A)	NO	2.04	0.04	0.0
	LIT-SecMCU(A)	NO	1.23	1.23	0.0
A907-FAN-E	PSOC(A)	NO	1.65	1.65	0.0
	PSOC(A)	NO	1.66	1.66	0.4
N560-4-FAN-H	PSOC(A)	NO	177.02	177.02	0.0
N560-4-FAN-H-CC	PSOC(A)	NO	177.02	177.02	0.0
N560-4-PWR-FAN	PSOC(A)	NO	177.08	177.08	0.0
N560-4-PWR-FAN-CC	PSOC(A)	NO	177.08	177.08	0.0
N560-4-RSP4	ADM(A)	NO	1.06	1.06	0.0
	IOFPGA(A)	YES	0.64	0.64	0.0
	PRIMARY-BIOS(A)	YES	0.17	0.17	0.0
	SATA(A)	YES	2.10	2.10	0.0
N560-4-RSP4-CC	ADM(A)	NO	1.06	1.06	0.0
	IOFPGA(A)	YES	0.64	0.64	0.0
	PRIMARY-BIOS(A)	YES	0.17	0.17	0.0
	SATA(A)	YES	2.10	2.10	0.0
N560-4-RSP4E	ADM(A)	NO	1.06	1.06	0.0
	IOFPGA(A)	YES	0.64	0.64	0.0
	PRIMARY-BIOS(A)	YES	0.17	0.17	0.0
	SATA(A)	YES	2.10	2.10	0.0
N560-4-RSP4E-CC	ADM(A)	NO	1.06	1.06	0.0
	IOFPGA(A)	YES	0.64	0.64	0.0
	PRIMARY-BIOS(A)	YES	0.17	0.17	0.0
	SATA(A)	YES	2.10	2.10	0.0
N560-FAN-H	PSOC(A)	NO	2.02	2.02	0.0
N560-IMA-8Q/4L	IMFPGA	YES	1.08	1.08	0.0
N560-IMA1W	CFP2-D-DCO	NO	38.273	38.273	0.0
	CFP2-DE-DCO	NO	38.273	38.273	0.0
	CFP2-DET-DCO	NO	38.273	38.273	0.0
	CFP2-DETS-DCO	NO	38.273	38.273	0.0
	CFP2-DS-DCO	NO	38.273	38.273	0.0
	CFP2-DS100-DCO	NO	38.273	38.273	0.0
	IMFPGA	YES	1.24	1.24	0.0

```
N560-IMA2C-CC        IMFPGA                    YES     5.04    5.04    0.0
--------------------------------------------------------------------------------
N560-PWR1200-D-E     QCS-PriMCU(A)             NO      1.82    1.82    0.0
                     QCS-SecMCU(A)             NO      1.84    1.84    0.0
--------------------------------------------------------------------------------
N560-RSP4            ADM(A)                    NO      1.06    1.06    0.0
                     IOFPGA(A)                 YES     0.64    0.64    0.0
                     PRIMARY-BIOS(A)           YES     0.17    0.17    0.0
                     SATA(A)                   YES     2.10    2.10    0.0
--------------------------------------------------------------------------------
N560-RSP4-E          ADM(A)                    NO      1.06    1.06    0.0
                     IOFPGA(A)                 YES     0.64    0.64    0.0
                     PRIMARY-BIOS(A)           YES     0.17    0.17    0.0
                     SATA(A)                   YES     2.10    2.10    0.0
--------------------------------------------------------------------------------
NCS560-IMA2C         IMFPGA                    YES     5.04    5.04    0.0
--------------------------------------------------------------------------------
NCS560-IMA2C-DD      IMFPGA                    YES     1.24    1.24    0.0
--------------------------------------------------------------------------------
NCS560-IMA2C-L       IMFPGA                    YES     1.24    1.24    0.0
```

upgrade hw-module fpd Command Output: Example

Use the **upgrade hw-module fpd** command to upgrade the FPD image. The upgrade can be executed for all FPDs or for specific FPDs that need an upgrade. To upgrade all FPDs, use **upgrade hw-module fpd all location all** command. To upgrade a specific FPD image type, use the FPD subtype value in the **upgrade hw-module fpd** command.

```
RP/0/RP0/CPU0:ios# upgrade hw-module location 0/RP0 fpd ADM
Wed Oct 28 07:46:49.805 UTC
upgrade command issued (use "show hw-module fpd" to check upgrade status)
RP/0/RP0/CPU0:ios#RP/0/RP0/CPU0:Oct 28 07:46:51.949 UTC: optics_driver[222]:
%PKT_INFRA-FM-2-FAULT_CRITICAL : ALARM_CRITICAL :OPTICS RX POWER LANE-0 LOW ALARM :CLEAR :
 Optics0/11/0/1:  Optics0/11/0/1
0/RP0/ADMIN0:Oct 28 07:46:54.154 UTC: fpdserv[4899]: %INFRA-FPD_Manager-1-UPGRADE_ALERT :
Upgrade for the following FPDs has been committed:
0/RP0/ADMIN0:Oct 28 07:46:54.154 UTC: fpdserv[4899]: %INFRA-FPD_Manager-1-UPGRADE_ALERT :
Location           FPD name             Force
0/RP0/ADMIN0:Oct 28 07:46:54.154 UTC: fpdserv[4899]: %INFRA-FPD_Manager-1-UPGRADE_ALERT :
================================================
0/RP0/ADMIN0:Oct 28 07:46:54.154 UTC: fpdserv[4899]: %INFRA-FPD_Manager-1-UPGRADE_ALERT :
0/RP0              ADM                  FALSE
0/RP0/ADMIN0:Oct 28 07:46:59.203 UTC: control_driver[3690]: %INFRA-FPD_Driver-6-UPGRADE_RESULT
 : Upgrade completes 20 percent for fpd ADM@location 0/RP0.
RP/0/RP0/CPU0:ios#0/RP0/ADMIN0:Oct 28 07:47:09.204 UTC: control_driver[3690]:
%INFRA-FPD_Driver-6-UPGRADE_RESULT : Upgrade completes 70 percent for fpd ADM@location
0/RP0.
0/RP0/ADMIN0:Oct 28 07:47:10.854 UTC: control_driver[3690]: %INFRA-FPD_Driver-1-UPGRADE_ALERT
 : FPD ADM@0/RP0 image programming completed with UPGRADE DONE state Info: [image 1.05 to
1.06 version]
0/RP0/ADMIN0:Oct 28 07:47:10.855 UTC: control_driver[3690]: %INFRA-FPD_Driver-1-UPGRADE_ALERT
 : FPD ADM @location 0/RP0 FPD upgraded and activated!
0/RP0/ADMIN0:Oct 28 07:47:10.857 UTC: shelf_mgr[3705]: %INFRA-SHELF_MGR-6-CARD_SW_OPERATIONAL
 : Card: 0/RP0 software state going to Operational
0/RP0/ADMIN0:Oct 28 07:47:10.857 UTC: shelf_mgr[3705]: %INFRA-SHELF_MGR-6-CARD_HW_OPERATIONAL
 : Card: 0/RP0 hardware state going to Operational
RP/0/RP0/CPU0:Oct 28 07:47:17.931 UTC: fpd-serv[393]: %INFRA-FPD_Manager-1-UPGRADE_ALERT :
  FPD Upgrade Completed(use "show hw-module fpd" to check upgrade status)
0/RP0/ADMIN0:Oct 28 07:47:19.155 UTC: fpdserv[4899]: %INFRA-FPD_Manager-1-UPGRADE_ALERT :
 FPD Upgrade Completed(use "show hw-module fpd" to check upgrade status)
RP/0/RP0/CPU0:ios#
```

show platform Command Output: Example

Use the **show platform** command to verify that the IM is up and running.

```
RP/0/RP1/CPU0:router# show platform
Tue Oct 20 04:42:31.936 UTC
Node              Type                         State             Config state
--------------------------------------------------------------------------------
0/0/CPU0          A900-IMA8CS1Z-M              OPERATIONAL       NSHUT
0/1/CPU0          A900-IMA8CS1Z-M              OPERATIONAL       NSHUT
0/2/CPU0          A900-IMA8CS1Z-M              OPERATIONAL       NSHUT
0/4/CPU0          A900-IMA8Z-L                 OPERATIONAL       NSHUT
0/5/CPU0          A900-IMA8Z-L                 OPERATIONAL       NSHUT
0/7/CPU0          N560-IMA1W                   OPERATIONAL       NSHUT
0/9/CPU0          N560-IMA2C-DD                OPERATIONAL       NSHUT
0/10/CPU0         A900-IMA8Z                   OPERATIONAL       NSHUT
0/11/CPU0         A900-IMA8Z-L                 OPERATIONAL       NSHUT
0/RP0/CPU0        N560-RSP4-E(Standby)         UNKNOWN           NSHUT
0/RP1/CPU0        N560-RSP4-E(Active)          IOS XR RUN        NSHUT
0/FT0/CPU0        N560-FAN-H                   OPERATIONAL       NSHUT
0/PM2/CPU0        A900-PWR1200-A               OPERATIONAL       NSHUT
```

Auto FPD Upgrade

Table 8: Feature History Table

Feature Name	Release Information	Feature Description
Auto FPD Upgrade	Release 7.3.2	This functionality enables automatic upgrade and reload for field-programmable devices (FPDs) whenever the Cisco IOS XR image has a newer FPD version. This functionality upgrades all route processors and line card FPDs simultaneously while displaying upgrade triggers on the console.

Effective Cisco IOS XR Release 7.3.2, you can enable automatic upgrade of FPD by using the “fpd auto-upgrade enable” command.

Note Automatic upgrade of FPD is not supported in Cisco IOS XR Release 7.4.1.

To automatically upgrade all FPDs, use:

```
RP/0/RP0/CPU0:IOS(config)#fpd auto-upgrade enable
```

To reload the interface modules following the fpd auto-upgrade, use:

```
RP/0/RP0/CPU0:IOS(config)#fpd auto-reload enable
```

Limitations and Usage Guidelines

Limitations

- FPD auto-upgrade should be enabled only in the XR VM and *not* in the System Admin VM.

- With auto-upgrade enabled, if any card is in RELOAD REQUIRED state, auto-upgrade is re-triggered during any SSO or FPD-serv process restart.
- When an interface module (IM) or route processor (RP) is in RELOAD REQUIRED state and auto-upgrade is enabled, FPD upgrades are triggered again.
- With auto-upgrade enabled, if line card is inserted, an auto-upgrade is triggered. During this phase optics alarms are generated. If auto-reload is not enabled, you must reload the line cards manually to clear these alarms.
- SATA allows you to upgrade or downgrade when an FPD version change is available. Therefore, when auto-upgrade is enabled, the system automatically downgrades if lower versions are available. This behavior is specific only to SATA FPDs.
- FPD auto-reload is applicable for line cards only. Line cards are automatically reloaded after the fpd auto-upgrade process is completed.
- You must disable auto-upgrade during XR ISSU; otherwise, the router goes into a state where redundancy cannot be achieved. In this case, standby RP must be reloaded to achieve redundancy.
-
- TimingICs do not support **auto fpd upgrade** on NCS5500 Series Routers as the TimingIC requires a card reload immediately after upgrade. For the same reason, the TimingICs are not upgraded if the user specifies **location all** in the **auto fpd upgrade** command. To upgrade a TimingIC FPD, specify the FPD name along with the card location. For example, **upgrade hw-module fpd TimngIC-A location 0/RP0/CPU0**

Usage Guidelines—Online Insertion of Line Cards

When a line card with a lower FPD version is inserted, one of the following scenarios apply:

- If fpd auto-upgrade and auto-reload are enabled, and a new line card is inserted, the system upgrades the line card FPDs automatically with the latest FPDs and reloads the line cards.
- If fpd auto-upgrade and auto-reload are both disabled, no action is required.
- If fpd auto-upgrade is enabled and auto-reload is disabled, the following alarms are displayed on the console:

```
RP/0/RP1/CPU0:Jun 1 10:05:46.095 UTC: optics_driver[231]: %PKT_INFRA-FM-3-FAULT_MAJOR
: ALARM_MAJOR :OPTICS SUPPORTED_ERROR :DECLARE : Optics0/5/0/6: Optics0/5/0/6
RP/0/RP1/CPU0:Jun 1 10:05:46.096 UTC: optics_driver[231]: %PKT_INFRA-FM-2-FAULT_CRITICAL
 : ALARM_CRITICAL :OPTICS NOT SUPPORTED :DECLARE : Optics0/5/0/6: Optics0/5/0/6
```

  You must reload the line cards manually to clear these alarms

Usage Guidelines—Online Insertion of RPs

When fpd auto-upgrade is enabled and a new RP is inserted, the system upgrades the RP FPDs automatically with the latest FPDs.

Note RPs are not reloaded automatically. You must manually reload the RP or chassis for the latest FPD version to reflect.

Note Reload of active RPs and line cards impacts the network traffic.

```
RP/0/RP0/CPU0:IOS# admin
Mon Jun 28 17:00:39.340 UTC

sysadmin-vm:0_RP1# hw-module location 0/RP1 reload
Mon Jun  28 17:00:52.178 UTC+00:00
Reload hardware module ? [no,yes] yes
#result Card graceful reload request on 0/RP1 succeeded.
RP/0/RP0/CPU0:IOS#

RP/0/RP0/CPU0:ios# show hw-module fpd
Fri Jun  4 10:08:01.784 UTC

Auto-upgrade:Enabled
                                                                          FPD Versions
                                                                          ==============
Location   Card type             HWver FPD device       ATR Status    Running   Programd
-------------------------------------------------------------------------------------------
0/1        N560-IMA2C-DD         0.0   IMFPGA               CURRENT     1.27      1.27
0/3        N560-IMA2C            0.0   IMFPGA               CURRENT     5.01      5.01
0/5        A900-IMA8CS1Z-M       0.0   IMFPGA               CURRENT     1.98      1.98
0/RP0      N560-4-RSP4           0.0   ADM                  CURRENT     1.06      1.06
0/RP0      N560-4-RSP4           0.0   IOFPGA               CURRENT     0.64      0.64
0/RP0      N560-4-RSP4           0.0   PRIMARY-BIOS         CURRENT     0.18      0.18
0/RP0      N560-4-RSP4           0.0   SATA                 CURRENT     2.10      2.10
0/RP1      N560-4-RSP4           0.0   ADM                  CURRENT     1.06      1.06
0/RP1      N560-4-RSP4           0.0   IOFPGA               CURRENT     0.64      0.64
0/RP1      N560-4-RSP4           0.0   PRIMARY-BIOS         CURRENT     0.19      0.19
0/RP1      N560-4-RSP4           0.0   SATA                 CURRENT     2.10      2.10
0/FT0      N560-4-PWR-FAN        0.1   PSOC                 CURRENT   177.08    177.08
0/FT1      N560-4-FAN-H          0.1   PSOC                 CURRENT   177.02    177.02
0/FT2      N560-4-FAN-H          0.1   PSOC                 CURRENT   177.02    177.02
RP/0/RP0/CPU0:ios#
```

Table 9: Action Required on FPDs After Auto Upgrade

FPD	Action Required
IOFPGA	Manual reload required
ADM	Upgraded version available immediately
PRIMARY-BIOS	Manual reload required
SATA	Upgraded version available immediately
PSOC	Upgraded version available immediately
IMFPGA	Manual reload required, if auto-reload is not configured

Configuring Auto FPD During System Upgrade

In case of Software upgrade (without ISSU), configure the **fpd auto-upgrade enable** command. All the FPDs are automatically upgraded in the currently installed image (V1). After the upgrade, the router automatically

reloads and comes up with the new image (V2) with the upgraded FPDs already running. No additional reloads are required.

Note System reloads are part of the SU process, therefore you can disable the FPD auto reload functionality by using the **fpd auto-reload disable** command.

1. **Enable FPD auto-upgrade**

```
RP/0/RP0/CPU0:IOS#conf
RP/0/RP0/CPU0:IOS(config)#fpd auto-upgrade enable
RP/0/RP0/CPU0:IOS#commit
```

2. **Check for FPD Versions**

```
RP/0/RP0/CPU0:Router#show hw-module fpd
Mon Jun 28 21:41:19.187 UTC

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

-------------------------------------------------------------------------------------
0/2        NCS4200-1T16G-PS      0.0   IMFPGA              CURRENT     1.98      1.98
0/3        A900-IMA8CS1Z-M       0.0   IMFPGA              CURRENT     1.95      1.95
0/4        A900-IMA8Z            0.0   IMFPGA              CURRENT    17.05     17.05
0/5        A900-IMA8Z-L          0.0   IMFPGA              CURRENT     1.48      1.48
0/8        NCS4200-1T16G-PS      0.0   IMFPGA              CURRENT     1.98      1.98
0/9       N560-IMA1W           66.32 CFP2-DS-DCO         CURRENT    38.27397  38.27397
0/9        N560-IMA1W            0.0   IMFPGA              CURRENT     1.28      1.28
0/15       NCS4200-1T16G-PS      0.0   IMFPGA              CURRENT     1.98      1.98
0/RP0      N560-RSP4             0.0   ADM                 CURRENT     1.06      1.06
0/RP0      N560-RSP4             0.0   IOFPGA              CURRENT     0.64      0.64
0/RP0      N560-RSP4             0.0   PRIMARY-BIOS        CURRENT     0.19      0.19
0/RP0      N560-RSP4             0.0   SATA                CURRENT     1.30      1.30
0/RP1      N560-RSP4             0.0   ADM                 CURRENT     1.05      1.05
0/RP1      N560-RSP4             0.0   IOFPGA              CURRENT     0.64      0.64
0/RP1      N560-RSP4             0.0   PRIMARY-BIOS        CURRENT     0.19      0.19
0/RP1      N560-RSP4             0.0   SATA                CURRENT     1.30      1.30
0/FT0      N560-FAN-H            1.0   PSOC                CURRENT     2.02      2.02

RP/0/RP0/CPU0:Router#
```

3. **Check that Auto Upgrades are Triggered for FPDs with Newer Versions Available**

Note At this step, all RSP, IMs, and fan FPD upgrades are initiated and completed. All cards are upgraded *before* the router reloads.

```
RP/0/RP1/CPU0:UUT-RSP4# copy tftp://<ncs560-mini-x-7.3.2.iso> harddisk:/
RP/0/RP0/CPU0:IOS#install add source harddisk: ncs560-mini-x-7.3.2.iso
ncs560-mcast-2.0.0.0-r732.x86_64.rpm ncs560-mgbl-2.0.0.0-r732.x86_64.rpm
ncs560-mpls-1.0.0.0-r732.x86_64.rpm
RP/0/RP0/CPU0:IOS#install ncs560-mini-x-7.3.2 ncs560-mcast-2.0.0.0-r732.x86_64
ncs560-mgbl-2.0.0.0-r732.x86_64 ncs560-mpls-1.0.0.0-r732.x86_64
RP/0/RP0/CPU0:IOS#install commit

RP/0/RP0/CPU0:ROUTER# install activate ncs560-mini-x-7.3.2.28I
Mon Jun 28 21:30:17.673 UTC
2021-06-28 21:30:20 Install operation 31 started by root123:
  install activate pkg ncs560-mini-x-7.3.2.28I
2021-06-28 21:30:20 Package list:
2021-06-28 21:30:20     ncs560-mini-x-7.3.2.28I
RP/0/RP0/CPU0:Jun 28 21:32:41.204 UTC: sdr_instmgr[1213]: %PKT_INFRA-FM-6-FAULT_INFO :
INSTALL-IN-PROGRESS :DECLARE :0/RP0/CPU0: INSTALL_IN_PROGRESS Alarm : being DECLARED for
 the system
This install operation will reload the system, continue?
 [yes/no]:[yes] yes
2021-06-28 21:33:01 Install operation will continue in the background
 ...
RP/0/RP0/CPU0:ROUTER#RP/0/RP0/CPU0:Jun 28 21:41:40.910 UTC: fpd-serv[168]:
%PKT_INFRA-FM-3-FAULT_MAJOR : ALARM_MAJOR :FPD-NEED-UPGRADE :DECLARE :0/RP0:
RP/0/RP0/CPU0:Jun 28 21:41:41.159 UTC: fpd-serv[168]: %PKT_INFRA-FM-3-FAULT_MAJOR :
ALARM_MAJOR :FPD-NEED-UPGRADE :CLEAR :0/RP0:
0/RP0/ADMIN0:Jun 28 21:41:42.565 UTC: control_driver[3205]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD SATA@0/RP0 image programming completed with
UPGRADE DONE state Info: [SDD firmware upgraded from 1.30 to 2.10]
0/RP0/ADMIN0:Jun 28 21:41:42.566 UTC: control_driver[3205]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD SATA @location 0/RP0 FPD upgraded and activated!

0/RP0/ADMIN0:Jun 28 21:41:42.570 UTC: shelf_mgr[3220]:
%INFRA-SHELF_MGR-6-CARD_SW_OPERATIONAL : Card: 0/RP0 software state going to Operational

0/RP0/ADMIN0:Jun 28 21:41:42.570 UTC: shelf_mgr[3220]:
%INFRA-SHELF_MGR-6-CARD_HW_OPERATIONAL : Card: 0/RP0 hardware state going to Operational

RP/0/RP0/CPU0:Jun 28 21:41:42.486 UTC: fpd-serv[168]: %PKT_INFRA-FM-3-FAULT_MAJOR :
ALARM_MAJOR :FPD-NEED-UPGRADE :DECLARE :0/RP1:
0/RP1/ADMIN0:Jun 28 21:41:44.182 UTC: control_driver[3220]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD SATA@0/RP1 image programming completed with
UPGRADE DONE state Info: [SDD firmware upgraded from 1.30 to 2.10]
0/RP1/ADMIN0:Jun 28 21:41:44.182 UTC: control_driver[3220]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD SATA @location 0/RP1 FPD upgraded and activated!

0/RP1/ADMIN0:Jun 28 21:41:48.905 UTC: control_driver[3220]:
%INFRA-FPD_Driver-6-UPGRADE_RESULT : Upgrade completes 20 percent for fpd ADM@location
0/RP1.
0/RP1/ADMIN0:Jun 28 21:41:48.905 UTC: control_driver[3220]:
%INFRA-FPD_Driver-6-UPGRADE_RESULT : Upgrade completes 50 percent for fpd
PRIMARY-BIOS@location 0/RP1.
0/RP1/ADMIN0:Jun 28 21:42:10.160 UTC: control_driver[3220]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD PRIMARY-BIOS@0/RP1 image programming completed
with UPGRADE DONE state Info: [  Upgrade Complete ]
0/RP1/ADMIN0:Jun 28 21:42:10.161 UTC: control_driver[3220]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD PRIMARY-BIOS @location 0/RP1 upgrade completed.

0/RP1/ADMIN0:Jun 28 21:42:11.060 UTC: control_driver[3220]:
```

```
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD ADM@0/RP1 image programming completed with UPGRADE
 DONE state Info: [image 1.05 to 1.06 version]
0/RP1/ADMIN0:Jun 28 21:42:11.061 UTC: control_driver[3220]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD ADM @location 0/RP1 FPD upgraded and activated!

0/RP0/ADMIN0:Jun 28 21:42:11.062 UTC: shelf_mgr[3220]:
%INFRA-SHELF_MGR-6-CARD_SW_OPERATIONAL : Card: 0/RP1 software state going to Operational

0/RP0/ADMIN0:Jun 28 21:42:11.062 UTC: shelf_mgr[3220]:
%INFRA-SHELF_MGR-6-CARD_HW_OPERATIONAL : Card: 0/RP1 hardware state going to Operational

RP/0/RP0/CPU0:Jun 28 21:45:14.615 UTC: fpd_imfpga[121]: %INFRA-FPD_Driver-6-UPGRADE_RESULT
 : Upgrade completes 50 percent for fpd IMFPGA@location 0/5.
RP/0/RP0/CPU0:Jun 28 21:45:14.616 UTC: fpd_imfpga[121]: %INFRA-FPD_Driver-6-UPGRADE_RESULT
 : Upgrade completes 50 percent for fpd IMFPGA@location 0/3.
RP/0/RP0/CPU0:Jun 28 21:48:24.763 UTC: fpd_imfpga[121]: %INFRA-FPD_Driver-6-UPGRADE_RESULT
 : Upgrade completes 90 percent for fpd IMFPGA@location 0/5.

RP/0/RP0/CPU0:ROUTER#RP/0/RP0/CPU0:Jun 28 21:48:43.929 UTC: fpd_imfpga[121]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD IMFPGA@0/5 image programming completed with
UPGRADE DONE state Info: [DONE ]
RP/0/RP0/CPU0:Jun 28 21:48:43.940 UTC: fpd_imfpga[121]: %INFRA-FPD_Driver-1-UPGRADE_ALERT
 : FPD IMFPGA @location 0/5 upgrade completed.
0/RP0/ADMIN0:Jun 28 21:48:45.354 UTC: shelf_mgr[3220]:
%INFRA-SHELF_MGR-6-CARD_HW_OPERATIONAL : Card: 0/5 hardware state going to Operational

0/RP0/ADMIN0:Jun 28 21:48:45.354 UTC: shelf_mgr[3220]:
%INFRA-SHELF_MGR-6-CARD_SW_OPERATIONAL : Card: 0/RP0 software state going to Operational

0/RP0/ADMIN0:Jun 28 21:48:45.354 UTC: shelf_mgr[3220]:
%INFRA-SHELF_MGR-6-CARD_HW_OPERATIONAL : Card: 0/RP0 hardware state going to Operational
P/0/RP0/CPU0:Jun 28 21:50:21.630 UTC: fpd_imfpga[121]: %INFRA-FPD_Driver-1-UPGRADE_ALERT
 : FPD IMFPGA@0/3 image programming completed with UPGRADE DONE state Info: [DONE ]
RP/0/RP0/CPU0:Jun 28 21:50:21.642 UTC: fpd_imfpga[121]: %INFRA-FPD_Driver-1-UPGRADE_ALERT
 : FPD IMFPGA @location 0/3 upgrade completed.
0/RP0/ADMIN0:Jun 28 21:50:23.056 UTC: shelf_mgr[3220]:
%INFRA-SHELF_MGR-6-CARD_HW_OPERATIONAL : Card: 0/3 hardware state going to Operational

0/RP0/ADMIN0:Jun 28 21:50:23.056 UTC: shelf_mgr[3220]:
%INFRA-SHELF_MGR-6-CARD_SW_OPERATIONAL : Card: 0/RP0 software state going to Operational

0/RP0/ADMIN0:Jun 28 21:50:23.057 UTC: shelf_mgr[3220]:
%INFRA-SHELF_MGR-6-CARD_HW_OPERATIONAL : Card: 0/RP0 hardware state going to Operational
```

4. **Check the RP FPD Versions and FPD Status**

   When the router is operational after the reload, all the RP, IMs, and fan FPDs are upgraded to the latest FPD versions.

```
sysadmin-vm:0_RP1# show hw-module fpd
Mon Jun 28 21:51:20.187 UTC

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

-------------------------------------------------------------------------------------------
0/2        NCS4200-1T16G-PS      0.0   IMFPGA               CURRENT     1.98      1.98

0/3        A900-IMA8CS1Z-M       0.0   IMFPGA               CURRENT     1.98      1.98

0/4        A900-IMA8Z            0.0   IMFPGA               CURRENT    17.05     17.05
```

```
0/5         A900-IMA8Z-L          0.0    IMFPGA               CURRENT     1.49       1.49
0/8         NCS4200-1T16G-PS      0.0    IMFPGA               CURRENT     1.98       1.98
0/9        N560-IMA1W           66.32 CFP2-DS-DCO          CURRENT    38.27397  38.27397
0/9         N560-IMA1W            0.0    IMFPGA               CURRENT     1.28       1.28
0/15        NCS4200-1T16G-PS      0.0    IMFPGA               CURRENT     1.98       1.98
0/RP0       N560-RSP4             0.0    ADM                  CURRENT     1.06       1.06
0/RP0       N560-RSP4             0.0    IOFPGA               CURRENT     0.64       0.64
0/RP0       N560-RSP4             0.0    PRIMARY-BIOS         CURRENT     0.19       0.19
0/RP0       N560-RSP4             0.0    SATA                 CURRENT     2.10       2.10
0/RP1       N560-RSP4             0.0    ADM                  CURRENT     1.05       1.05
0/RP1       N560-RSP4             0.0    IOFPGA               CURRENT     0.64       0.64
0/RP1       N560-RSP4             0.0    PRIMARY-BIOS         CURRENT     0.19       0.96
0/RP1       N560-RSP4             0.0    SATA                 CURRENT     2.10       2.10
0/FT0       N560-FAN-H            1.0    PSOC                 CURRENT     2.02       2.02
RP/0/RP0/CPU0:ROUTER#
```

Configuring Auto FPD During ISSU

ISSU occurs in two phases—in the System Admin VM and then in the XR VM.

In the System Admin VM mode, first execute the **fpd auto-upgrade enable** command. This configuration causes the FPDs on the route processor (RP) to automatically upgrade to the latest versions.

In the XR VM mode, you *must* disable the FPD auto upgrade. This is because IM FPGA upgrades are not automatically triggered as part of ISSU.

Note Less than 50ms traffic loss is expected during ISSU.

Note During both Admin and XR ISSU, always disable the auto reload functionality by using the **fpd auto-reload disable** command.

1. **Enable FPD auto-upgrade in XR VM**

```
RP/0/RP0/CPU0:IOS#conf
RP/0/RP0/CPU0:IOS(config)#fpd auto-upgrade enable
RP/0/RP0/CPU0:IOS#commit
```

2. **Disable FPD auto-reload in XR VM**

```
RP/0/RP0/CPU0:IOS#conf
RP/0/RP0/CPU0:IOS(config)#fpd auto-reload disable
RP/0/RP0/CPU0:IOS#commit
```

3. Check for FPD Versions

```
RP/0/RP0/CPU0:iso#show hw-module fpd
Sun Aug  1 19:36:13.869 UTC

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

--------------------------------------------------------------------------------------------
0/11       NCS4200-8T-PS         0.0   IMFPGA              CURRENT    17.05     17.05

0/RP0      NCS4216-RSP-800       0.0   ADM                 CURRENT     1.06      1.06

0/RP0      NCS4216-RSP-800       0.0   IOFPGA              CURRENT     0.78      0.78

0/RP0      NCS4216-RSP-800       0.0   PRIMARY-BIOS        CURRENT     0.21      0.21

0/RP0      NCS4216-RSP-800       0.0   SATA                CURRENT     2.20      2.20

0/RP1      NCS4216-RSP-800       0.0   ADM                 CURRENT     1.06      1.06

0/RP1      NCS4216-RSP-800       0.0   IOFPGA              CURRENT     0.78      0.78

0/RP1      NCS4216-RSP-800       0.0   PRIMARY-BIOS        CURRENT     0.21      0.21

0/RP1      NCS4216-RSP-800       0.0   SATA                CURRENT     2.20      2.20

0/FT0      NCS4216-F2B-FAN       1.0   PSOC                CURRENT     2.02      2.02

0/PM0      A900-PWR1200-A        0.1   DCA-PriMCU          CURRENT     0.13      0.13

0/PM0      A900-PWR1200-A        0.1   DCA-SecMCU          CURRENT     2.03      2.03

0/PM1      A900-PWR1200-A        0.1   DCA-PriMCU          CURRENT     0.13      0.13

0/PM1      A900-PWR1200-A        0.1   DCA-SecMCU          CURRENT     2.03      2.03
RP/0/RP0/CPU0:iso#
```

4. Perform sysadmin ISSU Upgrade

Check that FPD upgrades are triggered as part of ISSU:

```
RP/0/RP0/CPU0:IOS#admin
sysadmin-vm:0_RP0# install extract ncs560-mini-x-7.3.2
Mon Aug  2  04:54:17.250 UTC+00:00
result Mon Aug  2 04:54:18 2021 Install operation 1 (install extract) started by user
 'cafyauto' will continue asynchronously.
sysadmin-vm:0_RP0# 0/RP0/ADMIN0:Aug  2 04:58:47.990 UTC: inst_mgr[4350]:
%INFRA-INSTMGR-2-OPERATION_SUCCESS : Install operation 1 completed successfully
Mon Aug  2 04:58:47 2021 Install operation 1 completed successfully.
sysadmin-vm:0_RP0# install activate issu  host-7.3.2 ncs560-sysadmin-7.3.2
Mon Aug  2  05:26:31.442 UTC+00:00
op-initiated true
operation-id 2
result start-success
id 2
sysadmin-vm:0_RP0# 2021-08-02 05:26:31 Admin install operation 2 started
sysadmin-vm:0_RP0# 2021-08-02 05:26:31 install activate issu host-7.3.2
ncs560-sysadmin-7.3.2
```

```
sysadmin-vm:0_RP0# 2021-08-02 05:26:35 Install operation 2 started: ISSU prepare
install activate issu host-7.3.2 ncs560-sysadmin-7.3.2
sysadmin-vm:0_RP0# 0/RP0/ADMIN0:Aug  2 05:30:22.441 UTC: inst_mgr[4350]:
%INFRA-INSTMGR-2-OPERATION_SUCCESS : Install operation 2 completed successfully
2021-08-02 05:30:22 Install operation 2 completed successfully
sysadmin-vm:0_RP0# 2021-08-02 05:30:23 Admin install operation 3 started
sysadmin-vm:0_RP0# 2021-08-02 05:30:23 install activate issu host-7.3.2
ncs560-sysadmin-7.3.2
sysadmin-vm:0_RP0# 2021-08-02 05:30:25 Install operation 3 started: ISSU activate
install activate issu host-7.3.2 ncs560-sysadmin-7.3.2
sysadmin-vm:0_RP0#
sysadmin-vm:0_RP0# 2021-08-02 05:31:10 Admin VM of node 0/RP1 will now reload as part
 of ISSU operation 3
0/RP0/ADMIN0:Aug  2 05:31:17.314 UTC: shelf_mgr[3212]: %INFRA-SHELF_MGR-4-VM_RELOAD :
 Reloading VM on 0/RP1
0/RP0/ADMIN0:Aug  2 05:32:12.842 UTC: shelf_mgr[3212]:
%INFRA-SHELF_MGR-6-CARD_SW_OPERATIONAL : Card: 0/RP1 software state going to Operational

RP/0/RP0/CPU0:Aug  2 05:33:05.763 UTC: fpd-serv[419]: %PKT_INFRA-FM-3-FAULT_MAJOR :
ALARM_MAJOR :FPD-NEED-UPGRADE :DECLARE :0/RP1:

sysadmin-vm:0_RP0# 2021-08-02 05:33:28 Admin VM of node 0/RP0 will now reload as part
 of ISSU operation 3
RP/0/RP0/CPU0:Aug  2 05:35:44.331 UTC: fpd-serv[419]: %PKT_INFRA-FM-3-FAULT_MAJOR :
ALARM_MAJOR :FPD-NEED-UPGRADE :DECLARE :0/RP0:
0/RP1/ADMIN0:Aug  2 05:36:01.213 UTC: inst_mgr[6351]: %INFRA-INSTMGR-2-OPERATION_SUCCESS
 : Install operation 3 completed successfully
0/RP0/ADMIN0:Aug  2 05:36:29.242 UTC: control_driver[3920]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : Auto fpd triggered for FPD SATA upgrade @ location
 0/RP0 please wait until all fpd upgrades are done
0/RP0/ADMIN0:Aug  2 05:36:29.545 UTC: control_driver[3920]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD SATA@0/RP0 image programming completed with
UPGRADE DONE state Info: [SDD firmware upgraded from 1.30 to 2.10]
0/RP0/ADMIN0:Aug  2 05:36:29.545 UTC: control_driver[3920]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD SATA @location 0/RP0 FPD upgraded and activated!

0/RP1/ADMIN0:Aug  2 05:36:29.557 UTC: shelf_mgr[4032]:
%INFRA-SHELF_MGR-6-CARD_SW_OPERATIONAL : Card: 0/RP0 software state going to Operational

0/RP1/ADMIN0:Aug  2 05:36:29.557 UTC: shelf_mgr[4032]:
%INFRA-SHELF_MGR-6-CARD_HW_OPERATIONAL : Card: 0/RP0 hardware state going to Operational

0/RP1/ADMIN0:Aug  2 05:36:29.667 UTC: control_driver[4015]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : Auto fpd triggered for FPD SATA upgrade @ location
 0/RP1 please wait until all fpd upgrades are done
0/RP1/ADMIN0:Aug  2 05:36:29.981 UTC: control_driver[4015]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD SATA@0/RP1 image programming completed with
UPGRADE DONE state Info: [SDD firmware upgraded from 1.30 to 2.10]
0/RP1/ADMIN0:Aug  2 05:36:29.981 UTC: control_driver[4015]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD SATA @location 0/RP1 FPD upgraded and activated!

0/RP1/ADMIN0:Aug  2 05:36:29.987 UTC: shelf_mgr[4032]:
%INFRA-SHELF_MGR-6-CARD_SW_OPERATIONAL : Card: 0/RP1 software state going to Operational

0/RP1/ADMIN0:Aug  2 05:36:29.987 UTC: shelf_mgr[4032]:
%INFRA-SHELF_MGR-6-CARD_HW_OPERATIONAL : Card: 0/RP1 hardware state going to Operational

0/RP1/ADMIN0:Aug  2 05:36:30.059 UTC: control_driver[4015]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : Auto fpd triggered for FPD PRIMARY-BIOS upgrade @
 location 0/RP1 please wait until all fpd upgrades are done
0/RP0/ADMIN0:Aug  2 05:36:31.704 UTC: control_driver[3920]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : Auto fpd triggered for FPD PRIMARY-BIOS upgrade @
 location 0/RP0 please wait until all fpd upgrades are done
0/RP1/ADMIN0:Aug  2 05:36:35.977 UTC: control_driver[4015]:
```

```
%INFRA-FPD_Driver-6-UPGRADE_RESULT : Upgrade completes 50 percent for fpd
PRIMARY-BIOS@location 0/RP1.
0/RP0/ADMIN0:Aug  2 05:36:36.706 UTC: control_driver[3920]:
%INFRA-FPD_Driver-6-UPGRADE_RESULT : Upgrade completes 50 percent for fpd
PRIMARY-BIOS@location 0/RP0.
0/RP1/ADMIN0:Aug  2 05:36:55.985 UTC: control_driver[4015]:
%INFRA-FPD_Driver-6-UPGRADE_RESULT : Upgrade completes 100 percent for fpd
PRIMARY-BIOS@location 0/RP1.
0/RP1/ADMIN0:Aug  2 05:36:56.012 UTC: control_driver[4015]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD PRIMARY-BIOS@0/RP1 image programming completed
 with UPGRADE DONE state Info: [  Upgrade Complete ]
0/RP1/ADMIN0:Aug  2 05:36:56.012 UTC: control_driver[4015]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD PRIMARY-BIOS @location 0/RP1 upgrade completed.

0/RP1/ADMIN0:Aug  2 05:36:56.018 UTC: shelf_mgr[4032]:
%INFRA-SHELF_MGR-6-CARD_SW_OPERATIONAL : Card: 0/RP1 software state going to Operational

0/RP1/ADMIN0:Aug  2 05:36:56.018 UTC: shelf_mgr[4032]:
%INFRA-SHELF_MGR-6-CARD_HW_OPERATIONAL : Card: 0/RP1 hardware state going to Operational

0/RP0/ADMIN0:Aug  2 05:36:56.582 UTC: control_driver[3920]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD PRIMARY-BIOS@0/RP0 image programming completed
 with UPGRADE DONE state Info: [  Upgrade Complete ]
0/RP0/ADMIN0:Aug  2 05:36:56.869 UTC: control_driver[3920]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD PRIMARY-BIOS @location 0/RP0 upgrade completed.

0/RP1/ADMIN0:Aug  2 05:36:56.871 UTC: shelf_mgr[4032]:
%INFRA-SHELF_MGR-6-CARD_SW_OPERATIONAL : Card: 0/RP0 software state going to Operational

0/RP1/ADMIN0:Aug  2 05:36:56.872 UTC: shelf_mgr[4032]:
%INFRA-SHELF_MGR-6-CARD_HW_OPERATIONAL : Card: 0/RP0 hardware state going to Operational

RP/0/RP0/CPU0:IOS#admin
Mon Aug  2 05:49:27.392 UTC

cafyauto connected from 192.0.0.4 using ssh on sysadmin-vm:0_RP1
sysadmin-vm:0_RP1# install commit
Mon Aug  2  05:49:40.694 UTC+00:00
result Mon Aug  2 05:49:41 2021 Install operation 4 (install commit) started by user
'cafyauto' will continue asynchronously.
sysadmin-vm:0_RP1# Mon Aug  2 05:49:44 2021 Install operation 4 completed successfully.
sysadmin-vm:0_RP1# 0/RP1/ADMIN0:Aug  2 05:49:44.733 UTC: inst_mgr[6351]:
%INFRA-INSTMGR-2-OPERATION_SUCCESS : Install operation 4 completed successfully
0/RP1/ADMIN0:Aug  2 05:49:46.936 UTC: inst_mgr[6351]: %PKT_INFRA-FM-6-FAULT_INFO :
INSTALL-IN-PROGRESS :CLEAR :0/RP1: Calvados INSTALL_IN_PROGRESS Alarm : being CLEARED
 for the system
sysadmin-vm:0_RP1#
```

5. Check the FPD Versions

```
sysadmin-vm:0_RP1# show hw-module fpd
Mon Aug  2  05:49:31.591 UTC+00:00
                                                                      FPD Versions
                                                                     ==============
Location  Card type         HWver FPD device       ATR Status      Run      Programd
-------------------------------------------------------------------------------------
0/RP0     N560-RSP4-E       0.0   ADM              CURRENT     1.06      1.06
0/RP0     N560-RSP4-E       0.0   IOFPGA           CURRENT     0.67      0.64
0/RP0     N560-RSP4-E       0.0   PRIMARY-BIOS     RLOAD REQ   0.16      0.19
0/RP0     N560-RSP4-E       0.0   SATA             CURRENT     2.10      2.10
0/RP1     N560-RSP4-E       0.0   ADM              CURRENT     1.06      1.06
0/RP1     N560-RSP4-E       0.0   IOFPGA           CURRENT     0.67      0.64
0/RP1     N560-RSP4-E       0.0   PRIMARY-BIOS     RLOAD REQ   0.16      0.19
0/RP1     N560-RSP4-E       0.0   SATA             CURRENT     2.10      2.10
```

```
0/FT0     N560-FAN-H        1.0   PSOC                  CURRENT     2.02      2.02
sysadmin-vm:0_RP1#
```

6. Disable FPD auto-upgrade in XR VM

```
RP/0/RP1/CPU0:PP-Router#conf
RP/0/RP1/CPU0:PP-Router(config)#fpd auto-upgrade disable
RP/0/RP1/CPU0:PP-Router(config)#commit
```

7. Perform ISSU Upgrade in XR VM

```
RP/0/RP0/CPU0:IOS#install extract ncs560-mini-x-7.3.2
Mon Aug  2 05:55:19.880 UTC
2021-08-02 05:55:22 Install operation 100 started by cafyauto:
  install extract ncs560-mini-x-7.3.2
2021-08-02 05:55:22 Package list:
2021-08-02 05:55:22     ncs560-mini-x-7.3.2
2021-08-02 05:55:23 Install operation will continue in the background
RP/0/RP0/CPU0:IOS#2021-08-02 05:59:42 Install operation 100 finished successfully
RP/0/RP0/CPU0:Aug  2 05:59:42.840 UTC: sdr_instmgr[1213]:
%INSTALL-INSTMGR-2-OPERATION_SUCCESS : Install operation 100 finished successfully
RP/0/RP0/CPU0:IOS#install activate issu  ncs560-xr-7.3.2
Mon Aug  2 06:02:06.097 UTC
2021-08-02 06:02:08 Install operation 101 started by cafyauto:
  install activate issu ncs560-xr-7.3.2
2021-08-02 06:02:08 Package list:
2021-08-02 06:02:08     ncs560-xr-7.3.2
This install operation will start the issu, continue?
 [yes/no]:[yes] yes
2021-08-02 06:04:03 Install operation will continue in the background
RP/0/RP0/CPU0:IOS#RP/0/RP0/CPU0:Aug  2 06:04:11.315 UTC: issudir[342]:
%PKT_INFRA-FM-6-FAULT_INFO : ISSU-IN-PROGRESS :DECLARE :0/RP0/CPU0: ISSU_IN_PROGRESS
Alarm : being DECLARED for the system
…
RP/0/RP0/CPU0:IOS# RP/0/RP1/CPU0:Aug  2 06:18:13.640 UTC: imfpga[345]: Change ISSU
Node State from [Secondary] to [Primary]
RP/0/RP1/CPU0:Aug  2 06:18:13.659 UTC: tmgctrl[444]: ISSU Node Role Change Notification
 [Primary]
RP/0/RP1/CPU0:Aug  2 06:18:13.725 UTC: issudir[139]:
%INSTALL-ISSU_INFRA-1-OPERATION_ISSU_RUN_DONE : ISSU Run has completed for install
operation '101'. Issue the command "install activate issu cleanup" to continue the
operation.
RP/0/RP1/CPU0:Aug  2 06:18:13.729 UTC: issudir[139]: %PKT_INFRA-FM-6-FAULT_INFO :
ISSU-IN-PROGRESS :DECLARE :0/RP1/CPU0:
RP/0/RP1/CPU0:Aug  2 06:19:13.695 UTC: fpd-serv[216]: %PKT_INFRA-FM-3-FAULT_MAJOR :
ALARM_MAJOR :FPD-NEED-UPGRADE :DECLARE :0/3:
RP/0/RP1/CPU0:Aug  2 06:19:13.695 UTC: fpd-serv[216]: %PKT_INFRA-FM-3-FAULT_MAJOR :
ALARM_MAJOR :FPD-NEED-UPGRADE :DECLARE :0/7:
RP/0/RP1/CPU0:Aug  2 06:19:53.982 UTC: fpd-serv[216]: %PKT_INFRA-FM-3-FAULT_MAJOR :
ALARM_MAJOR :FPD-NEED-UPGRADE :DECLARE :0/1:
RP/0/RP1/CPU0:Aug  2 06:19:13.764 UTC: imfpga[343]: ISSU Phase Change Notification
[Phase Cleanup]
RP/0/RP1/CPU0:Aug  2 06:19:19.442 UTC: tmgctrl[444]: ISSU Phase Change Notification
[Phase Completed], Becoming Timing Primary now
RP/0/RP1/CPU0:Aug  2 06:19:19.452 UTC: sdr_instmgr[1213]: %PKT_INFRA-FM-6-FAULT_INFO
: INSTALL-IN-PROGRESS :DECLARE :0/RP1/CPU0: INSTALL_IN_PROGRESS Alarm : being DECLARED
 for the system
RP/0/RP1/CPU0:Aug  2 06:19:19.519 UTC: issudir[139]: %PKT_INFRA-FM-6-FAULT_INFO :
ISSU-IN-PROGRESS :CLEAR :0/RP1/CPU0:
RP/0/RP1/CPU0:Aug  2 06:19:19.538 UTC: tmgctrl[444]: ISSU Phase Change Notification
[Phase Completed] Processing done
RP/0/RP1/CPU0:Aug  2 06:19:20.480 UTC: sdr_instmgr[1213]:
%INSTALL-INSTMGR-2-OPERATION_SUCCESS : Install operation 101 finished successfully

RP/0/RP1/CPU0:IOS# install commit
```

```
Mon Aug  2 06:44:07.572 UTC
2021-08-02 06:44:10 Install operation 103 started by root123:
  install commit
2021-08-02  06:44:11 Install operation will continue in the background
RP/0/RP1/CPU0:ISO#0/RP1/ADMIN0:Aug  2 06:44:20.839 UTC: inst_mgr[6353]:
%PKT_INFRA-FM-6-FAULT_INFO : INSTALL-IN-PROGRESS :CLEAR :0/RP1: Calvados
INSTALL_IN_PROGRESS Alarm : being CLEARED for the system
RP/0/RP1/CPU0:Aug  2 16:44:20.579 UTC: sdr_instmgr[1213]: %PKT_INFRA-FM-6-FAULT_INFO
: INSTALL-IN-PROGRESS :CLEAR :0/RP1/CPU0: INSTALL_IN_PROGRESS Alarm : being CLEARED
for the system
2021-08-02 06:44:23 Install operation 103 finished successfully
RP/0/RP1/CPU0:Aug  2 06:44:23.201 UTC: sdr_instmgr[1213]:
%INSTALL-INSTMGR-2-OPERATION_SUCCESS : Install operation 103 finished successfully
sysadmin-vm:0_RP1# show version
Mon Aug  2  07:24:58.414 UTC+00:00
Cisco IOS XR Admin Software, Version 7.3.2
Copyright (c) 2013-2021 by Cisco Systems, Inc.

Build Information:
 Built By     : ingunawa
 Built On     : Thu Jul 29 01:27:41 PDT 2021
 Build Host   : iox-ucs-003
 Workspace    : /auto/iox-ucs-003-san2/prod/7.3.2.DT_IMAGE/ncs560/ws
 Version      : 7.3.2
 Location     : /opt/cisco/calvados/packages/
 Label        : 7.3.2
System uptime is 1 hour, 53 minutes

sysadmin-vm:0_RP1# exit
Mon Aug  2  07:25:02.260 UTC+00:00
RP/0/RP1/CPU0:IOS#show version
Mon Aug  2 07:25:04.063 UTC
Cisco IOS XR Software, Version 7.3.2
Copyright (c) 2013-2021 by Cisco Systems, Inc.

Build Information:
 Built By     : ingunawa
 Built On     : Thu Jul 29 03:34:42 PDT 2021
 Built Host   : iox-ucs-003
 Workspace    : /auto/iox-ucs-003-san2/prod/7.3.2.DT_IMAGE/ncs560/ws
 Version      : 7.3.2
 Location     : /opt/cisco/XR/packages/
 Label        : 7.3.2

cisco NCS-560 () processor
System uptime is 1 hour 13 minutes

RP/0/RP1/CPU0:IOS #
```

8. Check the IMFPGA FPD Versions

Note IMFPGA FPDs are not upgraded as part of ISSU, as auto-upgrade is disabled. Based on your network design, you must enable and upgrade the IM FPDs as well as reload the IMs at a time when it has the least impact on traffic.

```
RP/0/RP1/CPU0:IOS#show hw-module fpd
Mon Aug  2 06:38:52.253 UTC

Auto-upgrade:Disabled

                                                                FPD Versions
                                                               ==============
```

```
Location   Card type            HWver FPD device      ATR Status    Running   Programd
-------------------------------------------------------------------------------------
0/1        NCS4200-1T16G-PS     0.0   IMFPGA          NEED UPGD     1.95      1.95
0/3        NCS4200-1T16G-PS     0.0   IMFPGA          NEED UPGD     1.95      1.95
0/4        A900-IMA8Z           0.0   IMFPGA          CURRENT      17.05     17.05
0/7        N560-IMA2C           0.0   IMFPGA          NEED UPGD     4.96      4.96
0/9        A900-IMA8Z           0.0   IMFPGA          CURRENT      17.05     17.05
0/RP0      N560-RSP4-E          0.0   ADM             CURRENT       1.06      1.06
0/RP0      N560-RSP4-E          0.0   IOFPGA          CURRENT       0.67      0.67
0/RP0      N560-RSP4-E          0.0   PRIMARY-BIOS    RLOAD REQ     0.16      0.19
0/RP0      N560-RSP4-E          0.0   SATA            CURRENT       2.10      2.10
0/RP1      N560-RSP4-E          0.0   ADM             CURRENT       1.06      1.06
0/RP1      N560-RSP4-E          0.0   IOFPGA          CURRENT       0.67      0.67
0/RP1      N560-RSP4-E          0.0   PRIMARY-BIOS    RLOAD REQ     0.16      0.19
0/RP1      N560-RSP4-E          0.0   SATA            CURRENT       2.10      2.10
0/FT0      N560-FAN-H           1.0   PSOC            CURRENT       2.02      2.02

RP/0/RP1/CPU0:IOS#
```

9. **Verify Standby is Node and NSR ready**

```
RP/0/RP1/CPU0:IOS#show redundancy
Mon Aug  2 06:39:44.953 UTC
Redundancy information for node 0/RP1/CPU0:
==========================================
Node 0/RP1/CPU0 is in ACTIVE role
Partner node (0/RP0/CPU0) is in STANDBY role
Standby node in 0/RP0/CPU0 is ready
Standby node in 0/RP0/CPU0 is NSR-ready

Reload and boot info
----------------------
RP reloaded Mon Aug  2 06:11:55 2021: 27 minutes ago
Active node booted Mon Aug  2 06:11:55 2021: 27 minutes ago
Standby node boot Mon Aug  2 06:20:02 2021: 19 minutes ago
Standby node last went not ready Mon Aug  2 06:22:46 2021: 16 minutes ago
Standby node last went ready Mon Aug  2 06:26:47 2021: 12 minutes ago
Standby node last went not NSR-ready Mon Aug  2 06:15:01 2021: 24 minutes ago
Standby node last went NSR-ready Mon Aug  2 06:24:52 2021: 14 minutes ago
There have been 0 switch-overs since reload

Active node reload "CARD_SHUTDOWN"
Standby node reload "CARD_SHUTDOWN "

RP/0/RP1/CPU0:IOS#
```

10. **IM FPD auto-upgrade: After ISSU**

    a. Configure the **fpd auto-upgrade enable** command to auto upgrade FPDs for all IMs

Note After ISSU is complete for RPs, configure FPD auto upgrade and FPD auto reload.

Note FPD auto reload affects network traffic. Ensure you schedule the reload of IMs at a time when it has the least impact on traffic.

Note Do not exit configure exclusive mode until all IM FPGA upgrades are automatically completed. Use a separate VTY session to monitor and check status.

```
RP/0/RP1/CPU0:IOS# configure exclusive
Mon Aug  2 06:45:11.297 UTC
RP/0/RP1/CPU0:IOS(config)# fpd auto-upgrade enable
RP/0/RP0/CPU0:IOS(config)# fpd auto-reload enable
RP/0/RP0/CPU0:IOS# commit
```

Note During this step, all IMs that need FPD upgrades, are upgraded simultaneously. IMs are reloaded automatically for the versions to reflect.

b. Exit **configure exclusive** mode when IMFPGA upgrades are completed on all IMs.

```
RP/0/RP1/CPU0:ROUTER#RP/0/RP1/CPU0:Jun 29 06:44:28.808 UTC: fpd_imfpga[430]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : Auto fpd triggered for FPD IMFPGA upgrade @
location 0/5/CPU0 please wait until all fpd upgrades are done
RP/0/RP1/CPU0:Jun 29 06:45:25.105 UTC: fpd-serv[125]: %PKT_INFRA-FM-3-FAULT_MAJOR
: ALARM_MAJOR :FPD-NEED-UPGRADE :DECLARE :0/5:
RP/0/RP1/CPU0:Jun 29 06:48:02.020 UTC: fpd_imfpga[430]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD IMFPGA@0/5 image programming completed with
 UPGRADE DONE state Info: [DONE ]
RP/0/RP1/CPU0:Jun 29 06:48:02.032 UTC: fpd_imfpga[430]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD IMFPGA @location 0/5 upgrade completed.
0/RP1/ADMIN0:Jun 29 06:48:02.473 UTC: shelf_mgr[3695]:
%INFRA-SHELF_MGR-6-CARD_HW_OPERATIONAL : Card: 0/5 hardware state going to
Operational
0/RP1/ADMIN0:Jun 29 06:48:02.473 UTC: shelf_mgr[3695]:
%INFRA-SHELF_MGR-6-CARD_SW_OPERATIONAL : Card: 0/RP1 software state going to
Operational
0/RP1/ADMIN0:Jun 29 06:48:02.474 UTC: shelf_mgr[3695]:
%INFRA-SHELF_MGR-6-CARD_HW_OPERATIONAL : Card: 0/RP1 hardware state going to
Operational
RP/0/RP1/CPU0:Jun 29 06:48:08.741 UTC: fpd-serv[125]:
%INFRA-FPD_SERVER-1-UPGRADE_ALERT : fpd auto-reload configured on this box.  Going
 to reload node 0/5/CPU0 to activate newly upgraded FPD(s).
0/RP1/ADMIN0:Jun 29 06:48:19.182 UTC: shelf_mgr[3695]: %INFRA-SHELF_MGR-4-CARD_RELOAD
 : Reloading card 0/5
0/RP1/ADMIN0:Jun 29 06:48:19.186 UTC: shelf_mgr[3695]: %INFRA-SHELF_MGR-6-HW_EVENT
 : Rcvd HW event HW_EVENT_RESET, event_reason_str '' for card 0/5
0/RP1/ADMIN0:Jun 29 06:48:21.581 UTC: shelf_mgr[3695]: %INFRA-SHELF_MGR-6-HW_EVENT
 : Rcvd HW event HW_EVENT_POWERED_OFF, event_reason_str 'reset requested by user'
for card 0/5
RP/0/RP1/CPU0:Jun 29 06:48:58.746 UTC: fpd-serv[125]: %PKT_INFRA-FM-3-FAULT_MAJOR
: ALARM_MAJOR :FPD-NEED-UPGRADE :CLEAR :0/5:
```

```
0/RP1/ADMIN0:Jun 29 06:49:52.266 UTC: shelf_mgr[3695]: %INFRA-SHELF_MGR-6-HW_EVENT
 : Rcvd HW event HW_EVENT_POWERED_OFF, event_reason_str '' for card 0/5
0/RP1/ADMIN0:Jun 29 06:49:57.586 UTC: shelf_mgr[3695]: %INFRA-SHELF_MGR-6-HW_EVENT
 : Rcvd HW event HW_EVENT_POWERED_ON, event_reason_str '' for card 0/5
0/RP1/ADMIN0:Jun 29 06:50:23.999 UTC: shelf_mgr[3695]: %INFRA-SHELF_MGR-6-HW_EVENT
 : Rcvd HW event HW_EVENT_OK, event_reason_str '' for card 0/5
0/RP1/ADMIN0:Jun 29 06:50:23.999 UTC: shelf_mgr[3695]:
%INFRA-SHELF_MGR-6-CARD_HW_OPERATIONAL : Card: 0/5 hardware state going to
Operational
```

11. Break Reload Actions in Multiple Steps to Avoid Traffic Loss

a. Reload Standby RP

In the following example, RP1 is the active and RP0 is the standby route processor. After the standby RP reloads and boots up, the RP FPDs come to node ready and NSR ready state.

Note It is expected behavior to see different versions of the RPs in System Admin VM and XR VM.

```
RP/0/RP1/CPU0:IOS# admin
Mon Aug  2 07:34:26.615 UTC
Last login: Mon Aug  2 07:20:24 2021 from 192.0.4.4

cafyauto connected from 192.0.4.4 using ssh on sysadmin-vm:0_RP1
sysadmin-vm:0_RP1# hw-module location 0/RP0 reload
Mon Aug  2  07:34:47.432 UTC+00:00
Reload hardware module ? [no,yes] yes
0/RP1/ADMIN0:Aug  2 07:34:48.728 UTC: shelf_mgr[4032]: %INFRA-SHELF_MGR-6-USER_ACTION
 : User cafyauto(192.0.4.4) requested CLI action 'graceful card reload' for location
 0/RP0
result Card graceful reload request on 0/RP0 succeeded.
0/RP1/ADMIN0:Aug  2 07:35:01.151 UTC: shelf_mgr[4032]: %INFRA-SHELF_MGR-4-CARD_RELOAD
 : Reloading card 0/RP0
RP/0/RP1/CPU0:Aug  2 07:35:01.263 UTC: fpd-serv[216]: %PKT_INFRA-FM-3-FAULT_MAJOR
: ALARM_MAJOR :FPD-NEED-UPGRADE :CLEAR :0/RP0:
RP/0/RP1/CPU0:IOS#show hw-module fpd

Auto-upgrade:Enabled
                                                                        FPD Versions


==============
Location   Card type             HWver FPD device       ATR Status    Running
Programd
-----------------------------------------------------------------------------------
0/1        NCS4200-1T16G-PS      0.0   IMFPGA               CURRENT     1.102
1.10
0/3        NCS4200-1T16G-PS      0.0   IMFPGA               RLOAD REQ   1.95
1.10
0/4        A900-IMA8Z            0.0   IMFPGA               CURRENT    17.05
17.05
0/7        N560-IMA2C            0.0   IMFPGA               RLOAD REQ   4.96
5.01
0/9        A900-IMA8Z            0.0   IMFPGA               CURRENT    17.05
17.05
0/RP0      N560-RSP4-E           0.0   ADM                  CURRENT     1.06
1.06
0/RP0      N560-RSP4-E           0.0   IOFPGA               CURRENT     0.67
0.67
0/RP0      N560-RSP4-E           0.0   PRIMARY-BIOS         CURRENT     0.19
```

```
0.19
0/RP0      N560-RSP4-E           0.0   SATA                CURRENT     2.10
2.10
0/RP1      N560-RSP4-E           0.0   ADM                 CURRENT     1.06
1.06
0/RP1      N560-RSP4-E           0.0   IOFPGA              CURRENT     0.67
0.67
0/RP1      N560-RSP4-E           0.0   PRIMARY-BIOS        RLOAD REQ   0.16
0.19
0/RP1      N560-RSP4-E           0.0   SATA                CURRENT     2.10
2.10
0/FT0      N560-FAN-H            1.0   PSOC                CURRENT     2.02
2.02
RP/0/RP1/CPU0:IOS#
```

b. Redundancy Switchover—Swap to the Updated RP as Active RP

After the standby RP is reloaded and booted up, perform the redundancy switchover to this RP with upgraded FPD, as active RP.

Note Follow Step 11a to upgrade and reload the new standby RP.

```
RP/0/RP1/CPU0:IOS#redundancy switchover
Mon Jun 28 17:16:44.914 UTC
Proceed with switchover 0/RP0/CPU0 -> 0/RP1/CPU0? [confirm]

RP/0/RP1/CPU0:Jun 28 17:16:46.157 UTC: rmf_svr[234]: %HA-REDCON-4-FAILOVER_REQUESTED
 : failover has been requested by operator, waiting to initiate
Initiating switch-over.
```

c.

Note You can either perform steps 11a and 11b, *or* you can perform step 11c.

Reload the Router to Reflect the Latest FPD Versions

Note Reloading the router impacts network traffic.

```
RP/0/RP1/CPU0:IOS#admin
Wed Sep  8 06:10:48.761 UTC
Last login: Wed Sep  8 06:05:30 2021 from 192.0.4.6

root connected from 192.0.4.6 using ssh on sysadmin-vm:0_RP1
sysadmin-vm:0_RP1# hw-module location all reload
Wed Sep  8  06:10:56.203 UTC+00:00
Reloading the module will be traffic impacting if not properly drained. Continue
to Reload hardware module ? [no,yes] yes
result Card graceful reload request on all acknowledged.
sysadmin-vm:0_RP1#
```

Automatic FPD Upgrade for PSU

During the Power Supply Unit (PSU) insertion and installation process, the routers can now automatically upgrade the Field-Programmable Devices (FPD) associated with the PSUs.

Starting with Cisco IOS-XR Release 7.5.2, the automatic FPD upgrade includes the FPDs associated with the PSUs by default. This means that when automatic FPD upgrade is enabled, the FPDs associated with the PSUs will also be upgraded. The upgrades for the PSUs will occur sequentially, so the FPD upgrades for the PSUs will take longer than for other components.

You can choose to exclude PSUs from the automatic upgrade process to reduce the time taken for FPD automatic upgrade by preventing them from being upgraded upon insertion or during a system upgrade using the **fpd auto-upgrade exclude pm** command.

Configuration example for excluding PSUs from automatic FPD upgrade:

Configuration

```
Router# config
Router(config)# fpd auto-upgrade enable
Router(config)# fpd auto-upgrade exclude pm
Router(config)# commit
```

Show Running Configuration

```
Router# show running-config fpd auto-upgrade
fpd auto-upgrade enable
fpd auto-upgrade include pm
```

Upgrade Failure

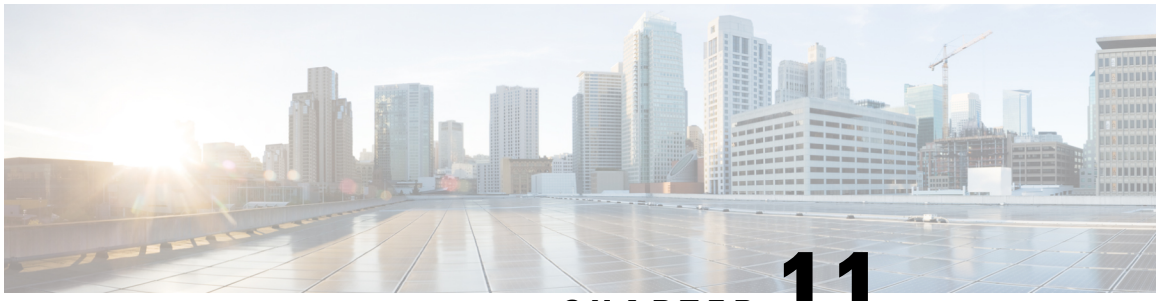
On failure of an FPD upgrade, you get a warning with the following syslog message:

```
LC/0/5/CPU0:Jun 27 05:02:25.742 UTC: optics_driver[216]: %INFRA-FPD_Driver-1-UPGRADE_ALERT
 : FPD MIFPGA@0/5 image programming completed with UPGD FAIL state Info: [Image verification
 failed at offset 0x5c8, flash value = 0x0, image value = 0x40, image size = 4194304]
LC/0/5/CPU0:Jun 27 05:02:26.570 UTC: optics_driver[216]: %INFRA-FPD_Driver-1-UPGRADE_ALERT
 : FPD MIFPGA@0/5 image programming completed with UPGD FAIL state Info: [Image verification
 failed at offset 0x1e, flash value = 0x56, image value = 0xff, image size = 4194304]
```

When you use the **show hw-module fpd** command, the status column displays **UPGD FAIL** to indicate failure of the FPD upgrade.

Note

- Do not reload the line card with a failed FPD upgrade image.
- Upgrade failed FPDs will be fixed with a manual upgrade.
- Contact Cisco TAC or your account representative if the FPD upgrade failure is not repaired.

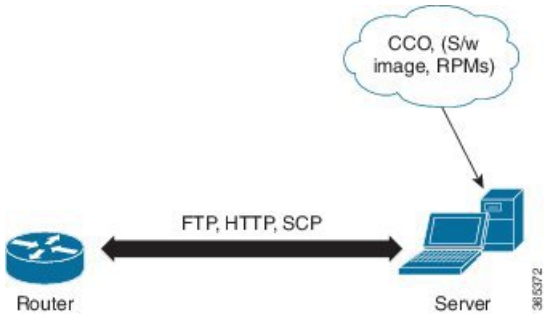


CHAPTER 11

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 9: 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 . 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).

The rest of this chapter contains these sections:

- Update RPMs and SMUs, on page 127
- Upgrade Base Software Version, on page 128
- Downgrade an RPM, on page 129

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]
  ```

- **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.)

- **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.

- **When an SMU is specified**

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

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**

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 is the RPM installed, then 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.

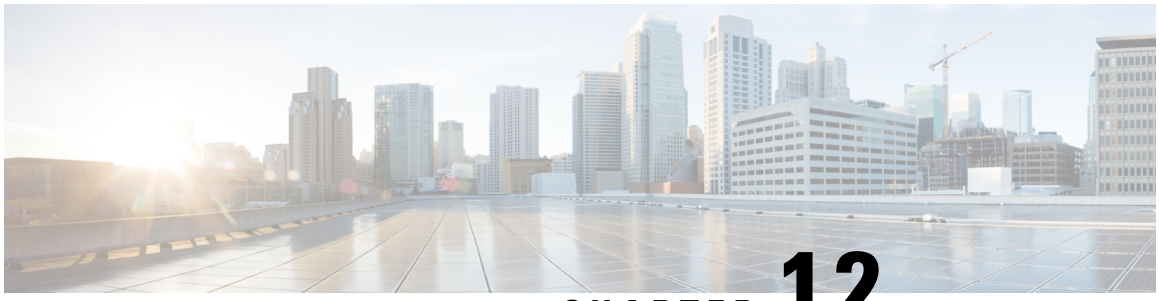
```
install deactivate <platform>-<rpm-name>
```

2. Check the active version of the RPM.

```
show install active
```

3. Commit the configuration.

```
install commit
```



CHAPTER 12

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 135.

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 132
- Customize Installation using Golden ISO, on page 132
- Golden ISO Workflow, on page 132
- Build Golden ISO Using Script, on page 133
- Install Golden ISO, on page 135
- Install Replace with Golden ISO, on page 137

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 135.

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

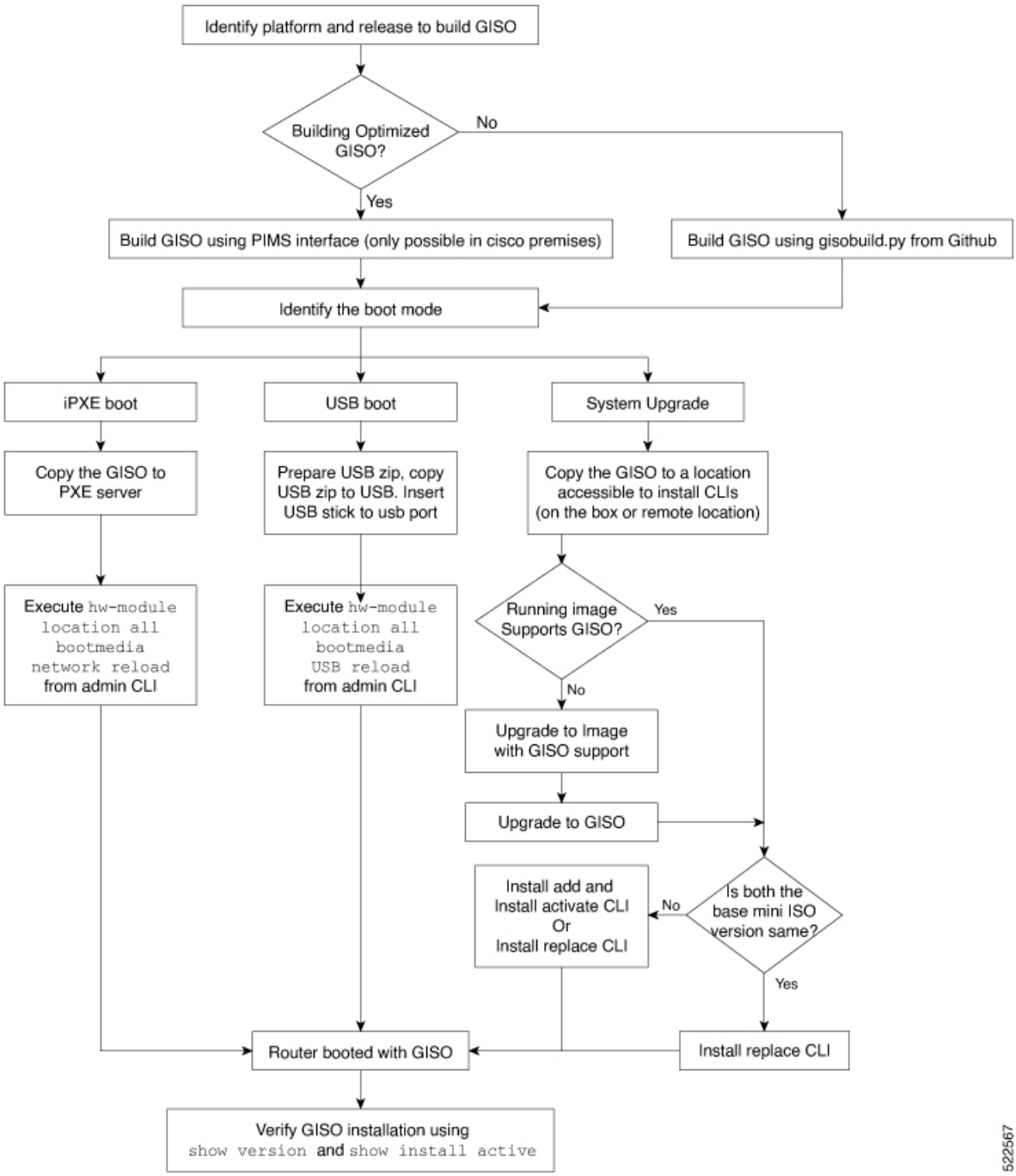
The following are the known problems and limitations with the customized ISO:

- 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 10: Golden ISO Workflow



Build Golden ISO Using Script

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

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]
```

To build GISO, perform the following steps:

Before you begin

- To upgrade from a release that did not support GISO to a release supporting GISO version, it is mandatory to first upgrade to mini ISO with GISO support.
- 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.
  - 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.

Procedure

Step 1 Copy the script `gisobuild.py` from the Github location 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.

Example:

```
[directory-path]$ gisobuild.py [-h] [-i <mini-x.iso>] [-r <rpm repository>]
[-c <config-file>] [-l <giso label>] [-m] [-v]
```

Note
The `-i` option is mandatory, and either or both `-r` or `-c` options must be provided.

The corresponding GISO and build logs are available under the specified `out_directory` path. The default directory is `/output_gisobuild`.

where:

- -i is the path to mini-x.iso
- -r is the path to RPM repository
- -c is the path to XR config file
- -l is the golden ISO label
- -h shows the help message
- -v is the version of the build tool `gisobuild.py`
- -m is to build the migration tar to migrate from IOS XR to IOS XR 64 bit

Note
It is recommended to build GISOs with a label name.

The corresponding GISO and build logs are available under the specified directory in `out_directory`. If a directory is not specified, the files are placed in `/output_gisobuild` directory.

Note The GISO script does not support verification of XR configuration.

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
  ```

- **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.

  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.

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.

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

Install Replace with Golden ISO

Procedure

Step 1 **install replace** *<GISO-location>* [**commit** | **noprompt**]

Example:

```
Router#install replace harddisk:/<giso-image>.iso
 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Install operation 11 started by root:
exec-timeout is suspended.
No install operation in progress at this moment
Label = More_Pkgs
ISO <giso-iso-image>.iso in input package list. Going to upgrade the system to

version <new-giso-image>.
System is in committed state
Current full-label: <giso-image>_R_Commit
Current only-label: R_Commit
Current label: R_Commit
Updating contents of golden ISO
Scheme : localdisk
Hostname : localhost
Username : None
SourceDir : /ws
Collecting software state..
Getting platform
Getting supported architecture
Getting active packages from XR
Getting inactive packages from XR
Getting list of RPMs in local repo
Getting list of provides of all active packages
Getting provides of each rpm in repo
```

```
Getting requires of each rpm in repo
Fetching .... <giso-image>.iso
Label within GISO: More_Pkgs
Skipping <platform>-mgbl-3.0.0.0-<release>.x86_64.rpm from GISO as it's active
Adding packages
        <platform>-golden-x-<release>-<Label>.iso
RP/0/RP0/CPU0:Jun 20 14:43:59.349 UTC: sdr_instmgr[1164]: %INSTALL-INSTMGR-2-OPERATION_SUCCESS
 :

Install operation 12 finished successfully
Install add operation successful
Activating <platform>-golden-x-<release>-<Label>
Jun 20 14:44:05 Install operation 13 started by root:
  install activate pkg <platform>-golden-x-<release>-<Label> replace noprompt
Jun 20 14:44:05 Package list:
Jun 20 14:44:05     <platform>-golden-x-<release>-<Label>.iso
Jun 20 14:44:29 Install operation will continue in the background
exec-timeout is resumed.
Router# Install operation 13 finished successfully
Router: sdr_instmgr[1164]: %INSTALL-INSTMGR-2-OPERATION_SUCCESS :

Install operation 13 finished successfully


Router#install replace <path-to-image> <platform-name-golden-x-<version>-<label>.iso
Tue Mar 17 08:07:15.176 UTC
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Mar 17 08:07:24 Install operation 46 started by root:
Mar 17 08:07:24   install replace source <path-to-image>
<platform-name-golden-x-<version>-<label>.iso
Mar 17 08:07:24 No install operation in progress at this moment
Mar 17 08:07:24 Checking system is ready for install operation
Mar 17 08:07:24 'install replace' in progress
Mar 17 08:07:24 Label = GISO_IMAGE_XRV9K_<version>
Mar 17 08:07:24 ISO xrv9k-goldenk9-x-<version>-<label>.iso in input package list. Going to
 upgrade the system to version <new-version>
Mar 17 08:07:25 Scheme : http
Mar 17 08:07:25 Hostname : 10.x.x.x
Mar 17 08:07:25 Collecting software state..
Mar 17 08:07:25 Getting platform
Mar 17 08:07:25 Getting supported architecture
Mar 17 08:07:25 Getting active packages from XR
Mar 17 08:07:25 Getting inactive packages from XR
Mar 17 08:07:28 Getting list of RPMs in local repo
Mar 17 08:07:28 Getting list of provides of all active packages
Mar 17 08:07:28 Getting provides of each rpm in repo
Mar 17 08:07:28 Getting requires of each rpm in repo
Mar 17 08:07:36 Fetching .... xrv9k-goldenk9-x-<version>-<label>.iso
Mar 17 08:08:02 Adding packages
        xrv9k-goldenk9-x-<version>-<label>.iso
Router:Mar 17 08:09:03.487 UTC: sdr_instmgr[1281]: %INSTALL-INSTMGR-2-OPERATION_SUCCESS :
Install operation 47 finished successfully
Mar 17 08:09:03 Install add operation successful
Mar 17 08:09:08 Activating xrv9k-goldenk9-x-<version>-<label>
Mar 17 08:09:10 Install operation 46 started by root:
  install activate pkg xrv9k-goldenk9-x-<version>-<label> replace
Mar 17 08:09:10 Package list:
Mar 17 08:09:10     xrv9k-goldenk9-x-<version>-<label>
This install operation will reload the system, continue?
 [yes/no]:[yes] yes
Mar 17 08:10:30 Install operation will continue in the background
Mar 17 08:10:30 Activate operation ID is: 46 for 'install source' ID:46

Router# Install operation 46 finished successfully
%INSTALL-INSTMGR-2-OPERATION_SUCCESS : Install operation 46 finished successfully
```

```
sdr_instmgr[1150]: %INSTALL-INSTMGR-2-SYSTEM_RELOAD_INFO : The whole system will be reloaded
 to complete install operation 46
```

For Cisco IOS XR Release 7.1.1, use the command `install replace harddisk:/<dir>/<giso-image>.iso`.

Important
For versions earlier than Cisco IOS XR Release 6.5.2, use the following command:

For example,

```
Router#install update source harddisk:/ <giso-image>.iso replace
```

The version and label of the newly added GISO is compared with the version and label of the currently active version. If a mismatch is identified, a new partition is created and the full package is installed. After installation, the system reloads with the image and packages from the newly added GISO.

Note
Activating or deactivating on a system that has a valid label invalidates the label. This action is irreversible. For example, running **show version** command on the system displays the label `6.3.3_633rev1005`. If any SMU is activated or deactivated on the system, the label `633rev1005` is invalidated, and the `show version` command displays only `6.3.3` as the label.

Step 2 **show version**

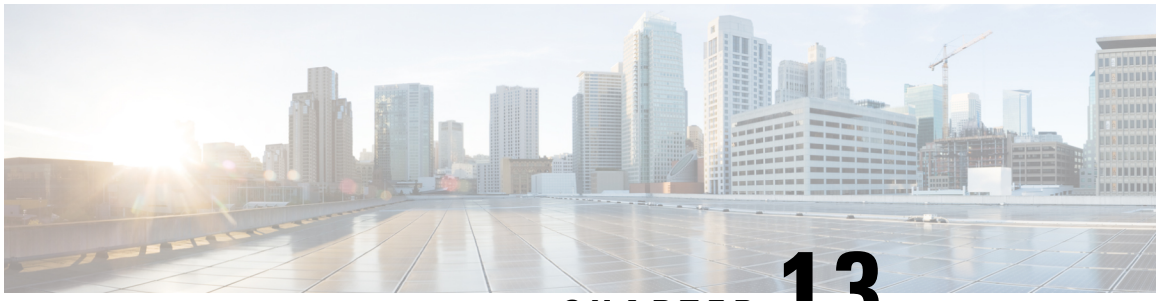
Example:

```
Router#show version
Wed Jun 20 15:06:37.915 UTC
Cisco IOS XR Software, Version <new-giso-image>
Copyright (c) 2013-2018 by Cisco Systems, Inc.

Build Information:
Built By     : <user>
Built On     : <date>
Build Host   : <host-name>
Workspace    : <workspace-name>
Version      : <version>
Location    : <path>
Label        : <label-name>

cisco <platform> () processor
System uptime is 3 hours 51 minutes
```

The system loads with the image and packages from the newly added GISO.



CHAPTER 13

Disaster Recovery

The topics covered in this chapter are:

- Boot using USB Drive, on page 141
- Boot the Router Using iPXE, on page 142

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 (max). USB 2.0 and USB 3.0 are supported.
- 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 *ncs560-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 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.

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.

Note Zero Touch Provisioning (ZTP) is not supported on the Cisco IOS XR Release 6.6.x routers.

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.

Ensure that the above configuration is successful.

- Use serial number of the router: 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 143.

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>/";
  } else
{
  # Auto-provision request, so provide ZTP script or configuration
  filename "http://<ip-address>/<script-directory-path>/";
  #filename "http://<ip-address>/<script-directory-path>/
  }
}
```

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 during installation, and then download and execute 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.

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/


http://10.37.1.235/ ... 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 Use the arrow keys (up, down) to select **UEFI: Built-in EFI IPXE** to enable iPXE boot. The iPXE boot launches the auto boot.

To manually boot using iPXE, press **Ctrl-B** keys to reach the iPXE command line.

Step 2 Identify the management interface. If the management interface is connected properly and is UP, it displays `Link:up` in the following output:

Example:

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.

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.