System Setup and Software Installation Guide for Cisco ASR 9000 Series Routers, IOS XR Release 7.11.x

First Published: 2023-11-30

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)

© 2023 Cisco Systems, Inc. All rights reserved.

CONTENTS

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

New and Changed System Setup Features 1

CHAPTER 2 **Cisco ASR 9000 System Features** 3

Cisco ASR 9000 Product Overview 3

Virtual Machine based Routing and System Administration 4

Command Modes 5

CHAPTER 3 **Bring-up the Router** 7

Boot the Router 7

Boot the router using USB 9

Boot the Router Using iPXE 11

Setup Root User Credentials 14

Access the System Admin Console 15

Configure the Management Port 16

Perform Clock Synchronization with NTP Server 18

CHAPTER 4 **Perform Preliminary Checks** 19

Verify Software Version 19

Verify Active VMs 20

Verify Status of Hardware Modules 21

Verify Firmware Version 22

Verify SDR Information 23

Verify Interface Status 25

CHAPTER 5 **Create User Profiles and Assign Privileges** 27

Create User Groups 28
Configure User Groups in XR VM 29
Create a User Group in System Admin VM 30
Create Users 31
Create a User Profile in XR VM 32
Create a User Profile in System Admin VM 35
Create Command Rules 36
Create Data Rules 39
Change Disaster-recovery Username and Password 41
Recover Password using PXE Boot 43

CHAPTER 6 **Perform System Upgrade and Install Feature Packages** 45
Upgrading the System 45
View supported software upgrade or downgrade versions 47
Compatibility checks for Cisco IOS XR software upgrades and downgrades 47
Show commands for software upgrade and downgrade 48
Supported software upgrade from running version 48
Supported releases to upgrade software from current version to target version 50
Supported releases from current version to an ISO version 50
Supported releases from running version to an ISO version 51
Upgrading Features 51
Pre-upgrade consideration for configuration commit history 52
Optimized Size of Install Image 53
Optional packages 55
Install Prepared Packages 55
Install Packages 57
Uninstall Packages 62
View Features and Capabilities Supported on a Platform 64

CHAPTER 7 **Manage Automatic Dependency** 69
Update RPMs and SMUs 70
Upgrade Base Software Version 70
Downgrade an RPM 71

CHAPTER 8 **Customize Installation using Golden ISO** 75

Limitations 76

Customize Installation using Golden ISO 77

Limitations 78

Golden ISO Workflow 78

Build Golden ISO 79

Build Golden ISO Using Script 80

Install Golden ISO 87

CHAPTER 9 **Deploy Router Using Classic ZTP** 93

Build your configuration file 94

Create User Script 95

ZTP Shell Utilities 96

ZTP Helper Python Library 97

Limitations of temporary ZTP files 102

Authentication on Data Ports 102

Set Up DHCP Server 104

Customize ZTP Initialization File 106

Zero Touch Provisioning on a Fresh Boot of a Router 108

Fresh Boot Using Removable Storage Device 108

Fresh Boot Using DHCP 110

Invoke ZTP Manually 112

CHAPTER 10 **Release-specific Caveats and Workarounds** 115

Caveats and Workarounds 115

CHAPTER 11 **Upgrading and Managing Cisco IOS XR Software** 117

Overview of Cisco IOS XR Software Packages 117

Package Installation Envelopes 118

Packages in the Cisco IOS XR Unicast Routing Core Bundle 118

PIE Filenames and Version Numbers 119

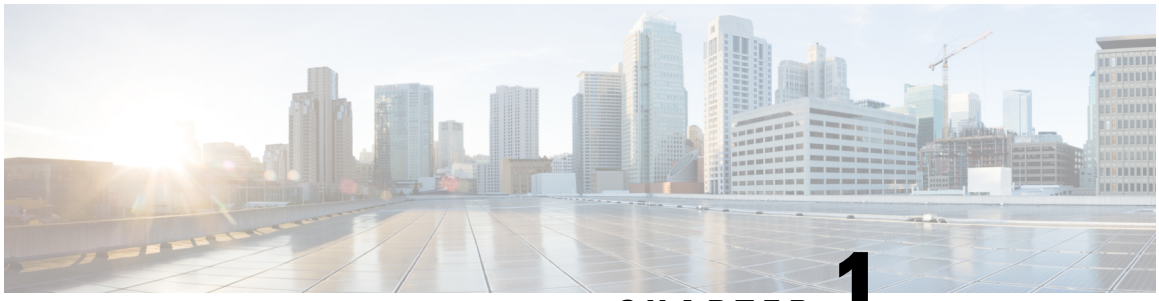
Summary of Package Management 120

Preparing for Software Installation Operations 123

Upgrading to Cisco IOS XR Software Release 4.0 134
Rolling Back to a Previous Software Set 141
Resetting Router to Factory Settings 144
Red Hat Package Manager 145
Cisco IOS-XR 64-bit package file names and versions 145
Rolling back to a previous installation operation 145
Filename components description 146
Copying the file to a local storage device or network server 147
Software Maintenance Upgrades 147
Deactivation of fully superseded SMUs 148
Upgrading Packages 148
Downgrading Packages 149
Impact of Package Version Changes 149
Impact of Package Activation and Deactivation 150
Delaying the return of the CLI prompt 150
Displaying Installation Log Information 151
Examples 151
Package Management Procedures 153
Activation and Deactivation Prerequisites 153
Obtaining and Placing Cisco IOS XR Software 154
Transferring installation files from a network file server to a local storage device 155
Adding and activating packages 157
Examples 163
Committing the Active Package Set 165
Examples 166
Deactivating and Removing Cisco IOS XR Software Packages 168
Examples 172
Additional References 174

CHAPTER 12 **Upgrading Field-Programmable Devices** 177
Upgrading Field-Programmable Device 178
Prerequisites for FPD Image Upgrades 178
Overview of FPD Image Upgrade Support 178
Parallel Power Module Upgrade 179

Manual Power Module Upgrade 180
Automatic Line Card Reload on FPD Upgrade 181
Implementation Considerations 181
Configuring Automatic Line Card Reload on FPD Upgrade 181
Hitless FPD Firmware Upgrade for Optical Transceiver Modules 182
FPD upgrade service 183
Determining Upgrade Requirement 183
Automatic FPD upgrade 183
Manual FPD Upgrade 184
Upgrade TimingIC-A and TimingIC-B FPDs 184
YANG Data Model for Field Programmable Device 185
How to Upgrade FPD Images 186
Configuration Examples for FPD Image Upgrade 189
show hw-module fpd Command Output: Example 189
show fpd package Command Output: Example 191
upgrade hw-module fpd Command Output: Example 193
show platform Command Output: Example 194
Troubleshooting Problems with FPD Image Upgrades 194
Power Failure or Removal of a SPA During an FPD Image Upgrade 194
Performing a SPA FPD Recovery Upgrade 195
Performing a SIP FPD Recovery Upgrade 195



CHAPTER 1

New and Changed Feature Information

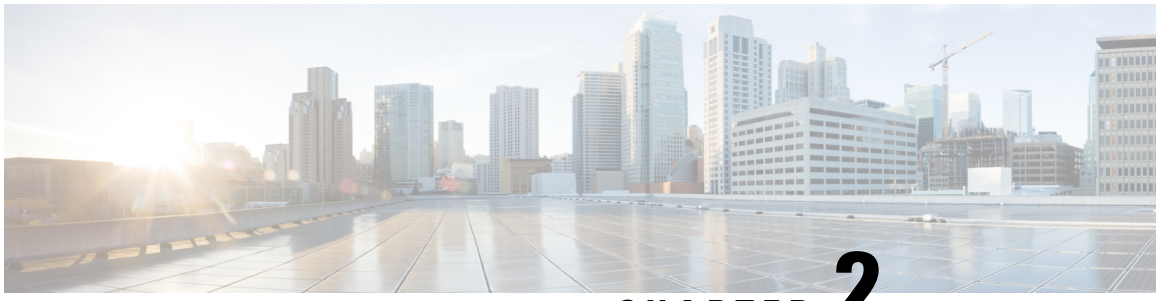
This table summarizes the new and changed feature information for the *System Setup and Software Installation Guide for Cisco ASR 9000 Series Routers* .

- New and Changed System Setup Features, on page 1

New and Changed System Setup Features

There are no new or changed software installation features in Release 7.11.1

Feature	Description	Changed in Release	Where Documented
Hitless FPD Firmware Upgrade for Optical Transceiver Modules	This feature is introduced.	Release 7.11.1	Hitless FPD Firmware Upgrade for Optical Transceiver Modules, on page 182



CHAPTER 2

Cisco ASR 9000 System Features

The topics covered in this chapter are:

- Cisco ASR 9000 Product Overview, on page 3
- Virtual Machine based Routing and System Administration, on page 4
- Command Modes, on page 5

Cisco ASR 9000 Product Overview

The Cisco ASR 9000 series routers are next-generation edge access routers that are optimized for service provider applications. These routers are designed to fulfill various roles in:

- Layer 2 and Layer 3 Ethernet aggregation
- Subscriber-aware broadband aggregation

The Cisco ASR 9000 series routers meet carrier-class requirements for redundancy, availability, packaging, power, and other requirements traditional to the service provider.

The Cisco ASR 9000 series consists of the following routers:

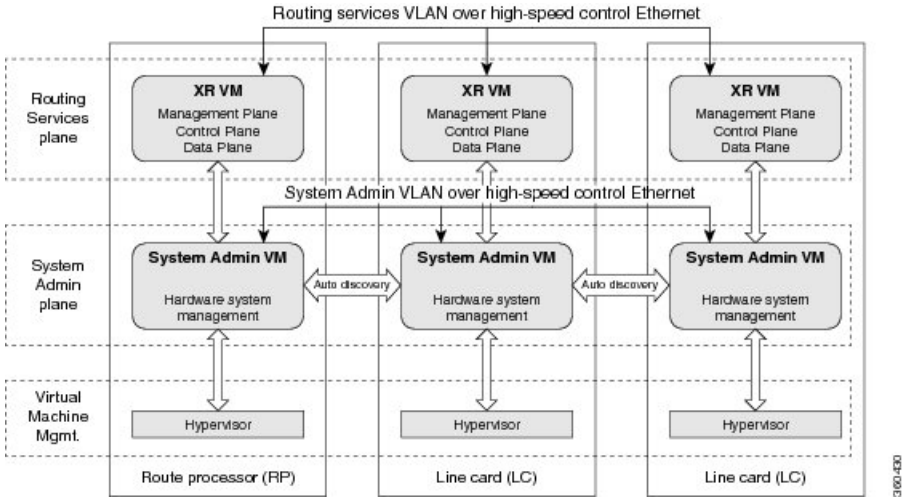
- Cisco ASR 9001 Router (32-bit)
- Cisco ASR 9001-S Router
- Cisco ASR 9006 Router
- Cisco ASR 9010 Router
- Cisco ASR 9901 Router
- Cisco ASR 9904 Router
- Cisco ASR 9906 Router
- Cisco ASR 9910 Router
- Cisco ASR 9912 Router
- Cisco ASR 9922 Router

Virtual Machine based Routing and System Administration

On the Cisco ASR 9000 series router running 64-bit IOS XR, the routing functions and the System Administration functions are run on separate virtual machines (VMs) over a Linux host operating system. The VMs simulate individual physical computing environments over a common hardware. Available hardware resources like processor, memory, hard disk, and so on, are virtualized and allocated to individual virtual machines by the hypervisor.

The VM topology on the Cisco ASR 9000 series router running 64-bit IOS XR is shown in this figure.

Figure 1: Virtualized IOS XR on Cisco ASR 9000 Series Router



Implementation of Virtualized IOS XR on Cisco ASR 9000 Series Router

- The hypervisor creates and manages individual VM environments.
- On every route processor (RP) there are two VMs; one for system administration (System Admin VM) and one for managing the routing functions (XR VM).
- The two VMs on each node operate on their respective planes. On each plane, the VMs are connected to each other using a dedicated VLAN over a high-speed Control Ethernet connection.
- The System Admin VMs can detect each other's presence by auto discovery and thus maintain complete system awareness.

To access the XR VM, connect to the XR VM console port on the RP. To access the System Admin VM, in the XR VM CLI, execute the **admin** command.

Note In 32-bit IOS XR OS, the management interfaces are available from XR VM. In 64-bit IOS XR OS, the Management ports on the RP/RSP are available as follows:

- MGT LAN 0 is available in XR VM.
- MGT LAN 1 is available in Admin VM.

Advantages of Virtualized IOS XR on the Router

- Faster boot time—Because the System Admin functions are on a dedicated VM, the boot time is considerably reduced.
- Independent upgrades—Software packages can be independently installed on the System Admin VM and the XR VM, resulting in minimal system downtime.
- Self-starting VMs—Both the System Admin VM and the XR VM are automatically launched during router boot-up without any user intervention. They have a default set-up that is ready for use.
- System redundancy—In spite of their interconnectivity, there is also a level of isolation between the VMs. Therefore, if a particular VM experiences any issues, it does not affect the functioning of other VMs.

Command Modes

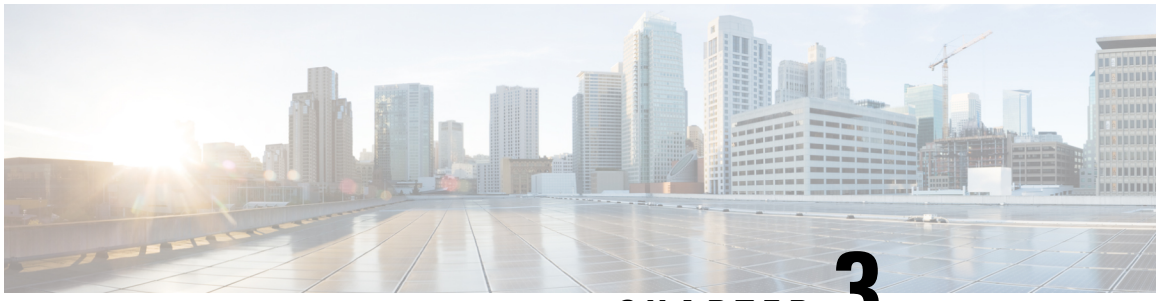
This table lists the command modes:

Command Mode	Description
XR VM Execution Mode	Run commands on the XR VM to display the operational state of the router. Example: `RP/0/RP0/CPU0:router#`

XR VM Global Configuration	Perform security, routing, and other XR feature configurations on the XR VM.
	Example:
	`RP/0/RP0/CPU0:router#configure`
	`RP/0/RP0/CPU0:router(config)#`

System Admin VM Execution Mode	Run commands on the System Admin VM 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 VM 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:router#admin`
	`sysadmin-vm:0_RP0#config`
	`sysadmin-vm:0_RP0(config)#`



CHAPTER 3

Bring-up the Router

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

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

For more information about completing the hardware installation, see Cisco ASR 9000 Series Aggregation Services Router Hardware Installation Guide.

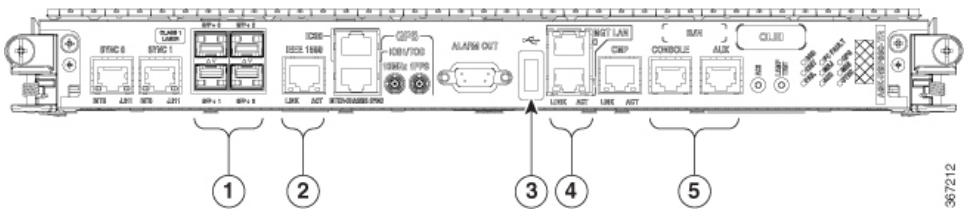
In a large-scale environment, to provision routers remotely without any manually intervention, we recommend you to use Zero Touch Provisioning (ZTP). See Deploy Router Using Classic ZTP, on page 93.

- Boot the Router, on page 7
- Boot the router using USB, on page 9
- Boot the Router Using iPXE, on page 11
- Setup Root User Credentials, on page 14
- Access the System Admin Console, on page 15
- Configure the Management Port, on page 16
- Perform Clock Synchronization with NTP Server, on page 18

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.

Figure 2: Route Processor Front Panel View



1	SFP/SFP+ ports

2	Service LAN port
3	External USB port
4	Management LAN ports
5	Console and Auxiliary (AUX) ports

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:

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

For chassis with RSP4 and RP2 cards, the console settings are baud rate 9600 bps, no parity, 1 stop bits and 8 data bits. The user can change this baud rate. For next generation RP3 and RSP5 cards, the console settings are baud rate 115200 bps, no parity, 1 stop bits and 8 data bits.

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 14.

Boot the router using USB

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

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

Before you begin

Ensure that:

- The compressed boot file, `asr9k-x64-usb_boot.zip`, is available.
- A USB drive is connected to the router and accessible at `/mnt/usb/vdg/`.

Procedure

Step 1 Copy the compressed boot file, `asr9k-x64-usb_boot.zip`, from its source location, say, `/misc/disk1`, to the mounted USB drive.

Example:

```
[router:/misc/disk1]$cp asr9k-x64-usb_boot.zip /mnt/usb/vdg/
```

Step 2 Verify whether the file is copied to the USB drive.

Example:

```
[router:/mnt/usb/vdg]$ls
'System Volume Information'   asr9k-x64-usb_boot.zip
```

Use `ls -ltr` to verify the file size. We also recommend you to compare the MD5 checksum value at the source and destination to ensure data integrity, though the command for MD5 checksum is not shown in the provided logs.

```
[router:/mnt/usb/vdg]$ls -ltr
total 1810392
drwxr-xr-x. 2 root root       8192 Jul 14 15:23 'System Volume Information'
-rwxr-xr-x. 1 root root 1853832854 Sep 19 06:11  asr9k-x64-usb_boot.zip
```

Step 3 Extract the contents of the `asr9k-x64-usb_boot.zip` file directly onto the USB drive. This process converts the USB drive into a bootable medium.

Example:

```
[router:/mnt/usb/vdg]$unzip asr9k-x64-usb_boot.zip
Archive:  asr9k-x64-usb_boot.zip
creating: boot/
inflating: boot/install-image.iso
creating: EFI/
creating: EFI/boot/
inflating: EFI/boot/grub.cfg
inflating: EFI/boot/bootx64.efi
```

Step 4 Verify whether the file contents are extracted correctly and the `boot/` and `EFI/` directories are created.

Example:

```
[router:/mnt/usb/vdg]$ls
EFI  'System Volume Information'   asr9k-x64-usb_boot.zip  boot
```

Step 5 Reload the router.

Example:

```
router# hw-module location all bootmedia network reload
```

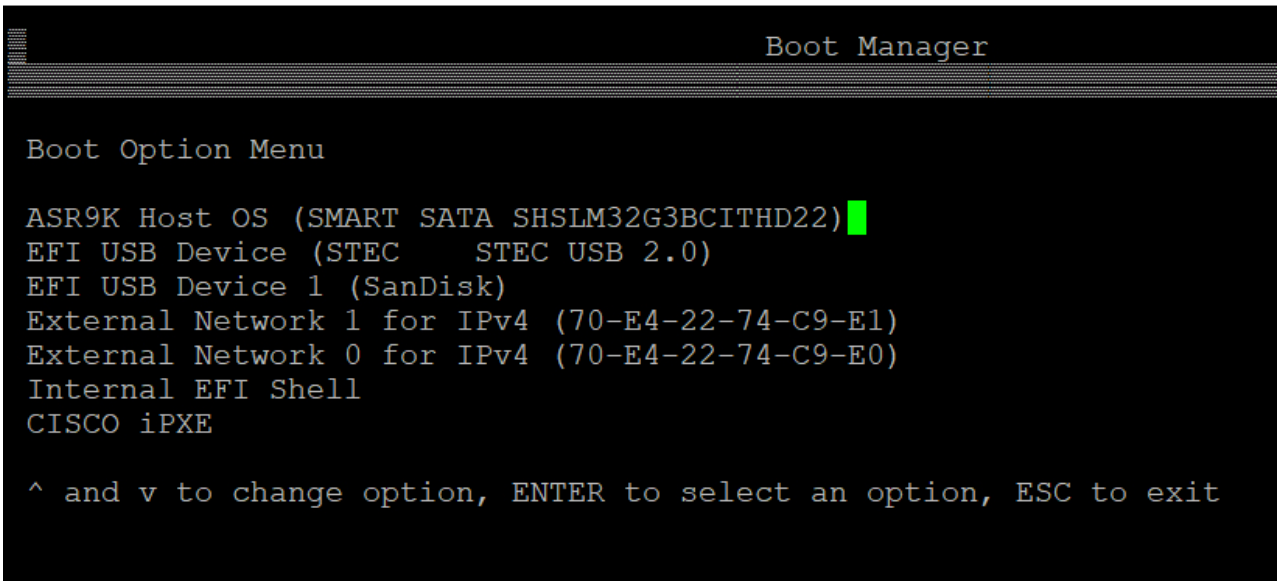
Step 6 While the router is reloading, press **Delete** or **Esc** to enter the **BIOS** menu.

From the **BIOS** menu, select the **Boot Manager** option using the up and down arrow keys and press **Enter**.

Figure 3: BIOS menu

Step 7 From the **Boot Manager** menu, select the desired USB boot option from the listed choices using the up and down arrow keys and press **Enter** to boot from the USB drive.

Figure 4: Boot Manager menu



Boot the Router Using iPXE

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

iPXE acts as a boot loader and provides the flexibility to choose the image that the system will boot based on the Platform Identifier (PID), the Serial Number, or the management mac-address. iPXE must be defined in the DHCP server configuration file.

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

Note **Cisco ASR 9901** — By default, iPXE uses the previous attempted boot method on the next reload. If the Network option was previously used, the iPXE register will be set to 1 (IPXE_PREF=1). To boot an Cisco ASR 9901 router via DHCP on the next reload, you must set the IPXE_PREF settings to 0 (IPXE_PREF=0).

From the system admin console, enter the **run chvrf 0 ssh rp0_admin /opt/cisco/calvados/bin/nvram_dump -s IPXE_PREF=0** command twice. After entering this command the first time, the host is added to the known list of hosts.

```
sysadmin-vm:0_RP0# run chvrf 0 ssh rp0_admin /opt/cisco/calvados/bin/nvram_dump -s IPXE_PREF=0
Sat May  2  10:39:52.740 UTC+00:00
Warning: Permanently added 'rp0_admin' (ECDSA) to the list of known hosts.
sysadmin-vm:0_RP0# run chvrf 0 ssh rp0_admin /opt/cisco/calvados/bin/nvram_dump -s IPXE_PREF=0
Sat May  2  10:39:54.995 UTC+00:00
sysadmin-vm:0_RP0# hw-module location all bootmedia network reload
```

iPXE boot can be performed during the following scenarios:

- migration from 32-bit to 64-bit using migration script
- recover password
- boot-up failure with 64-bit image

Before you begin

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

Procedure

Step 1 Login to the system admin console.

Example:

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

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

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

Example:

```
System Bootstrap, Version 10.57 [ASR9K x86 ROMMON],
Copyright (c) 1994-2018 by Cisco Systems, Inc.
Compiled on Mon 01/09/2017 17:15:01.98
BOARD_TYPE             : 0x100317
Rommon                 : 10.57 (Primary)
Board Revision         : 4
PCH EEPROM             : 3.4
IPU FPGA(PL)           : 0.40.0 (Backup)
```

```
IPU INIT(HW_FPD)       : 0.30.0
IPU FSBL(BOOT.BIN)     : 1.19.0
IPU LINUX(IMAGE.FPD)   : 1.21.0
OPTIMUS FPGA           : 0.12.0
OMEGA FPGA             : 0.13.0
ALPHA FPGA             : 0.14.0
CHA FPGA               : 0.5.1
CBC0                   : Part 1=34.38, Part 2=34.38, Act Part=2
Product Number         : A9K-RSP880-SE
Chassis                : ASR-9904-AC
Chassis Serial Number  : FOX1936GBDD
Slot Number            : 1
Pxe Mac Address LAN 0  : 70:e4:22:06:13:40
Pxe Mac Address LAN 1  : 70:e4:22:06:13:41
=========================================================
Got EMT Mode as 3
Got Boot Mode as 0
Booting IOS-XR (32 bit Classic XR) - Press Ctrl-c to stop
```

Step 3 Choose option 4 for iPXE boot.

Example:

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

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

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

Example:

```
iPXE>set cisco/cisco-server-url:string tftp://<path>/asr9k-mini-x64.iso
iPXE>set cisco/cisco-ipv4-address:string 1.3.24.202
iPXE>set cisco/cisco-netmask-address:str 255.255.0.0
iPXE>set cisco/cisco-gateway-address:str 1.3.0.1
```

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

Example:

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

where net0 and net1 represents management port0 and port1 respectively.

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

Example:

```
iPXE>
iPXE> ifopen net0:
iPXE> boot tftp://<path>/asr9k-mini-x64-<release-number>.iso
tftp://<path>/asr9k-mini-x64-<release-number>.iso...  0%
Booting iso-image@0x83c525000(1135456256), bzImage@0x83c55f000(4526671)

.......BIOS CODE SIGN ENTRY ...
```

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

Example:

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

Setup Root User Credentials

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

Procedure

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

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

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

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

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

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

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

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

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

Step 4 **Username:** *username*

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

Step 5 **Password:** *password*

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

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

Displays user details.

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

What to do next

- Configure routing functions from the XR console.
- Configure system administration settings from the System Admin prompt. The System Admin prompt is displayed on accessing the System Admin console. For details on how to get the System Admin prompt, see Access the System Admin Console, on page 15.

Access the System Admin Console

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

Procedure

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

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

a) **configure**

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

Example:

```
RP/0/RSP0/CPU0:router(config)#service sysadmin-login-banner disable
```

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

c) **commit**

d) **end**

Step 3 **admin**

Example:

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

```
RP/0/RSP0/CPU0:router#admin
```

```
Mon May 22 06:57:29.350 UTC

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

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

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

Step 4 (Optional) **exit**

Return to the XR mode from the System Admin mode.

Configure the Management Port

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

Before you begin

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

Step 1 **configure**

Example:

```
RP/0/RSP0/CPU0:router# configure
```

Enters global configuration mode.

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

Example:

```
RP/0/RSP0/CPU0:router(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/RSP0/CPU0:router(config-if)#ipv4 address 10.1.1.1/8
```

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

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

Example:

```
RP/0/RSP0/CPU0:router(config-if)#ipv4 address 1.70.31.160 255.255.0.0
```

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

Step 5 **no shutdown**

Example:

```
RP/0/RSP0/CPU0:router(config-if)#no shutdown
```

Places the interface in an "up" state.

Step 6 **exit**

Example:

```
RP/0/RSP0/CPU0:router(config-if)#exit
```

Exits the Management interface configuration mode.

Step 7 **router static address-family ipv4 unicast** *0.0.0.0/0 default-gateway*

Example:

```
RP/0/RSP0/CPU0:router(config)#router static address-family ipv4 unicast 0.0.0.0/0 12.25.0.1
```

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

Step 8 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/RSP0/CPU0:router# configure
```

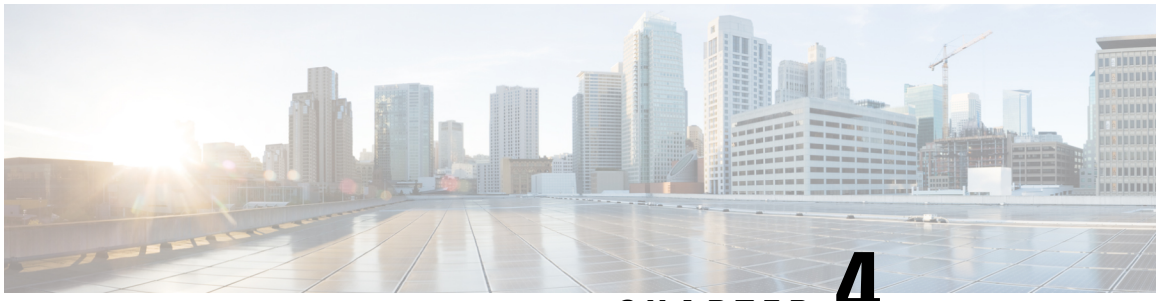
Enters global configuration mode.

Step 2 **ntp server** *server_address*

Example:

```
RP/0/RSP0/CPU0:router(config)#ntp server 64.90.182.55
```

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



CHAPTER 4

Perform Preliminary Checks

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

- Verify Software Version, on page 19
- Verify Active VMs, on page 20
- Verify Status of Hardware Modules, on page 21
- Verify Firmware Version, on page 22
- Verify SDR Information, on page 23
- Verify Interface Status, on page 25

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 Active VMs

On the router both the XR VM and the System Admin VM must be operational. Instances of both VMs should be running on every route processor (RP). Complete this task to verify the VMs are active.

SUMMARY STEPS

1. **show redundancy summary**
2. **admin**
3. **show vm**

DETAILED STEPS

Procedure

Step 1 **show redundancy summary**

Example:

```
RP/0/RP0:hostname#show redundancy summary
Mon Mar 9 16:32:19.276 IST
Active Node Standby Node
----------- ------------
0/RP0 0/RP1 (Node Ready, NSR: Not Configured)
0/LC0 0/LC1 (Node Ready, NSR: Not Configured)
RP/0/RP0:hostname#
```

Displays the readiness of the VMs.

Step 2 **admin**

Example:

```
RP/0/RSP0/CPU0:router# admin
```

Enters administration EXEC mode.

Step 3 **show vm**

Example:

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

Displays the status of the VMs running on various nodes.

```
sysadmin-vm:0_RP0# sh vm
Mon Mar 9 07:52:06.173 UTC
------ VMs found at location 0/RP0 ------
Id : sysadmin
Status : running
IP Addr: 192.0.44.1
HB Interval : NA
Last HB Sent: NA
Last HB Rec : NA
-------
Id : default-sdr
Status : running
```

```
IP Addr: 192.0.44.4
HB Interval : 0 s 500000000 ns
Last HB Sent: 663743
Last HB Rec : 663743
-------
Id : default-sdr
Status : running
IP Addr: 192.0.44.6
HB Interval : 10 s 0 ns
Last HB Sent: 33183
Last HB Rec : 33183
-------
------ VMs found at location 0/RP1 ------
Id : sysadmin
Status : running
IP Addr: 192.0.88.1
HB Interval : NA
Last HB Sent: NA
Last HB Rec : NA
-------
Id : default-sdr
Status : running
IP Addr: 192.0.88.4
HB Interval : 0 s 500000000 ns
Last HB Sent: 663749
Last HB Rec : 663749
-------
Id : default-sdr
Status : running
IP Addr: 192.0.88.6
HB Interval : 10 s 0 ns
Last HB Sent: 33183
Last HB Rec : 33183
-------
sysadmin-vm:0_RP0#
```

In the above result:

- Id—Name of the VM. "sysadmin" represents System Admin VM; "default-sdr" represents XR VM.
- Status—Status of the VM
- IP Addr—Internal IP address of the VM

If a VM is not running on a node, in the output of the **show vm** command, no output is shown for that node.

What to do next

If the XR VM is not running on a node, try reloading the node. To do so, use the **hw-module location** *node-id* **reload** command in the mode. Also, use the **show sdr** command in the mode to verify that the SDR is running on the node.

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.

Verify Firmware Version

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

SUMMARY STEPS

1. **show hw-module fpd**

DETAILED STEPS

Procedure

show hw-module fpd

Example:

Displays the list of hardware modules detected on the router.

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

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

- FPD Device- Name of the hardware component such as FPD, CFP, and so on.
- ATR-Attribute of the hardware component. Some of the attributes are:
  - B- Backup Image
  - S-Secure Image
  - P-Protected Image
- Status- Upgrade status of the firmware. The different states are:
  - CURRENT-The firmware version is the latest version.
  - READY-The firmware of the FPD is ready for an upgrade.
  - NOT READY-The firmware of the FPD is not ready for an upgrade.
  - NEED UPGD-A newer firmware version is available in the installed image. It is recommended that an upgrade be performed.
  - RLOAD REQ-The upgrade has been completed, and the ISO image requires a reload.
  - UPGD DONE-The firmware upgrade is successful.
  - UPGD FAIL- The firmware upgrade has failed.

- BACK IMG-The firmware is corrupted. Reinstall the firmware.
- UPGD SKIP-The upgrade has been skipped because the installed firmware version is higher than the one available in the image.

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

What to do next

- 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.
- It is recommended to upgrade all FPGAs on a given node using the **upgrade hw-module fpd all location** {**all** | *node-id*} command. Do not upgrade the FPGA on a node using the **upgrade hw-module fpd** <*individual-fpd*> **location** {**all** | *node-id*} as it may cause errors in booting the card.
- If required, turn on the auto fpd upgrade function. To do so, use the **fpd auto-upgrade enable** command in the XR configuration [**(config)#**] mode. After it is enabled, if there are new FPD binaries present in the image being installed on the router, FPDs are automatically upgraded during the system upgrade operation.

Verify SDR Information

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

Procedure

Step 1 **admin**

Example:

```
RP/0/RSP0/CPU0:router# admin
```

Enters administration EXEC mode.

Step 2 **show sdr**

Example:

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

Displays the SDR information for every node.

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

sdr default-sdr
 location 0/0/VM1
```

```
 sdr-id             2
 IP Address of VM   192.0.4.3
 MAC address of VM  A4:6C:2A:2B:AA:A6
 VM State           RUNNING
 start-time         2015-12-03T15:38:38.74514+00:00
 Last Reload Reason "SMU:Reboot triggered by install"
 Reboot Count       2
location 0/1/VM1
 sdr-id             2
 IP Address of VM   192.0.8.3
 MAC address of VM  B0:AA:77:E7:5E:DA
 VM State           RUNNING
 start-time         2015-12-03T15:38:39.730036+00:00
 Last Reload Reason "SMU:Reboot triggered by install"
 Reboot Count       2
location 0/2/VM1
 sdr-id             2
 IP Address of VM   192.0.12.3
 MAC address of VM  B0:AA:77:E7:67:34
 VM State           RUNNING
 start-time         2015-12-03T15:38:38.886947+00:00
 Last Reload Reason "SMU:Reboot triggered by install"
 Reboot Count       2
location 0/3/VM1
 sdr-id             2
 IP Address of VM   192.0.16.3
 MAC address of VM  B0:AA:77:E7:58:86
 VM State           RUNNING
 start-time         2015-12-03T15:38:40.391205+00:00
 Last Reload Reason "SMU:Reboot triggered by install"
 Reboot Count       2
location 0/4/VM1
 sdr-id             2
 IP Address of VM   192.0.20.3
 MAC address of VM  B0:AA:77:E7:46:C2
 VM State           RUNNING
 start-time         2015-12-03T15:38:39.84469+00:00
 Last Reload Reason "SMU:Reboot triggered by install"
 Reboot Count       2
location 0/5/VM1
 sdr-id             2
 IP Address of VM   192.0.24.3
 MAC address of VM  B0:AA:77:E7:84:40
 VM State           RUNNING
 start-time         2015-12-04T03:48:24.017443+00:00
 Last Reload Reason "VM_REQUESTED_UNGRACEFUL_RELOAD:Headless SDR"
 Reboot Count       3
location 0/6/VM1
 sdr-id             2
 IP Address of VM   192.0.28.3
 MAC address of VM  B0:AA:77:E7:55:FE
 VM State           RUNNING
 start-time         2015-12-03T15:38:38.74753+00:00
 Last Reload Reason "SMU:Reboot triggered by install"
 Reboot Count       2
location 0/7/VM1
 sdr-id             2
 IP Address of VM   192.0.32.3
 MAC address of VM  B0:AA:77:E7:60:C6
 VM State           RUNNING
 start-time         2015-12-03T15:38:38.691481+00:00
 Last Reload Reason "SMU:Reboot triggered by install"
 Reboot Count       2
location 0/RP0/VM1
```

```
  sdr-id             2
  IP Address of VM   192.0.108.4
  MAC address of VM  10:05:CA:D7:FE:6F
  VM State           RUNNING
  start-time         2015-12-04T07:03:04.549294+00:00
  Last Reload Reason CARD_SHUTDOWN
  Reboot Count       1
 location 0/RP1/VM1
  sdr-id             2
  IP Address of VM   192.0.112.4
  MAC address of VM  10:05:CA:D8:3F:43
  VM State           RUNNING
  start-time         2015-12-04T09:21:42.083046+00:00
  Last Reload Reason CARD_SHUTDOWN
  Reboot Count       1
```

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

What to do next

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

Verify Interface Status

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

SUMMARY STEPS

1. **show ipv4 interface summary**

DETAILED STEPS

Procedure

show ipv4 interface summary

Example:

```
RP/0/RSP0/CPU0:router#show ipv4 interface summary
```

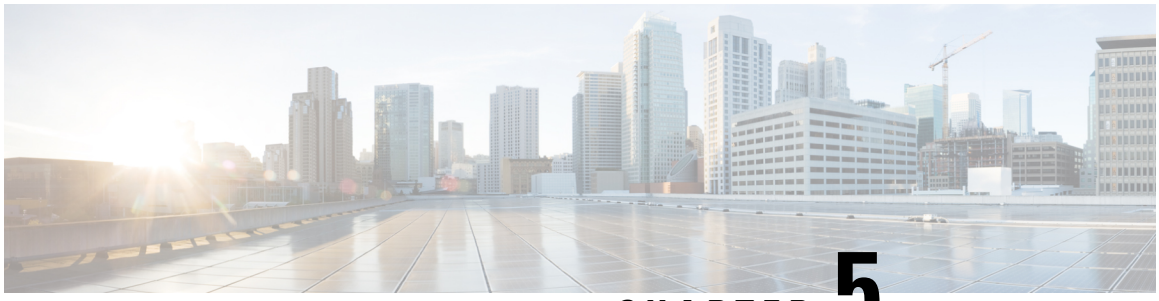
When a router is turned on for the first time, all interfaces are in the 'unassigned' state. Verify that the total number of interfaces displayed in the result matches with the actual number of interfaces present on the router.

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 mode to verify the interface status.



CHAPTER 5

Create User Profiles and Assign Privileges

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

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

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

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

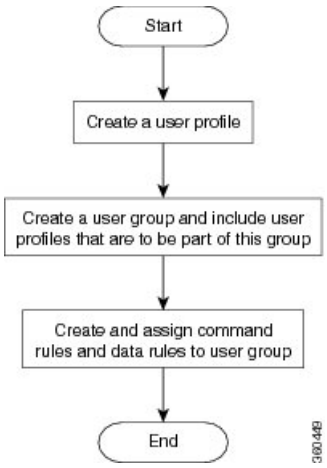
Note If any user on XR is deleted, the local database checks whether there is a first user on System Admin VM.

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

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

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

Figure 5: Workflow for Creating User Profiles



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

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

The topics covered in this chapter are:

- Create User Groups, on page 28
- Create Users , on page 31
- Create Command Rules, on page 36
- Create Data Rules, on page 39
- Change Disaster-recovery Username and Password, on page 41
- Recover Password using PXE Boot, on page 43

Create User Groups

Create a new user group to associate command rules and data rules with it. The command rules and data rules are enforced on all users that are part of the user group.

For extensive information about creating user groups, task groups, RADIUS and TACACS configurations, see the *Configuring AAA Services* chapter in the *System Security Configuration Guide for Cisco ASR 9000 Series Routers*. For detailed information about commands, syntax and their description, see the *Authentication, Authorization, and Accounting Commands* chapter in the *System Security Command Reference for Cisco ASR 9000 Series Routers*.

Configure User Groups in XR VM

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

Before you begin

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

Procedure

Step 1 **configure**

Example:

```
RP/0/RSP0/CPU0:router# configure
```

Enters global configuration mode.

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

Example:

```
RP/0/RSP0/CPU0:router(config)# usergroup beta
```

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

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

Step 3 **description** *string*

Example:

```
RP/0/RSP0/CPU0:router(config-ug)#
description this is a sample user group description
```

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

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

Example:

```
RP/0/RSP0/CPU0:router(config-ug)#
inherit usergroup sales
```

- Explicitly defines permissions for the user group.

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

Example:

```
RP/0/RSP0/CPU0:router(config-ug)# taskgroup beta
```

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

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

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

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

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

end —Prompts user to take one of these actions:

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

Create a User Group in System Admin VM

Create a user group for the System Admin VM.

The router supports a maximum of 32 user groups.

Before you begin

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

Procedure

Step 1 **admin**

Example:

```
RP/0/RSP0/CPU0:router# admin
```

Enters administration EXEC mode.

Step 2 **config**

Example:

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

Enters mode.

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

Example:

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

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

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

Step 4 **users** *user_name*

Example:

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

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

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

Step 5 **gid** *group_id_value*

Example:

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

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

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

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

end —Prompts user to take one of these actions:

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

What to do next

- Create command rules.
- Create data rules.

Create Users

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

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

XR VM and System Admin VM User Profile Synchronization

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

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

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

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

For extensive information about creating user groups, task groups, RADIUS and TACACS configurations, see the *Configuring AAA Services* chapter in the *System Security Configuration Guide for Cisco ASR 9000 Series Routers*. For detailed information about commands, syntax and their description, see the *Authentication, Authorization, and Accounting Commands* chapter in the *System Security Command Reference for Cisco ASR 9000 Series Routers*.

Create a User Profile in XR VM

Table 1: Feature History Table

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

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

For more information about AAA, and creating users, see the *Configuring AAA Services* chapter in the *System Security Configuration Guide for Cisco ASR 9000 Series Routers*. For detailed information about related commands, syntax and their description, see the *Authentication, Authorization, and Accounting Commands* chapter in the *System Security Command Reference for Cisco ASR 9000 Series Routers*.

Procedure

Step 1 **configure**

Example:

```
RP/0/RSP0/CPU0:router# configure
```

Enters global configuration mode.

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

Example:

```
RP/0/RSP0/CPU0:router(config)# username user1
```

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

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

Step 3 Do one of the following:

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

Example:

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

or

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

Specifies a password for the user named in Step 2.

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

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

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

    Note
    - Type 10 secret is supported only for Cisco IOS XR 64 bit platform.
    - Backward compatibility issues such as configuration loss, authentication failure, and so on, are expected when you downgrade to lower versions that still use **MD5** or **SHA256** encryption algorithms. If there are any type 10 secrets, convert the **secrets** to type 5 if you are downgrading the system from versions 7.0.1 and above to versions 6.5.3 and above. If you are downgrading the system from versions 7.0.1 and above to versions below 6.5.3, then un-configure all users from the XR-vm and sysadmin-vm before executing install activate. Backward compatibility issue does not occur in Cisco ASR 9000 Series Routers running Cisco IOS XR 32-Bit software because Type 10 secret is not applicable to such routers.
    - In a first user configuration scenario or when you reconfigure a user, the system synchronises only the Type 5 and Type 10 secrets from XR VM to System Admin VM and Host VM. It does not synchronize the Type 8 and Type 9 secrets in such scenarios.

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

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

Example:

```
RP/0/RSP0/CPU0:router(config-un)# group sysadmin
```

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

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

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

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

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

Enable or disable the login banner using these commands:

Example:

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

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

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

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

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

end —Prompts user to take one of these actions:

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

Create a User Profile in System Admin VM

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

The router supports a maximum of 1024 user profiles.

The root-lr user of XR VM can access the System Admin VM by entering **Admin** command in the 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/RSP0/CPU0:router# admin
```

Enters administration EXEC mode.

Step 2 **config**

Example:

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

Enters mode.

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

Example:

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

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

Step 4 **password** *password*

Example:

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

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

Step 5 **uid** *user_id_value*

Example:

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

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

Step 6 **gid** *group_id_value*

Example:

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

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

Step 7 **ssh_keydir** *ssh_keydir*

Example:

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

Specify any alphanumeric value.

Step 8 **homedir** *homedir*

Example:

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

Specify any alphanumeric value.

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

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

end —Prompts user to take one of these actions:

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

Create Command Rules

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

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

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

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

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

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

Before you begin

Create an user group. See Create a User Group in System Admin VM, on page 30.

SUMMARY STEPS

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

DETAILED STEPS

Procedure

Step 1 **admin**

Example:

```
RP/0/RSP0/CPU0:router# admin
```

Enters administration EXEC mode.

Step 2 **config**

Example:

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

Enters 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 39.

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 30.

SUMMARY STEPS

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

DETAILED STEPS

Procedure

Step 1 **admin**

Example:

```
RP/0/RSP0/CPU0:router# admin
```

Enters administration EXEC mode.

Step 2 **config**

Example:

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

Enters mode.

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

Example:

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

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

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

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

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

Step 4 **keypath** *keypath*

Example:

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

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

Step 5 **ops** *operation*

Example:

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

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

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

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

Example:

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

Specify whether users are permitted or denied the operation.

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

Step 7 **group** *user_group_name*

Example:

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

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

Step 8 **context** *connection type*

Example:

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

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

Step 9 **namespace** *namespace*

Example:

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

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

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

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

end —Prompts user to take one of these actions:

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

Change Disaster-recovery Username and Password

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

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

Step 1 **admin**

Example:

```
RP/0/RSP0/CPU0:router# admin
```

Enters administration EXEC mode.

Step 2 **config**

Example:

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

Enters mode.

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

Example:

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

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

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

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

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

end —Prompts user to take one of these actions:

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

Recover Password using PXE Boot

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

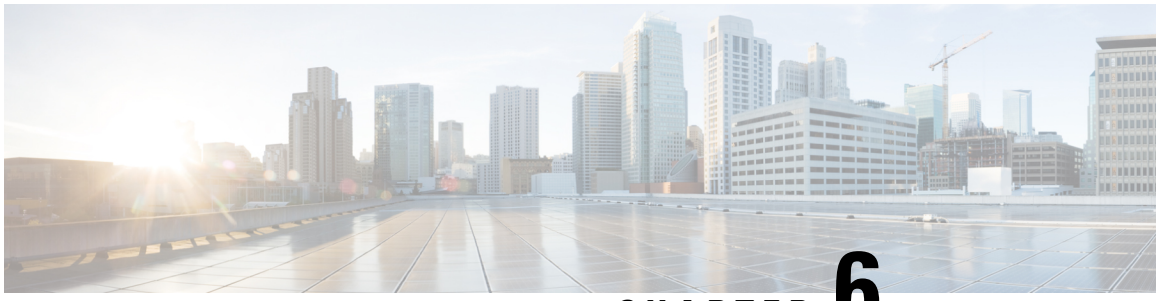
Procedure

Step 1 Boot the router using PXE.

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

To PXE boot a router, see Boot the Router Using iPXE, on page 11.

Step 2 Reset the password.



CHAPTER 6

Perform System Upgrade and Install Feature Packages

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

The topics covered in this chapter are:

- Upgrading the System, on page 45
- View supported software upgrade or downgrade versions, on page 47
- Upgrading Features, on page 51
- Pre-upgrade consideration for configuration commit history, on page 52
- Optimized Size of Install Image, on page 53
- Optional packages, on page 55
- Install Prepared Packages, on page 55
- Install Packages, on page 57
- Uninstall Packages, on page 62
- View Features and Capabilities Supported on a Platform, on page 64

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 *asr9k-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 CSM Server is a web-based, server-side automation and orchestration framework. It gives service providers the ability to simultaneously schedule and deploy SMUs and perform software upgrades across hundreds of routers in a scheduled manner through a simple click Web interface. For more information, see Cisco Software Manager.

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.

View supported software upgrade or downgrade versions

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

Table 2: Feature History Table

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

Compatibility checks for Cisco IOS XR software upgrades and downgrades

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

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

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

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

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

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

Show commands for software upgrade and downgrade

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

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

Note If your running version is below Cisco IOS XR Release 7.5.1, upgrade the system to Cisco IOS XR Release 7.5.1. Then, use the **show install upgrade-matrix** command to check the required SMUs for your target version.

Supported software upgrade from running version

This example shows all supported releases for upgrade from the current version on the ASR 9000 router:

```
Router# show install upgrade-matrix running
Fri Jul 29 10:18:43.413 IST
This may take a while ...
```

```
The current software [7.5.1] can be upgraded from and downgraded to the following releases:

================================================================================
From      To        Bridge SMUs Required        Caveats
================================================================================
7.0.2     7.5.1     r702.CSCvw57276             None
--------------------------------------------------------------------------------
6.5.3     7.5.1     r653.CSCvw57276             None
--------------------------------------------------------------------------------
7.5.1     7.0.2     None                        None
--------------------------------------------------------------------------------
7.5.1     6.5.3     None                        - https://www.cisco.com/c/en/us/
                                                td/docs/routers/asr9000/software
                                                /asr9k-r7-4/system-
                                                security/configuration/guide/b
                                                -system-security-cg-asr9000-74x
                                                /configuring-aaa-
                                                services.html#id_128191
--------------------------------------------------------------------------------
7.5.1     7.4.1     None                        None
--------------------------------------------------------------------------------
7.5.1     7.1.25    None                        None
--------------------------------------------------------------------------------
7.5.1     7.1.3     None                        None
--------------------------------------------------------------------------------
7.5.1     7.1.2     None                        None
--------------------------------------------------------------------------------
7.5.1     7.3.1     None                        None
--------------------------------------------------------------------------------
7.5.1     6.6.3     None                        - https://www.cisco.com/c/en/us/
                                                td/docs/routers/asr9000/software
                                                /asr9k-r7-4/system-
                                                security/configuration/guide/b
                                                -system-security-cg-asr9000-74x
                                                /configuring-aaa-
                                                services.html#id_128191
--------------------------------------------------------------------------------
7.5.1     7.3.2     None                        None
--------------------------------------------------------------------------------
7.4.1     7.5.1     None                        None
--------------------------------------------------------------------------------
7.1.25    7.5.1     None                        None
--------------------------------------------------------------------------------
7.1.3     7.5.1     None                        None
--------------------------------------------------------------------------------
7.1.2     7.5.1     None                        None
--------------------------------------------------------------------------------
7.3.1     7.5.1     None                        None
--------------------------------------------------------------------------------
6.6.3     7.5.1     r663.CSCvw57276             None
--------------------------------------------------------------------------------
7.3.2     7.5.1     None                        None
--------------------------------------------------------------------------------
```

Note Bridge SMUs must be activated on both the XR VM and the System Admin VM, if applicable.

Supported releases to upgrade software from current version to target version

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

```
Router#show install upgrade-matrix iso /harddisk:/asr9k-goldenk9-x64-7.5.2-rev1.iso

Fri Jul 29 10:19:24.185 IST
This may take a while ...
Upgrade from the current software [7.5.1] to 7.5.2 is supported
================================================================================
From      To        Bridge SMUs Required        Caveats
================================================================================
7.5.1     7.5.2     None                        None
--------------------------------------------------------------------------------
```

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

Supported releases from current version to an ISO version

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

```
Router#show install upgrade-matrix iso /harddisk:/ asr9k-goldenk9-x64-7.5.2-rev1.iso all
Fri Jul 29 10:20:10.961 IST
This may take a while ...

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

================================================================================
From      To        Bridge SMUs Required        Caveats
================================================================================
7.0.2     7.5.2     r702.CSCvw57276             None
--------------------------------------------------------------------------------
7.5.1     7.5.2     None                        None
--------------------------------------------------------------------------------
7.4.2     7.5.2     None                        None
--------------------------------------------------------------------------------
7.4.1     7.5.2     None                        None
--------------------------------------------------------------------------------
7.5.2     7.0.2     None                        None
--------------------------------------------------------------------------------
7.5.2     6.5.3     None                        - https://www.cisco.com/c/en/us/
                                                td/docs/routers/asr9000/software
                                                /asr9k-r7-4/system-
                                                security/configuration/guide/b
                                                -system-security-cg-asr9000-74x
                                                /configuring-aaa-
                                                services.html#id_128191
--------------------------------------------------------------------------------
7.5.2     7.1.25    None                        None
--------------------------------------------------------------------------------
7.5.2     7.4.2     None                        None
--------------------------------------------------------------------------------
7.5.2     7.6.1     None                        None
--------------------------------------------------------------------------------
7.5.2     7.4.1     None                        None
```

```
-------------------------------------------------------------------------------
7.5.2     7.1.3     None                        None
-------------------------------------------------------------------------------
7.5.2     7.1.2     None                        None
-------------------------------------------------------------------------------
7.5.2     7.3.1     None                        None
-------------------------------------------------------------------------------
7.5.2     6.6.3     None                        - https://www.cisco.com/c/en/us/
                                                td/docs/routers/asr9000/software
                                                /asr9k-r7-4/system-
                                                security/configuration/guide/b
                                                -system-security-cg-asr9000-74x
                                                /configuring-aaa-
                                                services.html#id_128191
-------------------------------------------------------------------------------
7.5.2     7.3.2     None                        None
-------------------------------------------------------------------------------
7.1.25    7.5.2     None                        None
-------------------------------------------------------------------------------
7.1.3     7.5.2     None                        None
-------------------------------------------------------------------------------
7.1.2     7.5.2     None                        None
-------------------------------------------------------------------------------
7.6.1     7.5.2     None                        None
-------------------------------------------------------------------------------
6.5.3     7.5.2     r653.CSCvw57276             None
-------------------------------------------------------------------------------
7.3.1     7.5.2     None                        None
-------------------------------------------------------------------------------
6.6.3     7.5.2     r663.CSCvw57276             None
-------------------------------------------------------------------------------
7.3.2     7.5.2     None                        None
-------------------------------------------------------------------------------
```

Supported releases from running version to an ISO version

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

```
Router#show install upgrade-matrix iso /harddisk:/asr9k-goldenk9-x64-7.5.2-rev1.iso
from-running
Fri Jul 29 10:21:31.957 IST
This may take a while ...
Upgrade from the current software [7.5.1] to 7.5.2 is supported
================================================================================
From      To        Bridge SMUs Required        Caveats
================================================================================
7.5.1     7.5.2     None                        None
--------------------------------------------------------------------------------
```

Upgrading Features

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

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

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

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

For example:

```
asr9k-9000v-nV-x64-1.0.0.0-r702.x86_64.rpm
```

Use the **install** commands to install packages and SMUs. For more information about the install process, see Install Packages, on page 57.

Note

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

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

For example, `asr9k-px-<release>.<bug ID>.pie` is an example of a package for the XR VM. `asr9k-sysadmin-<release>.pie` is associated with the system admin VM.

The XR packages or SMUs are activated from the XR VM, whereas the System Admin packages or SMUs are activated from the System Admin VM. If a bug ID is associated with both XR and System Admin SMUs, ensure both are installed.

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

Pre-upgrade consideration for configuration commit history

Before upgrading IOS XR software, keep in mind the following points about configuration commit history:

- The system stores configuration commit history in an internal binary database format, which uses a schema known as the configuration namespace version.
- This schema may change between software releases. If the namespace version changes, the upgraded software may not interpret commit records created on the previous release.
- Commands that reference pre-upgrade commits, such as **show configuration rollback changes** *<commit-id>*, may return the following error after upgrading:

```
Configuration namespace version of one or more of the processed commits is not supported
by the current software. The incompatible configuration will not be processed.
```

- This behavior is expected and does not indicate a system defect.

To preserve configuration commit history for audit or troubleshooting purposes, export commit details prior to the upgrade.

Example commands to export commit information:

- **show configuration commit list**
- **show configuration commit changes** <*commit-id*>

Save the output externally so it can be referenced after the upgrade.

After completing the upgrade, all commits created on the new software version will be fully accessible using standard rollback and commit history commands.

Optimized Size of Install Image

Table 3: Feature History Table

Feature Name	Release Information	Description
Optimized Size of Install Image	Release 7.3.1	With this release, the size of the Cisco ISO image installed on the router is reduced by approximately 300 MB. This frees-up the disk space on the router, and speeds-up time taken for installing the package.

Cisco IOS XR, Release 7.3.1 introduces an optimized version of mini-x ISO image with the overall image size reduced by approximately 300MB. This optimized ISO reduces the time for install operations, and the utilization of disk space for files stored in the install repository.

This optimized ISO supports booting the router using the following methods:

- iPXE boot
- USB boot
- System upgrade
- In-Service Software Upgrade (ISSU)

For ISSU, upgrading from a lower version to release 7.3.1 or later requires a bridge SMU in the running lower version. Whereas rolling back from release 7.3.1 or later to a lower version does not require a bridge SMU. There are no SMU dependencies when downgrading from release 7.3.1 with SMU installed to an older version. The bridge SMU is a mandatory prerequisite and must be installed on the running version before an upgrade is performed to release 7.3.1 or later.

The following are the bridge SMUs for host, XR and System admin that must be installed on the lower running version. For example, here, release 7.1.1:

- asr9k-iosxr-infra-64-3.0.0.2-r711.CSCvq46421.x86_64.rpm
- asr9k-iosxr-os-64-3.0.0.2-r711.CSCvq46421.x86_64.rpm
- asr9k-sysadmin-system-7.1.1-r711.CSCvq46421.x86_64.rpm

Bridge SMUs are included in the Cisco IOS XR software tar bundles located in Download Software Center for the specific release.

The commands related to install operations will display a new package with the term `common`. The common package is also listed in the output of install commands when executed using NETCONF Yang mode.

Note The `common` term is not listed for **show install active summary** and **show install committed summary** commands.

The common package is available in `/misc/disk1/tftpboot` of System admin VM.

The term `common` is also displayed in the output of **show install repository** or **show install repository all** commands in Sysadmin VM and XR VM.

The following example shows the output from Sysadmin VM:

```
sysadmin-vm:0_RSP0# show install repository
Sat May 2 16:33:25.200 UTC+00:00
Admin repository
---------------------
asr9k-common-7.3.1
asr9k-goldenk9-x64-7.3.1-supersetOptInstallSmu
asr9k-mini-x64-7.3.1
asr9k-sysadmin-7.3.1
asr9k-sysadmin-hostos-7.3.1-r731.admin.x86_64
asr9k-sysadmin-hostos-7.3.1-r731.host.x86_64
asr9k-sysadmin-hostos-7.3.1-r731.CSCho77777.admin.x86_64
asr9k-sysadmin-hostos-7.3.1-r731.CSCho77777.host.x86_64
asr9k-sysadmin-hostos-7.3.1-r731.admin.x86_64
asr9k-sysadmin-hostos-7.3.1-r731.host.x86_64
asr9k-sysadmin-hostos-7.3.1-r731.CSCho77777.admin.x86_64
asr9k-sysadmin-hostos-7.3.1-r731.CSCho77777.host.x86_64
asr9k-sysadmin-shared-7.3.1-r731.CSCcv33333.x86_64
asr9k-sysadmin-shared-7.3.1-r731.CSCcv33333.x86_64
asr9k-sysadmin-system-7.3.1-r731.CSCcv11111.x86_64
asr9k-sysadmin-system-7.3.1-r731.CSCcv22222.x86_64
---------------------- Truncated for Brevity -------------------------
```

The following example shows the output from XR VM:

```
Router#show install repository
Sat May 2 13:55:59.996 UTC
26 package(s) in XR repository:
asr9k-iosxr-infra-64-1.0.0.1-r731.CSCxr22222.x86_64
asr9k-iosxr-infra-64-1.0.0.2-r731.CSCxr44444.x86_64
asr9k-k9sec-x64-2.1.0.0-r731.x86_64
asr9k-optic-x64-1.0.0.0-r731.x86_64
asr9k-mgbl-x64-2.0.0.0-r731.x86_64
asr9k-mpls-te-rsvp-x64-2.1.0.0-r731.x86_64
asr9k-common-7.3.1
asr9k-goldenk9-x64-7.3.1-supersetOptInstallSmu
asr9k-bng-x64-1.0.0.0-r731.x86_64
```

```
asr9k-mini-x64-7.3.1
asr9k-mpls-x64-2.0.0.0-r731.x86_64
asr9k-li-x64-1.1.0.0-r73104I.x86_64
---------------------- Truncated for Brevity -------------------------
```

In the command output, `asr9k-common-7.3.1` represents the optimized package to reduce the image size.

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.

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 mode or 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 57.

Step 2 **show install repository**

Example:

```
RP/0/RSP0/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/RSP0/CPU0:router#show install request
```

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

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

Step 4 Execute one of these:

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

Example:

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/RSP0/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/RSP0/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/RSP0/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/RSP0/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 mode. The progress of FPD upgrade process can be monitored using the **show hw-module fpd** command in the mode. Reload the router after the FPD upgrade is completed.
- Verify the installation using the **install verify packages** command.
- Uninstall the packages or SMUs if their installation causes any issues on the router. See Uninstall Packages.

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

Install Packages

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

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

Note

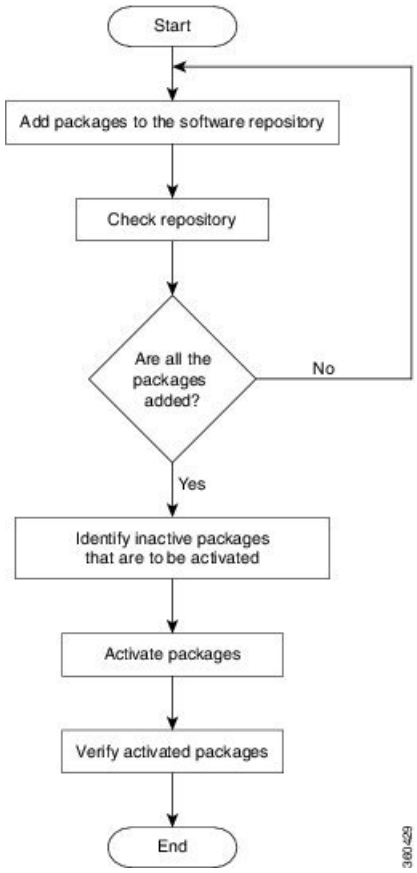
- Ensure that you have adequate disk space.
- Run the **fsck** command to check the status of the file system, for a successful IOS XR upgrade. You must run the **fsck** command in the System Admin EXEC mode to install a System Admin package, and in the XR EXEC mode to install the XR package.
- All install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

Note

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

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

Figure 6: Installing Packages Workflow



Before you begin

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

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

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

- Until Cisco IOS XR Release 7.11.21, after every user-triggered, ungraceful, or upgrade RP reload, you can check the sanity of the files from the install repository using the **run rpm -K --nosignature /install_repo/gl/xr/<package_name>** command in System Admin EXEC mode. This sanity check helps to detect the corrupt RPM files. Remove corrupt files, if detected and add valid files to avoid any file or file system corruption error that you may encounter during the installation or upgrade or post-upgrade process.

Procedure

Step 1 Execute one of these:

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

Example:

```
RP/0/RSP0/CPU0:router#install add source /harddisk:/ asr9k-mpls-x64-2.1.0.0-r731.x86_64.rpm
asr9k-mpls-x64-2.1.0.0-r732.x86_64.rpm
```

or

```
RP/0/RSP0/CPU0:router#install add source sftp://root@8.33.5.15:/auto/ncs/package/
asr9k-mcast-1.0.0.0-731.x86_64.rpm asr9k-iosxr-mpls-1.0.0.0-732.x86_64.rpm
```

or

```
RP/0/RSP0/CPU0:router#install add source /harddisk:/ asr9k-mpls-x64-2.1.0.0-<release-number>.x86_64.rpm
 asr9k-mpls-x64-2.1.0.0-<release-number>.x86_64.rpm
```

or

```
RP/0/RSP0/CPU0:router#install add source sftp://root@8.33.5.15:/auto/ncs/package/
asr9k-mcast-1.0.0.0-<release-number>.x86_64.rpm asr9k-iosxr-mpls-1.0.0.0-<release-number>.x86_64.rpm
```

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

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

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

Important
Starting from Cisco IOS XR Release 7.11.21, if a file or file system corruption error is reported during the installation or upgrade or post-upgrade process, an alarm is raised to indicate that the RPM files are corrupted and a syslog is displayed. The alarm is automatically cleared once you replace the corrupt files with valid files.

Step 2 **show install request**

Example:

```
RP/0/RSP0/CPU0:router#show install request
```

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

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

Step 3 **show install repository**

Example:

```
RP/0/RSP0/CPU0:router#show install repository
```

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

Step 4 **show install inactive**

Example:

```
RP/0/RSP0/CPU0:router#show install inactive
```

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

Step 5 Execute one of these:

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

Example:

The *operation_id* is that of the **install add** operation, see Install Packages, on page 57 Step Step 2, on page 60. This command can also be run from the Sys Admin mode.

The **install activate** operation makes the package configurations active on the router. Once activation begins, new features and software fixes take effect. By default, this operation runs in asynchronous mode, meaning the command executes in the background and the EXEC prompt is returned immediately. You can run the activate operation in synchronous mode by specifying the `sync` option from the CLI.

If you use an operation_id, all packages added as part of that operation are activated together. For example, if five packages were added under operation_id 8, running **install activate id 8** activates all five packages in a single step. You do not need to activate each package individually.

If an upgrade fails during activation, you can reuse the same operation_id to retry the activation after addressing the failure. Reusing the operation_id ensures that the same set of packages is activated consistently without re-adding them.

Activation does not occur instantaneously and may take some time to complete. Once activation is complete, the system reloads automatically. For restart SMU activation, the SMU takes effect when the processes impacted by the SMU are restarted.

Step 6 **show install active**

Example:

```
RP/0/RSP0/CPU0:router#show install active
```

Displays packages that are active.

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

Table 4: Example: Installing Packages: Related Commands

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

Step 7 **install commit**

Example:

```
RP/0/RSP0/CPU0:router#install commit
```

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

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

What to do next

- After performing a system upgrade, upgrade FPD by using the **upgrade hw-module location all fpd all** command from the mode. The progress of FPD upgrade process can be monitored using the **show hw-module fpd** command in the mode. Reload the router after the FPD upgrade is completed.
- Verify the installation using the **install verify packages** command.
- Uninstall the packages or SMUs if their installation causes any issues on the router. See Uninstall PackagesUninstall Packages, on page 62.

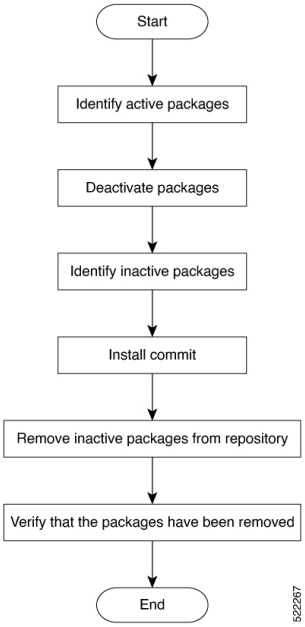
Uninstall Packages

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 7: Uninstalling Packages Workflow



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

Procedure

Step 1 **show install active**

Example:

```
RP/0/RSP0/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/RSP0/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/RSP0/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. .

View Features and Capabilities Supported on a Platform

Table 5: Feature History Table

Feature Name	Release Information	Description
View Features and Capabilities Supported on a Platform	Release 7.5.2	This functionality displays a list of supported and unsupported features and their capabilities in a release for your router. With this feature, you are better equipped to plan your network configuration with features annotated for their support information. This feature introduces the **show features** command.

This feature provides an answer to the question `Is feature X supported on my router?`

You can determine whether a feature and their capabilities are supported on your router for the release. The support information is based on the release and platform-specific data such as platform variants, RP, or LC present on the router.

Note In Cisco IOS XR Software Release 7.5.2, only the capabilities for Access Control List (ACL) feature is supported.

The functionality to determine the capabilities information is enabled by default when the supported release is installed on the router.

Use the **show features** command to view the list of supported features and their capabilities. The feature capabilities are displayed in a tree structure with notations for the support information. For example, in ACL, the capability to use compression to accommodate a large number of Access Control Elements (ACEs) is supported, whereas IPv6 ACL BNG does not have support data in Cisco IOS XR Software Release 7.5.2. This support information about the feature is represented with the following key in the tree structure:

Key	Capability Support Information	Description
X	Unsupported	The feature capability is not supported on the platform for the release
-	Supported	The feature capability is supported on the platform for the release
?	Support unknown	The support for the feature capability is unknown on the platform for the release. This data could be because the optional package for the feature is not installed on the router.
*	Support data not available	The support for the feature capability is not available on the platform for the release. This data could be because the feature may be specific to a line card that is not present on the router.

View the List of Supported Features

In this example, the supported features on the router are displayed.

Note In Cisco IOS XR Software Release 7.5.2, only the feature capabilities for Access Control List (ACL) are supported.

```
Router#show features
Fri June 3 19:16:58.298 UTC
Key:
X - Unsupported
- - Supported
? - Support unknown (optional package not installed)
* - Support data not available

[-] Cisco IOS XR
|--[-] XR Protocols
|  |--[-] XR Base Protocols
|  |  |--[-] Services
|  |  |  |--[-] Access Control List (ACL)
|  |  |  |  |--[-]  IPv6 ACL Support
|  |  |  |  |  |--[*] IPv6 ACL ABF Track
|  |  |  |  |  |--[*] IPv6 ACL BNG
|  |  |  |  |  |--[*] IPv6 ACL Chaining (Meta ACL)
|  |  |  |  |  |--[-] IPv6 ACL Common ACL
|  |  |  |  |  |--[-] IPv6 ACL Compression
|  |  |  |  |  |--[*] IPv6 ACL Default ABF
|  |  |  |  |  |--[*] IPv6 ACL Fragment
|  |  |  |  |  |--[-] IPv6 ACL ICMP Off
|  |  |  |  |  |--[-] IPv6 ACL ICMP Protocol
|  |  |  |  |  |--[-] IPv6 ACL Interface Statistics
|  |  |  |  |  |--[-] IPv6 ACL Log Rate
```

```
|   |   |   |   |   |--[-] IPv6 ACL Log Threshold
|   |   |   |   |   |--[-] IPv6 ACL Logging
|   |   |   |   |   |--[-] IPv6 ACL MIB
|   |   |   |   |   |--[-] IPv6 ACL Object Groups (Scale)
|   |   |   |   |   |--[-] IPv6 ACL Police
|   |   |   |   |   |--[-] IPv6 ACL Priority
|   |   |   |   |   |--[*] IPv6 ACL Protocol Range
|   |   |   |   |   |--[-] IPv6 ACL Set Qos-Group
|   |   |   |   |   |--[-] IPv6 ACL Set TTL
|   |   |   |   |   |--[-] IPv6 ACL TCP Flags
|   |   |   |   |   |--[-] IPv6 ACL TTL Match
|   |   |   |   |   |--[-] IPv6 ACL UDF
|   |   |   |   |--[-] ES-ACL Support (L2 ACL)
|   |   |   |   |--[-] IPv4 ACL Support
|   |   |   |   |   |--[-] IPv4 ACL  Set Qos-group
|   |   |   |   |   |--[*] IPv4 ACL ABF Track
|   |   |   |   |   |--[*] IPv4 ACL BNG
|   |   |   |   |   |--[*] IPv4 ACL Chaining (Meta ACL)
|   |   |   |   |   |--[-] IPv4 ACL Common ACL
|   |   |   |   |   |--[-] IPv4 ACL Compression
|   |   |   |   |   |--[*] IPv4 ACL Default ABF
|   |   |   |   |   |--[*] IPv4 ACL Fragment
|   |   |   |   |   |--[-] IPv4 ACL Fragment Flags
|   |   |   |   |   |--[-] IPv4 ACL ICMP Off
|   |   |   |   |   |--[-] IPv4 ACL ICMP Protocol
|   |   |   |   |   |--[-] IPv4 ACL Interface Statistics
|   |   |   |   |   |--[-] IPv4 ACL Log Rate
|   |   |   |   |   |--[-] IPv4 ACL Log Threshold
|   |   |   |   |   |--[-] IPv4 ACL Logging
|   |   |   |   |   |--[-] IPv4 ACL MIB
|   |   |   |   |   |--[-] IPv4 ACL Object Groups (Scale)
|   |   |   |   |   |--[-] IPv4 ACL Police
|   |   |   |   |   |--[-] IPv4 ACL Priority
|   |   |   |   |   |--[*] IPv4 ACL Protocol Range
|   |   |   |   |   |--[-] IPv4 ACL Set TTL
|   |   |   |   |   |--[-] IPv4 ACL TCP Flags
|   |   |   |   |   |--[-] IPv4 ACL TTL
|   |   |   |   |   |--[-] IPv4 ACL UDF
|   |   |   |   |--[-] IPv4 Prefix-List
|   |   |   |   |--[-] IPv6 Prefix-List
```

View the List of Supported ACL Features

In this example, the capabilities for ACL features on the router are displayed.

```
Router#show features acl
Fri June 3 19:17:31.635 UTC
Key:
X - Unsupported
- - Supported
? - Support unknown (optional package not installed)
* - Support data not available

[-] Access Control List (ACL)
|--[-]  IPv6 ACL Support
|   |--[*] IPv6 ACL ABF Track
|   |--[*] IPv6 ACL BNG
|   |--[*] IPv6 ACL Chaining (Meta ACL)
|   |--[-] IPv6 ACL Common ACL
|   |--[-] IPv6 ACL Compression
|   |--[*] IPv6 ACL Default ABF
|   |--[*] IPv6 ACL Fragment
```

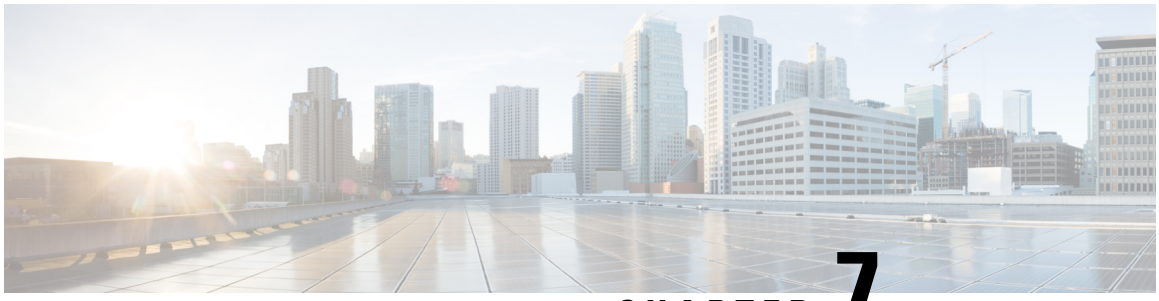
```
|   |--[-] IPv6 ACL ICMP Off
|   |--[-] IPv6 ACL ICMP Protocol
|   |--[-] IPv6 ACL Interface Statistics
|   |--[-] IPv6 ACL Log Rate
|   |--[-] IPv6 ACL Log Threshold
|   |--[-] IPv6 ACL Logging
|   |--[-] IPv6 ACL MIB
|   |--[-] IPv6 ACL Object Groups (Scale)
|   |--[-] IPv6 ACL Police
|   |--[-] IPv6 ACL Priority
|   |--[*] IPv6 ACL Protocol Range
|   |--[-] IPv6 ACL Set Qos-Group
|   |--[-] IPv6 ACL Set TTL
|   |--[-] IPv6 ACL TCP Flags
|   |--[-] IPv6 ACL TTL Match
|   |--[-] IPv6 ACL UDF
|--[-] ES-ACL Support (L2 ACL)
|--[-] IPv4 ACL Support
|   |--[-] IPv4 ACL  Set Qos-group
|   |--[*] IPv4 ACL ABF Track
|   |--[*] IPv4 ACL BNG
|   |--[*] IPv4 ACL Chaining (Meta ACL)
|   |--[-] IPv4 ACL Common ACL
|   |--[-] IPv4 ACL Compression
|   |--[*] IPv4 ACL Default ABF
|   |--[*] IPv4 ACL Fragment
|   |--[-] IPv4 ACL Fragment Flags
|   |--[-] IPv4 ACL ICMP Off
|   |--[-] IPv4 ACL ICMP Protocol
|   |--[-] IPv4 ACL Interface Statistics
|   |--[-] IPv4 ACL Log Rate
|   |--[-] IPv4 ACL Log Threshold
|   |--[-] IPv4 ACL Logging
|   |--[-] IPv4 ACL MIB
|   |--[-] IPv4 ACL Object Groups (Scale)
|   |--[-] IPv4 ACL Police
|   |--[-] IPv4 ACL Priority
|   |--[*] IPv4 ACL Protocol Range
|   |--[-] IPv4 ACL Set TTL
|   |--[-] IPv4 ACL TCP Flags
|   |--[-] IPv4 ACL TTL
|   |--[-] IPv4 ACL UDF
|--[-] IPv4 Prefix-List
|--[-] IPv6 Prefix-List
```

View the List of Supported ACL Features for Specific RP

In this example, the capabilities for ACL features on the RP location `0/RP0/CPU0` are displayed.

```
Router#show features acl detail location 0/RP0/CPU0
Fri June 3 19:15:49.889 UTC
Key:
X - Unsupported
- - Supported
? - Support unknown (optional package not installed)
* - Support data not available

[-] Access Control List (ACL)
    Cisco provides basic traffic filtering capabilities with access control
    lists (also referred to as access lists). User can configure access
    control lists (ACLs) for all routed network protocols to filter protocol
    packets when these packets pass through a device. User can configure
```

```
    access lists on your device to control access to a network, access lists
    can prevent certain traffic from entering or exiting a network.
|--[-]  IPv6 ACL Support
|      IPv6 based ACL is a list of source IPv6 addresses that use Layer 3 or
|      Layer 4 information to permit or deny access to traffic. IPv6 router
|      ACLs apply only to IPv6 packets that are routed.. A filter contains the
|      rules to match the packet matches, the rule also stipulates if the
|      packet should be permitted or denied.
|  |--[*] IPv6 ACL ABF Track
|  |      IPv6 ACL ABF Track allows the user to configure a rule with track as
|  |      nexthop inside the ACL rule . ACL Based Forwarding (ABF) denotes the
|  |      ability to forward packets to another next hop router based on the
|  |      criteria defined in the rule. Track takes precedence over VRF and
|  |      IP, if present in the nexthop
|  |--[*] IPv6 ACL BNG
|  |      IPv6 ACL BNG is an ACL subscriber BNG feature. It allows the use of
|  |      ACL on dynamic template.
|  |--[*] IPv6 ACL Chaining (Meta ACL)
|  |      IPv6 ACL Chaining (Meta ACL) allows the user to apply more than one
|  |      ACL on the interface.  is known as Meta ACL or ACL chaining.
|  |--[-] IPv6 ACL Common ACL
|  |      IPv6 ACL Common allows the user to apply the ACL on the interface
|  |      using the common keyword. Using this feature the ACL won't be
|  |      applied to the specific interface but it will be common to th entire
|  |      NPU to which the interface belongs.
|  |--[-] IPv6 ACL Compression
|  |      IPv6 ACL Compression allows the user to apply the ACL on the
|  |      interface using a compression level. This helps in reducing the
|  |      hardware resources needed to program the ACL.
|  |--[*] IPv6 ACL Default ABF
|  |      IPv6 ACL Default ABF allows the user to configure a rule with
|  |      default nexthop inside the ACL rule . ACL Based Forwarding (ABF)
|  |      denotes the ability to forward packets to another next hop router
|  |      based on the criteria defined in the rule
|  |--[*] IPv6 ACL Fragment
|  |      IPv6 ACL Fragment allows the user to configure a rule with fragment
|  |      inside the ACL rule and use it as a match criteria to filter traffic.
|  |--[-] IPv6 ACL ICMP Off
|  |      IPv6 ACL ICMP Off allows the user to not genearte the ICMP error
|  |      message on a deny action. When configured it will not send the
|  |      packet to FIB to generate ICMP error message.
-------------------------- Truncated for Brevity --------------------------
```

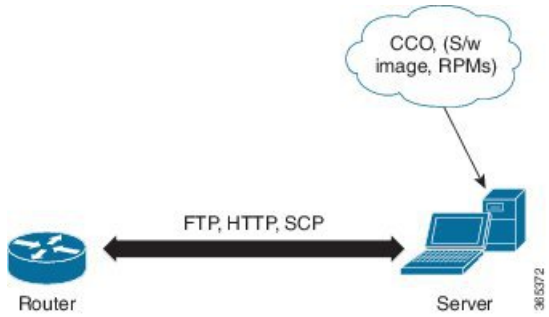


CHAPTER 7

Manage Automatic Dependency

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

Figure 8: Flow for Installation (base software, RPMs and SMUs)



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

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

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

Note

1. Cisco IOS XR Version 6.1.1 does not provide third party SMUs as part of automatic dependency management (**install source** command). The third party SMUs must be installed separately, and in isolation from other installation procedures (installation of SMUs and RPMs in IOS XR or admin containers).

The rest of this chapter contains these sections:

- Update RPMs and SMUs, on page 70
- Upgrade Base Software Version, on page 70
- Downgrade an RPM, on page 71

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.)

  ```
  install source [repository] asr9k-mpls.rpm
  ```

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

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

  ```
  install source [repository] asr9k-mpls-1.0.2.0-r710.x86_64.rpm
  ```

- **When an SMU is specified**

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

  ```
  install source [repository] asr9k-mpls-1.0.2.1-r611.CSCub12345.x86_64.rpm
  ```

Upgrade Base Software Version

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

Note SMUs are not upgraded as part of this process.

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

install source *repository*

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

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

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

- **The version number is specified**

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

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

  For example,

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

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

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

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

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

  install source *repository* **version 7.0.1 asr9k-mpls-1.0.2.0-r701.x86_64.rpm**

Downgrade an RPM

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

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

  For example:

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

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

  For example:

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

To deactivate the RPMs, perform the following steps:

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

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

    2. Activate the downloaded RPMs.

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

    3. Commit the configuration.

       ```
       install commit
       ```

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

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

       For example, if RPM 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.

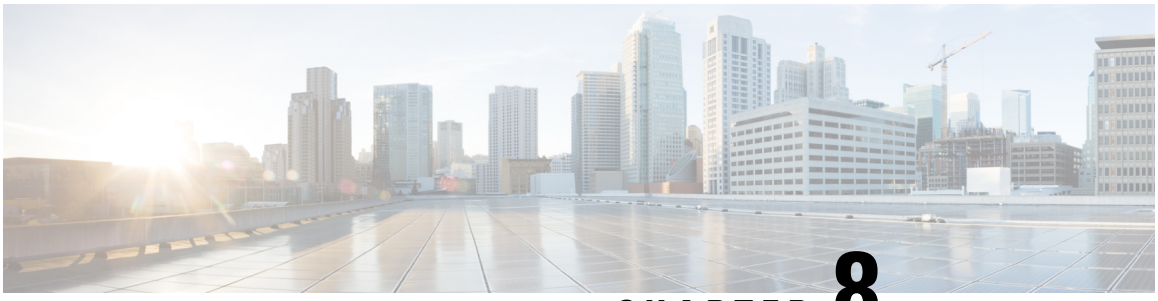
     ```
     install deactivate <platform>-<rpm-name>
     ```

  2. Check the active version of the RPM.

     ```
     show install active
     ```

  3. Commit the configuration.

```
install commit
```



CHAPTER 8

Customize Installation using Golden ISO

Table 6: Feature History Table

Feature Name	Release Information	Description
Automatic Install of Bridging Bug Fix RPMs	Release 7.5.1	This feature enables an easy one-step, no prompt upgrade, or downgrade, based on GISO. This removes the dependency on having to manually install the bridging bug fix RPMs before performing an upgrade or a downgrade.

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.

From Cisco IOS XR Release 7.5.1, you can use the Automatic Install of Bridging Bug Fix RPMs feature to install the bridging bug fix RPMs that are prerequisite for a system upgrade or a downgrade. You need to add the required Bridging Bug Fix RPMs into the customized ISO built using Cisco Golden ISO (GISO) build script **gisobuild.py**. The GISO can include bridging Bug Fix RPMs for multiple releases, and installs only the specific bridging Bug Fix RPMs required for the target release. The bridging bug fix RPMs can be used in the following scenarios:

- To resolve a bug that might stop upgrade.
- The latest version has new prerequisite requirements that are not met by the earlier version.

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 87.

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

- Migration from IOS XR 32-bit to IOS XR 64-bit

- 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 78
- Customize Installation using Golden ISO, on page 77
- Golden ISO Workflow, on page 78
- Build Golden ISO, on page 79
- Install Golden ISO, on page 87

Limitations

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

- GISO image size more than 1.8 GB is not supported. The maximum image size for RSP880-LT-SE/TR is 1.599 GB.
- 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.
- Migrating from IOS XR 32-bit to 64-bit OS using GISO involves the following restrictions:
  - The IOS XR 32-bit to 64-bit conversion script does not support file names exceeding 48 characters.
  - The IOS XR 32-bit OS has a maximum file size limit of 2 GB. Ensure that GISO does not exceed that limit.

    For more information about migration methods and system requirements, see the Migration Guide for Cisco ASR 9000 Series Routers.

Customize Installation using Golden ISO

Table 7: Feature History Table

Feature Name	Release Information	Description
Automatic Install of Bridging Bug Fix RPMs	Release 7.5.1	This feature enables an easy one-step, no prompt upgrade, or downgrade, based on GISO. This removes the dependency on having to manually install the bridging bug fix RPMs before performing an upgrade or a downgrade.

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.

From Cisco IOS XR Release 7.5.1, you can use the Automatic Install of Bridging Bug Fix RPMs feature to install the bridging bug fix RPMs that are prerequisite for a system upgrade or a downgrade. You need to add the required Bridging Bug Fix RPMs into the customized ISO built using Cisco Golden ISO (GISO) build script **gisobuild.py**. The GISO can include bridging Bug Fix RPMs for multiple releases, and installs only the specific bridging Bug Fix RPMs required for the target release. The bridging bug fix RPMs can be used in the following scenarios:

- To resolve a bug that might stop upgrade.
- The latest version has new prerequisite requirements that are not met by the earlier version.

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 87.

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

- Migration from IOS XR 32-bit to IOS XR 64-bit
- 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:

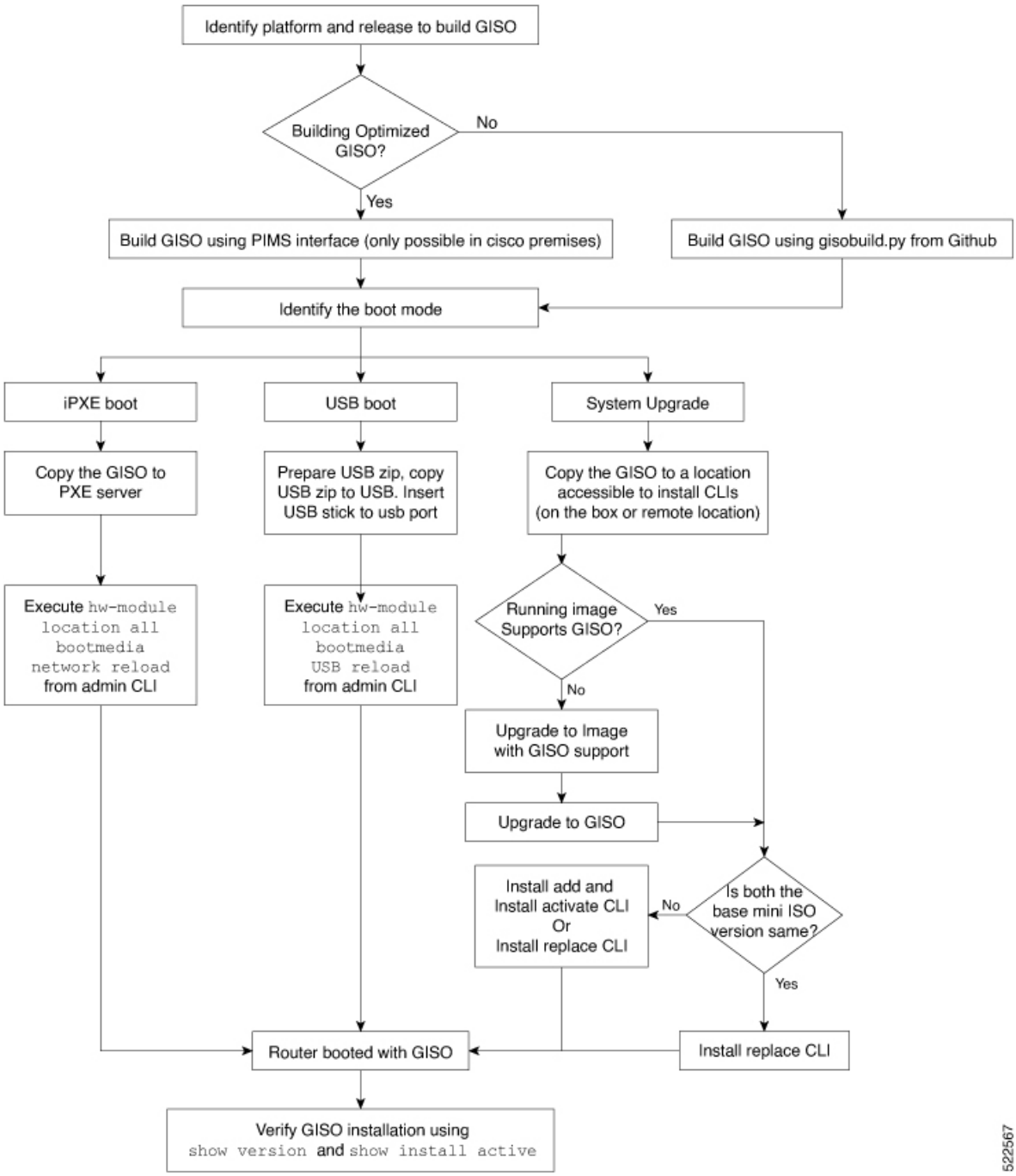
- GISO image size more than 1.8 GB is not supported. The maximum image size for RSP880-LT-SE/TR is 1.599 GB.
- Building and booting GISO for asynchronous package (a package of different release than the ISO) is not supported.
- Verifying the XR configuration is not supported in the GISO build script `gisobuild.py`.
- Renaming a GISO build and then installing from the renamed GISO build is not supported.
- Install operation over IPv6 is not supported.
- Migrating from IOS XR 32-bit to 64-bit OS using GISO involves the following restrictions:
  - The IOS XR 32-bit to 64-bit conversion script does not support file names exceeding 48 characters.
  - The IOS XR 32-bit OS has a maximum file size limit of 2 GB. Ensure that GISO does not exceed that limit.

    For more information about migration methods and system requirements, see the Migration Guide for Cisco ASR 9000 Series Routers.

Golden ISO Workflow

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

Figure 9: Golden ISO Workflow



Build Golden ISO

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

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

- Builds RPM database of all the packages present in package repository.
- Scans the repositories and selects the relevant Cisco RPMs that matches the input iso.
- Skips and removes third-party RPMs that are not SMUs of already existing third-party base package in mini-x.iso.
- Displays an error and exits build process if there are multiple base RPMs of same release but different versions.
- Performs compatibility check and dependency check for all the RPMs. For example, the child RPM asr9k-mpls-te-rsvp is dependent on the parent RPM asr9k-mpls. If only the child RPM is included, the Golden ISO build fails.

Build Golden ISO Using Script

Table 8: Feature History Table

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

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

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

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

The GISO script does not support verification of XR configuration.

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

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

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

Note

- Set the `migration` option to `true` when migrating from IOS XR 32-bit to IOS XR 64-bit software for Cisco ASR 9000 series routers.
- Set the `clean` option to `true` if you use the same build directory after the first GISO is created. Ensure that you set the option to `true` for every successive GISO build.
- Set the `docker` option to `true` if you are building GISO using docker.
- Ensure that the format and syntax of the YAML file is intact to avoid errors when building a GISO. For example, if the `:` symbol is missing, or if an unsupported symbol is used in the template, the GISO build displays errors.

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

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

To build GISO, perform the following steps:

Before you begin

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

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

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

Procedure

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

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

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

Utility to build Golden ISO for IOS-XR.

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

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

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

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

Example

Example: Build Docker-Based GISO Image

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

```
[root@xr src]# ./gisobuild.py --docker --iso /auto/asr9kgiso/asr9k-mini-x-7.5.1.iso
--repo /auto/asr9kgiso --pkglist asr9k-7.5.1.CSCsb88888 asr9k-bgp-2.0.0.0-r751.x86_64.rpm
asr9k-eigrp-1.0.0.0-r751.x86_64.rpm asr9k-isis-2.1.0.0-r751.x86_64.rpm
asr9k-k9sec-3.1.0.0-r751.x86_64.rpm
asr9k-li-1.0.0.0-r751.x86_64.rpm asr9k-mcast-3.0.0.0-r751.x86_64.rpm
asr9k-mgbl-3.0.0.0-r751.x86_64.rpm
asr9k-mpls-2.1.0.0-r751.x86_64.rpm asr9k-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
asr9k-ospf-2.0.0.0-r751.x86_64.rpm
asr9k-parser-2.0.0.0-r751.x86_64.rpm --label dockerbasedgiso

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

Platform: asr9000 Version: 7.5.1

Scanning repository [/auto/asr9000giso]...

Building RPM Database...

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

```
[ 1] asr9k-mpls-2.1.0.0-r751.x86_64.rpm
[ 2] asr9k-mgbl-3.0.0.0-r751.x86_64.rpm
[ 3] asr9k-bgp-2.0.0.0-r751.x86_64.rpm
[ 4] asr9k-parser-2.0.0.0-r751.x86_64.rpm
[ 5] asr9k-isis-2.1.0.0-r751.x86_64.rpm
[ 6] asr9k-mcast-3.0.0.0-r751.x86_64.rpm
[ 7] asr9k-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
[ 8] asr9k-ospf-2.0.0.0-r751.x86_64.rpm
[ 9] asr9k-li-1.0.0.0-r751.x86_64.rpm
[10] asr9k-eigrp-1.0.0.0-r751.x86_64.rpm
[11] asr9k-k9sec-3.1.0.0-r751.x86_64.rpm
Following XR x86_64 rpm(s) will be used for building Golden ISO:

        (+) asr9k-ospf-2.0.0.0-r751.x86_64.rpm
        (+) asr9k-bgp-2.0.0.0-r751.x86_64.rpm
        (+) asr9k-parser-2.0.0.0-r751.x86_64.rpm
        (+) asr9k-mcast-3.0.0.0-r751.x86_64.rpm
        (+) asr9k-li-1.0.0.0-r751.x86_64.rpm
        (+) asr9k-eigrp-1.0.0.0-r751.x86_64.rpm
        (+) asr9k-mgbl-3.0.0.0-r751.x86_64.rpm
        (+) asr9k-mpls-2.1.0.0-r75114I.x86_64.rpm
        (+) asr9k-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
        (+) asr9k-isis-2.1.0.0-r751.x86_64.rpm
        (+) asr9k-k9sec-3.1.0.0-r751.x86_64.rpm

        ...RPM signature check [PASS]

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

        (-) asr9k-parser-2.0.0.0-r751.x86_64.rpm
        (-) asr9k-bgp-2.0.0.0-r751.x86_64.rpm

        ...RPM compatibility check [PASS]

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

XR rpms:
        asr9k-mcast-3.0.0.0-r751.x86_64.rpm
        asr9k-mgbl-3.0.0.0-r751.x86_64.rpm
        asr9k-isis-2.1.0.0-r751.x86_64.rpm
        asr9k-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
        asr9k-eigrp-1.0.0.0-r751.x86_64.rpm
        asr9k-mpls-2.1.0.0-r751.x86_64.rpm
        asr9k-ospf-2.0.0.0-r751.x86_64.rpm
        asr9k-li-1.0.0.0-r751.x86_64.rpm
        asr9k-k9sec-3.1.0.0-r751.x86_64.rpm

        ...Golden ISO creation SUCCESS.

Golden ISO Image Location: /var/tmp/giso/gisobuild-toolkit-master/src/output_gisobuild/
                           asr9k-golden-x-7.5.1-dockerbasedgiso.iso
```

View that the GISO file is created succesfully.

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

[root@xr src]# cd output_gisobuild/
[root@xr output_gisobuild]# ls
img_built_name.txt logs asr9k-golden-x-7.5.1-dockerbasedgiso.iso
rpms_packaged_in_giso.txt
```

Example: Build YAML-Based GISO Image

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

The following example shows a sample YAML template:

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

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

user-content:
```

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

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

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

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

```
packages:
    iso: /auto/751_repo/asr9k-mini-x-7.5.1.iso
    repo:
        - /auto/751_repo/
    pkglist:
        - asr9k-bgp-2.0.0.0-r751.x86_64.rpm
        - asr9k-eigrp-1.0.0.0-r751.x86_64.rpm
        - asr9k-isis-2.1.0.0-r751.x86_64.rpm
        - asr9k-li-1.0.0.0-r751.x86_64.rpm
        - asr9k-mcast-3.0.0.0-r751.x86_64.rpm
        - asr9k-mgbl-3.0.0.0-r751.x86_64.rpm
        - asr9k-mpls-2.1.0.0-r751.x86_64.rpm
        - asr9k-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
        - asr9k-ospf-2.0.0.0-r751.x86_64.rpm
        - asr9k-parser-2.0.0.0-r751.x86_64.rpm
        - asr9k-k9sec-3.1.0.0-r751.x86_64.rpm
        - asr9k-mcast-3.0.0.1-r751.CSCxr33333.x86_64.rpm
        - asr9k-os-5.0.0.1-r751.CSCxr11111.x86_64.rpm
        - asr9k-sysadmin-hostos-7.5.1-r751.CSCho99999.admin.x86_64.rpm
        - asr9k-sysadmin-hostos-7.5.1-r751.CSCho99999.host.x86_64.rpm
        - asr9k-sysadmin-topo-7.5.1-r751.CSCcv55555.x86_64.rpm
        - asr9k-sysadmin-system-7.5.1-r751.CSCcv44444.x86_64.rpm
        - openssh-scp-6.6p1.p1-r0.5.0.r751.CSCtp11111.xr.x86_64.rpm
        - cisco-klm-zermatt-0.1.p1-r0.0.r751.CSCtp11111.xr.x86_64.rpm

    remove_rpms: []

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

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

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

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

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

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

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

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

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

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

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

What to do next

Install the GISO image on the router.

Install Golden ISO

Golden ISO (GISO) automatically performs the following actions:

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

Procedure

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

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

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

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

  The following example shows the logs from installation of GISO using PXE boot:

  ```
  ...
  Fri Dec  2 19:18:03 UTC 2016: ---Starting to prepare host logical volume---
  ...
  Fri Dec 02 19:18:14 UTC 2016: Skipping tp base rpm(openssh-scp-6.6p1-r0.0.host.x86_64.rpm) from
   installation
  Fri Dec 02 19:18:14 UTC 2016: Skipping tp base rpm(kernel-modules-3.14-r0.1.host.x86_64.rpm) from
   installation
  Fri Dec 02 19:18:15 UTC 2016: Installing asr9k-sysadmin-hostos-6.1.3-r613.CSChu77777.host.x86_64
            [SUCCESS]
  ```

```
...

Fri Dec  2 19:18:23 UTC 2016: ---Starting to prepare calvados logical volume---

...

Fri Dec 02 19:18:48 UTC 2016: Skipping tp base rpm(openssh-scp-6.6p1-r0.0.admin.x86_64.rpm) from
 installation
Fri Dec 02 19:18:48 UTC 2016: Skipping tp base rpm(kernel-modules-3.14-r0.1.admin.x86_64.rpm)
from installation
Fri Dec 02 19:18:49 UTC 2016: Installing asr9k-sysadmin-system-6.1.3-r613.CSCcv44444.x86_64
            [SUCCESS]
Fri Dec 02 19:18:50 UTC 2016: Installing asr9k-sysadmin-shared-6.1.3-r613.CSCcv33333.x86_64
            [SUCCESS]
Fri Dec 02 19:18:51 UTC 2016: Installing asr9k-sysadmin-hostos-6.1.3-r613.CSChu77777.admin.x86_64
          [SUCCESS]

...

Fri Dec  2 19:19:07 UTC 2016: ---Starting to prepare repository---
Fri Dec  2 19:19:11 UTC 2016: File system creation on /test took 3 seconds
Fri Dec  2 19:19:11 UTC 2016: Copying /iso/host.iso to repository /iso directory
Fri Dec  2 19:19:11 UTC 2016: Copy Host rpms to repository
Fri Dec  2 19:19:13 UTC 2016: Copying /iso/asr9k-sysadmin.iso to repository /iso directory
Fri Dec  2 19:19:13 UTC 2016: Copy Sysadmin rpms to repository
Fri Dec  2 19:19:16 UTC 2016: Copy HostOs rpms to repository
Fri Dec  2 19:19:16 UTC 2016: Copy XR rpms to repository
Fri Dec  2 19:19:16 UTC 2016: Copy giso_info.txt to repository
Fri Dec  2 19:19:17 UTC 2016: Copying /iso/asr9k-xr.iso to repository /iso directory
Fri Dec  2 19:19:21 UTC 2016: Copying all ISOs to repository took 10 seconds

...
```

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

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

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

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

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

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

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

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

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

```
Router#admin

sysadmin-vm:0_RSP0#run chvrf 0 ssh rp0_admin
```

```
[sysadmin-vm:0_RSP0:~]$ssh my_host
[host:~]$cd /misc/disk1/
[host:~]$./usb-install-712-or-latest.sh asr9k-goldenk9-x64-7.0.2-dr.iso /dev/sdc EFI

Preparing USB stick for EFI
parted gpt: Failed to create partition - continuing ...
Create filesystem on /dev/sdc1
Mounting source iso at //misc/disk1/cdtmp.CnuKnA
Mounting destination /dev/sdc1 at //misc/disk1/usbdev.SSBb4R
Copying image to USB stick
Initrd path is //misc/disk1/cdtmp.CnuKnA/boot/initrd.img
Getting boot
3749342 blocks
Copying boot
Copying initrd.img
Copying signature.initrd.img
Copying certs
Creating grub files
Copying /misc/disk1/asr9k-goldenk9-x64-7.0.2-dr.iso in USB Stick
USB stick set up for EFI boot!
```

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

The options to upgrade the system are as follows:

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

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

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

```
sysadmin-vm:0_RP0#show install repository all
Admin repository
---------------------
 asr9k-sysadmin-6.1.1

 asr9k-sysadmin-hostos-6.1.1-r611.CSCcv10001.admin.x86_64
 asr9k-sysadmin-system-6.1.1-r611.CSCcv10005.x86_64
....

  XR repository
------------------
 asr9k-iosxr-mgbl-3.0.0.0-r611.x86_64
```

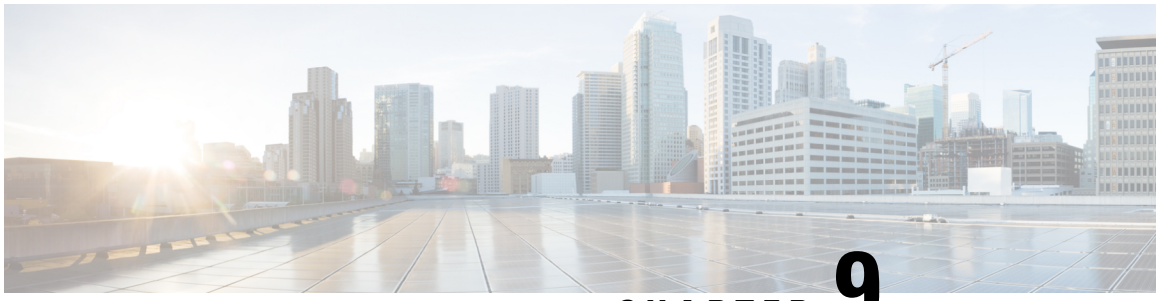
```
 asr9k-xr-6.1.1
....

  Host repository
---------------------
 host-6.1.1
```

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

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

```
Router#show install package asr9k-goldenk9-x64-6.1.3

Sun Dec  4 13:52:48.279 UTC
This may take a while ...
    ISO Name: asr9k-goldenk9-x64-6.1.3
    ISO Type: bundle
    ISO Bundled: asr9k-mini-x64-6.1.3
    Golden ISO Label: temp
    ISO Contents:
        ISO Name: asr9k-xr-6.1.3
        ISO Type: xr
        rpms in xr ISO:
            iosxr-os-asr9k-64-5.0.0.0-r613
            iosxr-ce-asr9k-64-3.0.0.0-r613
            iosxr-infra-asr9k-64-4.0.0.0-r613
            iosxr-fwding-asr9k-64-4.0.0.0-r613
            iosxr-routing-asr9k-64-3.1.0.0-r613
        ...

        ISO Name: asr9k-sysadmin-6.1.3
        ISO Type: sysadmin
        rpms in sysadmin ISO:
            asr9k-sysadmin-topo-6.1.3-r613
            asr9k-sysadmin-shared-6.1.3-r613
            asr9k-sysadmin-system-6.1.3-r613
            asr9k-sysadmin-hostos-6.1.3-r613.admin
        ...

        ISO Name: host-6.1.3
        ISO Type: host
        rpms in host ISO:
            asr9k-sysadmin-hostos-6.1.3-r613.host

        Golden ISO Rpms:
            xr rpms in golden ISO:
                asr9k-k9sec-x64-2.2.0.1-r613.CSCxr33333.x86_64.rpm
                openssh-scp-6.6p1.p1-r0.0.CSCtp12345.xr.x86_64.rpm
                openssh-scp-6.6p1-r0.0.xr.x86_64.rpm
                asr9k-mpls-x64-2.1.0.0-r613.x86_64.rpm
                asr9k-k9sec-x64-2.2.0.0-r613.x86_64.rpm

            sysadmin rpms in golden ISO:
                asr9k-sysadmin-system-6.1.3-r613.CSCcv11111.x86_64.rpm
                openssh-scp-6.6p1-r0.0.admin.x86_64.rpm
                openssh-scp-6.6p1.p1-r0.0.CSCtp12345.admin.x86_64.rpm

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

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



CHAPTER 9

Deploy Router Using Classic ZTP

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

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

ZTP provides multiple options, such as:

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

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

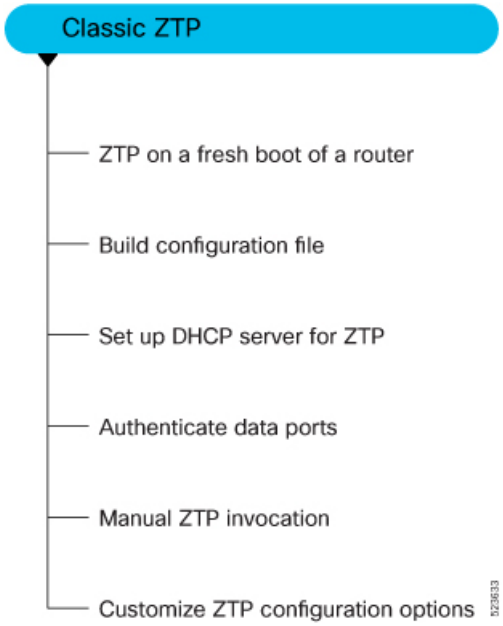
- Using ZTP to install Puppet

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 108
- **Manual Invocation:** Use this method when you want to forcefully initiate ZTP on a fully configured device. See Invoke ZTP Manually, on page 112.

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

Figure 10: Workflow to Configure Classic ZTP



- Build your configuration file, on page 94
- Limitations of temporary ZTP files, on page 102
- Authentication on Data Ports, on page 102
- Set Up DHCP Server, on page 104
- Customize ZTP Initialization File, on page 106
- Zero Touch Provisioning on a Fresh Boot of a Router, on page 108
- Invoke ZTP Manually, on page 112

Build your configuration file

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

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

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

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

The following is the sample configuration file:

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

hostname ios
alias exec al show alarms brief system active

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

Create User Script

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

Build your ZTP script with either shell and python.

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

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

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

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

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

ZTP Shell Utilities

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

API Implementation Behavior

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

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

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

ZTP Helper Python Library

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

ZtpHelpers Class Methods

Following are utility methods of the `ZtpHelpers` class:

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

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

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

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

```
Class Method for setting the network namespace
           :param cls: Reference to the class ZtpHelpers
```

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

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

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

- `toggle_debug(self, enable):`

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

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

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

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

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

- `setup_syslog(self):`

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

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

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

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

```
Apply Configuration to XR using a file
            :param file: Filepath for a config file
```

```
                        with the following structure:
                        !
                        XR config command
                        !
                        end

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

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

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

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

```
 Replace XR Configuration using a file

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

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

API Implementation Behavior

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

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

Example

The following shows the sample script in phyton

```
[testenv]$ python sample_ztp_script.py

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


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


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

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


###### Debugs Disabled ######


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

Building configuration...
{'output': ['!! IOS XR Configuration version = 6.2.1.21I',
            '!! Last configuration change at Sat Dec 17 04:23:25 2016 by UNKNOWN',
            '!',
            'hostname customer2',
            'username root',
```

```
            'group root-lr',
            'group cisco-support',
            'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1',
            '!',
            'username noc',
            'group root-lr',
            'group cisco-support',
            'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1',
            '!',
            'username netops',
            'group root-lr',
            'group cisco-support',
            'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1',
            '!',
            'username netops2',
            'group root-lr',
            'group cisco-support',
            'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1',
            '!',
            'username netops3',
            'group root-lr',
            'group cisco-support',
            'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1',
            '!',
            'cdp',
            'service cli interactive disable',
            'interface MgmtEth0/RP0/CPU0/0',
            'ipv4 address 11.11.11.59 255.255.255.0',
            '!',
            'interface TenGigE0/0/0/0/24',
            'shutdown',
            '!',
            'interface TenGigE0/0/0/0/25',
            'shutdown',
            '!',

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

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

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

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

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

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

{'output': ['!! SYNTAX/AUTHORIZATION ERRORS: This configuration failed due to',
            '!! one or more of the following reasons:',
            '!!  - the entered commands do not exist,',
```

```
'!!  - the entered commands have errors in their syntax,',
'!!  - the software packages containing the commands are not active,',
```

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

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.

Authentication on Data Ports

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

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

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

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

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

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

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

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

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

```
  option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/cisco-2/image.iso";
 }
 else {
   #option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/cisco-2/cisco-mini-x.iso.sh";
   option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/cisco-2/ztp.cfg";
  }
}
```

Set Up DHCP Server

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

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

- **Option 60**: “vendor-class-identifier” : Used to Identify the following four elements:
  - The type of client: For example, PXEClient
  - The architecture of The system (Arch): For example: 00009 Identify an EFI system using a x86-64 CPU
  - The Universal Network Driver Interface (UNDI):

    For example 003010 (first 3 octets identify the major version and last 3 octets identify the minor version)
  - The Product Identifier (PID):
- **Option 61**: “dhcp-client-identifier” : Used to identify the Serial Number of the device.
- **Option 66** : Used to request the TFTP server name.
- **Option 67**: Used request the TFTP filename.
- **Option 97**: “uuid” : Used to identify the Universally Unique Identifier a 128-bit value (not usable at this time)

Example

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

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

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

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

```
    }
}
```

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

The DHCP server responds with the following DHCP options:

- DHCPv4 using BOOTP filename to supply script/config location.
- DHCPv4 using Option 67 (bootfile-name) to supply script/config location.
- DHCPv6 using Option 15: If you have configured this option for the server to identify ztp requests, ensure that you update the server configuration, for Linux or ISC servers. Sample server-side configuration required to check user-class for ZTP is shown in the following example:

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

- DHCPv6 using Option 59 (OPT_BOOTFILE_URL) to supply script/config location

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

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

######### Network 11.11.11.0/24 #################
shared-network 11-11-11-0 {

####### Pools ##############
  subnet 11.11.11.0 netmask 255.255.255.0 {
    option subnet-mask 255.255.255.0;
 option broadcast-address 11.11.11.255;
 option routers 11.11.11.2;
 option domain-name-servers 11.11.11.2;
 option domain-name "cisco.local";
 # DDNS statements
   ddns-domainname "cisco.local.";
 # use this domain name to update A RR (forward map)
   ddns-rev-domainname "in-addr.arpa.";
   # use this domain name to update PTR RR (reverse map)

    }

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

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


  pool {
      allow members of "cisco";
      range 11.11.11.47 11.11.11.50;
      next-server 11.11.11.2;

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


      if exists user-class and option user-class =  "xr-config"
```

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

           }
         }


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

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

In Cisco IOS XR Release 7.3.2 and later, use:

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

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

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

Customize ZTP Initialization File

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

- `ZTP`: You can enable or disable ZTP at boot using CLI or by editing the *ztp.ini* file.
- `Retry`: Set the ZTP DHCP retry mechanism: The available values are infinite and once.
- `Fetcher Priority`: Fetcher defines which port ZTP should use to get the provisioning details. By default, each port has a fetcher priority defined in the *ztp.ini* file. You can modify the default priority of the fetcher. Allowed range is 0–10.

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

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`.

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 94.

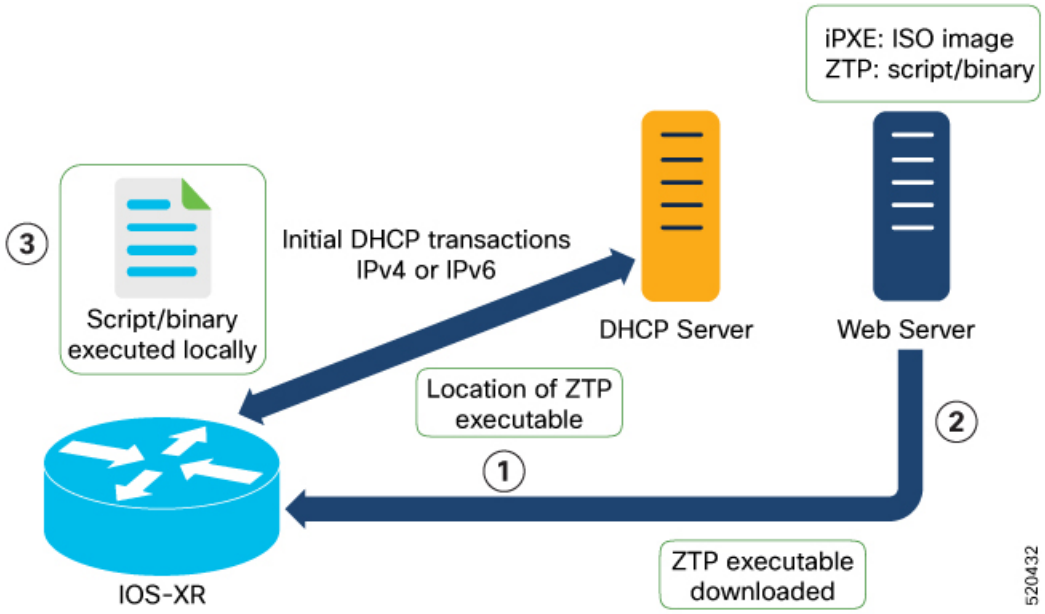
Note When you use a USB flash drive as a source for ZTP, you can't use the script file for provisioning. The script file isn't supported for USB fetcher.

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

Fresh Boot Using DHCP

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

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



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

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

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

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

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

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

   DHCP server should be configured to respond with the DHCP options.
   - DHCPv4 using BOOTP filename to supply script/config location.
   - DHCPv4 using Option 67 (bootfile-name) to supply script/config location.
   - DHCPv6 using Option 59 (OPT_BOOTFILE_URL) to supply script/config location

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

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

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

Invoke ZTP Manually

You can invoke Zero Touch Provisioning (ZTP) manually through the Command Line Interface. This method is Ideal for verifying the ZTP configuration without a reboot. This manual approach helps you to provision the router in stages. To invoke ZTP on an interface (data ports or management port), you don't have to bring up and configure the interface first.

Even when the interface is down, you can run the `ztp initiate` command, and the ZTP script will bring it up and invoke `dhclient`. Hence, ZTP can run on all interfaces irrespective of their availability.

Use the following commands to manually invoke the ZTP commands and to force ZTP to run on all interfaces:

- **ztp initiate** — Invokes a new ZTP DHCP session. Logs can be found in **/disk0:/ztp/ztp.log**.

  Configuration Example:

```
Router#ztp initiate debug verbose interface HundredGigE 0/0/0/24
Invoke ZTP? (this may change your configuration) [confirm] [y/n] :
```

- **ztp terminate** —Terminates any ZTP session in progress.

  Configuration Example:

```
Router #ztp terminate verbose
Mon Oct 10 16:52:38.507 UTC
Terminate ZTP? (this may leave your system in a partially configured state) [confirm]
[y/n] :y
ZTP terminated
```

- **ztp enable** —Enables the ZTP at boot.

  Configuration Example:

```
Router#ztp enable
Fri Jul 12 16:09:02.154 UTC
Enable ZTP? [confirm] [y/n] :y
ZTP Enabled.
```

- **ztp disable** —Disables the ZTP at boot.

  Configuration Example:

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

- **ztp clean** —Removes only the ZTP state files.

  Configuration Example:

  ```
  Router#ztp clean verbose
  Mon Oct 10 17:03:43.581 UTC
  Remove all ZTP temporary files and logs? [confirm] [y/n] :y
  All ZTP files have been removed.
  If you now wish ZTP to run again from boot, do 'conf t/commit replace' followed by
  reload.
  ```

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

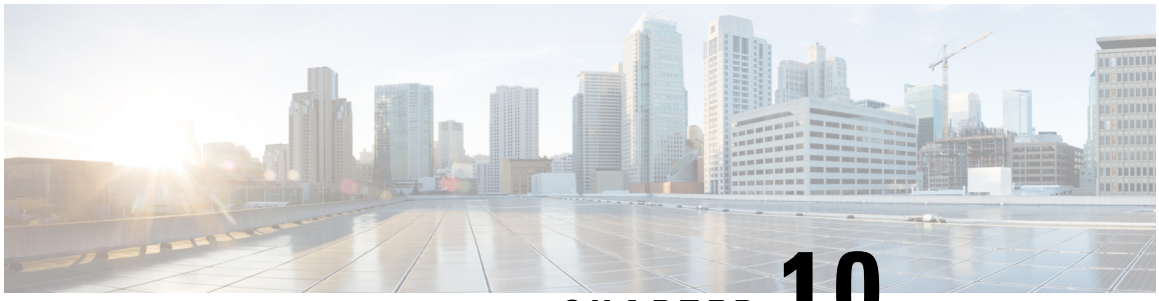
Configuration

This task shows the most common use case of manual ZTP invocation: invoke ZTP.

1. Invoke DHCP sessions on all data ports which are up or could be brought up. ZTP runs in the background. Use show logging or look at /disk0:/ztp/ztp.log to check progress.

Configuration Example:

```
Router# ztp initiate dataport
```



CHAPTER 10

Release-specific Caveats and Workarounds

This section lists the caveats and workarounds when setting up or upgrading the software for each Cisco IOS XR release.

- Caveats and Workarounds, on page 115

Caveats and Workarounds

Release 7.4.1

There are no caveats for system upgrade to Cisco IOS XR Release 7.4.1.

Release 7.4.2

CSCvp19524: When downgrading from Cisco IOS XR Release 7.4.2 to any release lower than Cisco IOS XR Release 7.0.1, it is necessary to remove AAA user configurations from SYSADMIN/XR VMs before proceeding with the downgrade.

Release 7.5.1

CSCvp19524: When downgrading from Cisco IOS XR Release 7.5.1 to any release lower than Cisco IOS XR Release 7.0.1, it is necessary to remove AAA user configurations from SYSADMIN/XR VMs before proceeding with the downgrade.

Release 7.5.2

CSCvp19524: When downgrading from Cisco IOS XR Release 7.5.2 to any release lower than Cisco IOS XR Release 7.0.1, it is necessary to remove AAA user configurations from SYSADMIN/XR VMs before proceeding with the downgrade.

Release 7.6.2

CSCvp19524: When downgrading from Cisco IOS XR Release 7.6.2 to any release lower than Cisco IOS XR Release 7.0.1, it is necessary to remove AAA user configurations from SYSADMIN/XR VMs before proceeding with the downgrade.

Release 7.7.2

CSCvp19524: When downgrading from Cisco IOS XR Release 7.7.2 to any release lower than Cisco IOS XR Release 7.0.1, it is necessary to remove AAA user configurations from SYSADMIN/XR VMs before proceeding with the downgrade.

Release 7.8.1

CSCvp19524: When downgrading from Cisco IOS XR Release 7.8.1 to any release lower than Cisco IOS XR Release 7.0.1, it is necessary to remove AAA user configurations from SYSADMIN/XR VMs before proceeding with the downgrade.

Release 7.8.2

CSCvp19524: When downgrading from Cisco IOS XR Release 7.8.2 to any release lower than Cisco IOS XR Release 7.0.1, it is necessary to remove AAA user configurations from SYSADMIN/XR VMs before proceeding with the downgrade.

Release 7.9.1

CSCvp19524: When downgrading from Cisco IOS XR Release 7.9.1 to any release lower than Cisco IOS XR Release 7.0.1, it's necessary to remove AAA user configurations from SYSADMIN/XR VMs before proceeding with the downgrade.

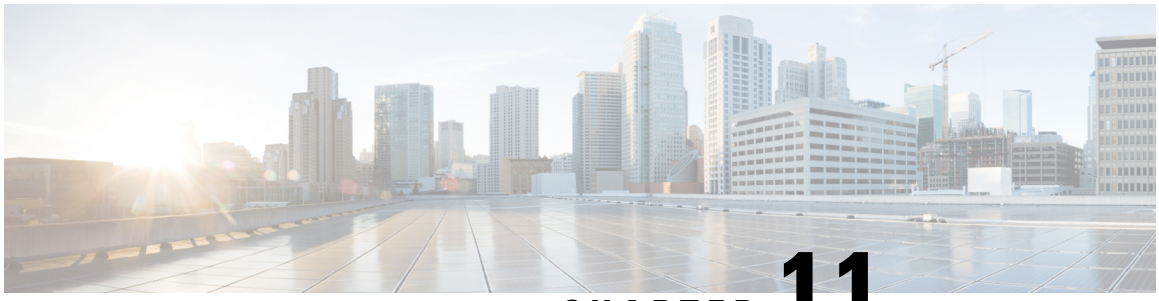
1. **CSCwd30936:** When downgrading from Cisco IOS XR Release 7.9.1 to an older release, the "ema_server_sdr" process may crash.

Release 7.9.2

1. **CSCvp19524:** When downgrading from Cisco IOS XR Release 7.9.2 to any release lower than Cisco IOS XR Release 7.0.1, it is necessary to remove AAA user configurations from SYSADMIN/XR VMs before proceeding with the downgrade.
2. **CSCwd30936:** When downgrading from Cisco IOS XR Release 7.9.2 to an older release, the "ema_server_sdr" process may crash.

Release 7.10.1

1. **CSCvp19524:** When downgrading from Cisco IOS XR release 7.10.1 to any release lower than Cisco IOS XR Release 7.0.1, it is necessary to remove AAA user configurations from SYSADMIN/XR VMs before proceeding with the downgrade.
2. **CSCwd30936:** When downgrading from Cisco IOS XR Release 7.10.1 to an older release, the "ema_server_sdr" process may crash.
3. **SR695094155:** When upgrading the software version of an ASR 9000 router with an mLACP/ICCP Redundancy Model setup, ensure that both the active and standby nodes are upgraded to the same IOS XR version.



CHAPTER 11

Upgrading and Managing Cisco IOS XR Software

Cisco IOS XR software is divided into software packages so that you can select which features run on your router. This module describes the concepts and tasks necessary to add feature packages, upgrade the active set of packages, roll back to a previously active set of packages, and perform other related package management tasks.

For complete descriptions of the commands listed in this module, see Related Documents, on page 174. To locate documentation for other commands that might appear in the course of performing a configuration task, search online in *Cisco ASR 9000 Series Aggregation Services Router Commands Master List*.

Table 9: Feature History for Upgrading and Managing Cisco IOS XR Software

Release	Modification
Release 3.7.2	The feature was introduced.
Release 4.0.0	A procedure to upgrade software from Cisco IOS XR Release 3.x was introduced. See Upgrading to Cisco IOS XR Software Release 4.0, on page 134. Support for installation commands was removed from EXEC mode. The ability to install software on a specific SDR was removed.
Release 6.3.1	Support for parallel FPD upgrade for power modules was added.

This module contains the following topics:

- Overview of Cisco IOS XR Software Packages, on page 117
- Package Management Procedures, on page 153
- Additional References, on page 174

Overview of Cisco IOS XR Software Packages

Cisco IOS XR software is divided into software packages so that you can select which features run on your router. Each package contains the components to perform a specific set of router functions, such as routing, security, or modular services card (MSC) support. Bundles are groups of packages that can be downloaded as a set. For example, Cisco IOS XR Unicast Routing Core Bundle (known as *mini*) provides the main packages for use on every router.

Adding a package to the router does not affect the operation of the router—it only copies the package files to a local storage device on the router, known as the *boot device* (such as the compact flash drive). To make the package functional on the router, you must activate it for one or more cards.

To upgrade a package, you activate a newer version of the package. When the automatic compatibility checks have been passed, the new version is activated, and the old version is deactivated.

Note Activating a software maintenance upgrade (SMU) does not cause any earlier SMUs or the package to which the SMU applies to be automatically deactivated.

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

To downgrade a package, you activate an older version of the package. When the automatic compatibility checks have been passed, the older version is activated, and the newer version is deactivated.

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

Note For more information on the features and components included in each package, refer to the release notes.

Package Installation Envelopes

Package Installation Envelopes (PIEs) are nonbootable files that contain a single package or a set of packages (called a *composite package* or *bundle*). Because the files are nonbootable, they are used to add software package files to a running router.

PIE files have a `pie` extension. When a PIE file contains software for a specific bug fix, it is called a *software maintenance upgrade* (SMU).

Note Files with the `vm` extension are bootable installation files used only to replace all current Cisco IOS XR software. These files are installed from ROM Monitor mode, which causes significant router downtime. Cisco Systems recommends installing or upgrading software packages only using PIE files as described in this document. For more information on `vm` files, see *ROM Monitor Configuration Guide for Cisco ASR 9000 Routers*.

Packages in the Cisco IOS XR Unicast Routing Core Bundle

Every router includes a basic set of required packages contained in the Cisco IOS XR Unicast Routing Core Bundle. Additional optional packages can be added and activated on the router to provide specific features.

The packages contained in the Cisco IOS XR Unicast Routing Core Bundle are as follows:

- Operating system (OS) and minimum boot image (MBI)—Kernel, file system, memory management, and other slow changing core components.
- Base—Interface manager, system database, checkpoint services, configuration management, other slow-changing components.
- Infra—Resource management: rack, fabric.
- Routing—RIB, BGP, ISIS, OSPF, EIGRP, RIP, RPL, and other routing protocols.
- Forwarding—FIB, ARP, QoS, ACL, and other components.
- LC— Line card drivers.

The filename for this bundle is: `asr9k-mini.pie-`*`version`*.

Refer to the release notes for additional information on the specific features provided by each package.

PIE Filenames and Version Numbers

PIE filenames have two formats: one for composite-package PIEs (bundles) and one for single-package PIEs. A *composite-package file* is a PIE file that contains multiple packages.

Note Hyphens in the filename are part of the filename.

Table 10: PIE Filenames, on page 119 shows the filenames for available PIE types.

Table 10: PIE Filenames

Software Delivery Type	Filename	Example
Composite (Bundle) PIE	*platform-composite_name*.pie-*major.minor.maintenance*	`asr9k-mini.pie-3.7.2`
Single package PIE	*platform-package_type*.-p.pie-*major.minor.maintenance*	`asr9k-mpls.pie-3.7.2`
Composite SMU	comp-*platform-composite_name.ddts*.pie	`comp-asr9k-001.CSCec98xxx.pie`
Single package SMU	*platform-package_type-major.minor.maintenance.ddts*.pie	`asr9k-base-3.7.2.CSCei45xxx.pie`
Note A SMU composite name usually is “001”, which means the SMU is the first SMU for that DDTS. In rare cases in which the same DDTS requires multiple composite SMUs, a second composite version number is released as “002”. In the previous example, a second composite SMU “comp-002.CSCec98766” would be created for DDTS CSCec98766.		

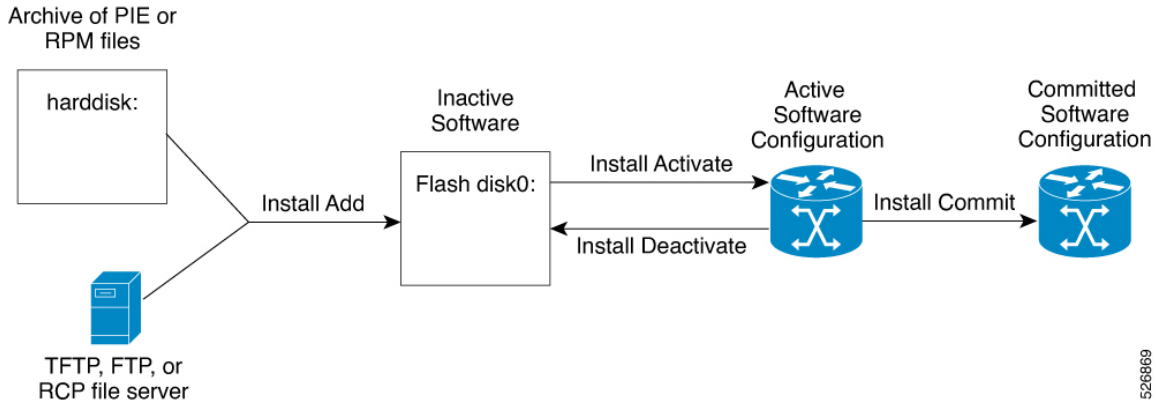
Summary of Package Management

The general procedure for adding optional packages, upgrading a package or package set, or downgrading packages on the router is as follows:

1. Copy the package file or files to a local storage device or file server.
2. Add the package or packages on the router using the command **install add**.
3. Activate the package or packages on the router using the **install activate** command.
4. Commit the current set of packages using the **install commit** command.

Figure 11: Process to Add, Activate, and Commit Cisco IOS XR Software Packages, on page 120 illustrates key steps in the package management process.

Figure 11: Process to Add, Activate, and Commit Cisco IOS XR Software Packages



Adding Packages

Use the **install add** command to unpack the package software files from a PIE file and copy them to the boot device (usually disk0:).

- The package software files are added to the boot device of the DSC of the router from either administration EXEC or EXEC mode.

Note The disk that holds the unpacked software files is also known as the *boot device*. By default, flash disk0: is used as the boot device. To use an alternate storage device, such as flash disk1:, see the *Router Recovery with ROM Monitor* module of *ROM Monitor Configuration Guide for Cisco ASR 9000 Routers*. Remember that all RSPs in a system must use the same boot device. If the boot device on the primary RSP is flash disk0:, then the standby RSP must also have a flash disk0:.

Activating Packages

Software packages remain inactive until activated with the **install activate** command.

After a package has been added to the router , use the **install activate** command to activate the package or SMUs for all valid cards. Information within the package is used to verify compatibility with the target cards

and with the other active software. Actual activation is performed only after the package compatibility and application programming interface (API) compatibility checks have been passed.

Note SDR-specific activation is supported for specific packages and upgrades, such as optional packages and SMUs. Packages that do not support SDR-specific EXEC mode activation can only be activated from administration EXEC mode.

Activating a Package on the Router

To activate a package on your router, use the **install activate** command in either administration EXEC mode or EXEC mode. If used in administration EXEC mode, the **install activate** command also activates the package on all administration plane nodes and resources, including service processors (SPs), fabric SCs, fan controllers, alarm modules, and power modules.

Note To enter administration EXEC mode, you must be logged in to the owner secure domain router (SDR) and have root-system access privileges.

You can also activate a package using the **install activate** command from EXEC mode.

Activating Multiple Packages or SMUs

To install multiple packages or software maintenance upgrades (SMUs) with a single command, use the **install activate** command and either specify up to 16 packages by repeating *device: package* arguments or use wildcard syntax to specify multiple packages. Some SMUs may require a reload. If the operation requires a node reload, the user is prompted before the installation operation occurs.

Note After activating SMU CSCwc03813, ensure that you either reload the Line Card or remove and reapply the existing Access Control Lists, for the updates to take effect.

Activating All Packages Added in a Specific Operation

To install all packages that were added in a specific **install add** operation, use the **install activate** command with the **id** *add-id* keyword and argument, specifying the operation ID of the **install add** operation. You can specify up to 16 operations in a single command.

Adding and Activating a Package with a Single Command

To add and activate a package with a single command, use the **install add** command with the **activate** keyword.

- To add and activate a package, enter the **install add** command with the **activate** keyword from administration EXEC mode.
- To add and activate a package in EXEC mode (where supported), enter the **install add** command with the **activate** keyword.

Multiple Disks Support during Installations

In installations on platforms where Cisco IOS XR Software is supported, only a single disk is used as an install device; that is, either disk0 or disk1. When multiple packages are installed on a single disk, it results in space constraints. To resolve this space limitation, the disk supported for the install operations has been extended to another disk called the disk1. When installing multiple packages, this feature enables you to choose between disk0 and disk1.

To add packages to a specific disk name, use the **install media** command in the admin configuration mode.

```
RP/0/RSP0/CPU0: router (admin) # install media disk1
```

Restrictions

- Before enabling the addition of disk1 through the **install media** command, the disk mirroring feature should be explicitly disabled. For details regarding disk mirroring, see the Disk Mirroring chapter.
- All single version packages should be installed into one disk; that is, either disk0 or disk1.
- When downgrading to an image that does not support extended disk, the rollback points of the extended disk will not be available on the downgraded image. For example, assume a case where the version1 (V1) image does not support the extended disk functionality and version2 (V2) image supports the functionality. Upgrading from V1(disk0) to V2(disk1), in such a case, makes the rollback points of V1 available on V2. However, when downgrading from V2(disk1) to V1(disk0), the rollback points of V2 will not be available on V1. For more information about the rollback feature and rollback points, see the Upgrading and Managing Software chapter.

Support for the Ignore Package Presence Check Option

During any software package upgrade in Cisco IOS XR Software, two versions of the packages get stored, both the previous version and the upgraded version. In Route Switch Processor 2 (RSP2), the disk space is insufficient to hold all packages of these two versions. To address this, a new optional keyword, **ignore-pkg-presence-check**, is added to the **install activate** command, which allows upgrading with lesser number of packages. For example, assume a case where version1 (V1) of the software consists of packages A, B, C, and D, and you want to upgrade to the version2 (V2) with only 3 packages (A, B, and C). The ignore-pkg-presence-check option allows only packages A, B, and C to be upgraded to V2 and deactivates package D of V1. Thus, an explicit deactivation of package D is not required and the user can add package D of V1 after upgrading to V2.

To upgrade software with lesser number of packages, use the **install activate [ignore-pkg-presence-check]** command in the admin mode.

```
RP/0/RSP0/CPU0: router(admin) # install activate [ignore-pkg-presence-check] V2 packages
```

Restrictions

The restrictions for this option are:

- The ignore-pkg-presence-check keyword is supported only with the **install activate** command and is not supported with the **install add activate** command.
- When you upgrade using the ignore-pkg-presence-check option, the deactivation of packages always happens synchronously, using the synchronous keyword in the **install deactivate** command.

Upgrading and Downgrading Packages

To upgrade a package, activate the latest version of the package; the previous version is automatically deactivated. To downgrade a package, activate the previous version of the package; the latest version is automatically deactivated.

Actual activation is performed only after compatibility checks have been passed.

Note

- Activating a software maintenance upgrade (SMU) does not cause previous versions of the SMUs, or the package to which the SMU applies, to be automatically deactivated.
- If you upgrade an ASR 9000 router with low RSP card memory, then the RSP440-TR route-switch processor and Cisco ASR 9000 2nd Generation line card can become inaccessible due to insufficient memory. Power cycling the router may help bring the router back to the up state.

Committing the Active Software Set

When a package is activated on the router, it becomes part of the current running configuration. To make the package activation persistent across reloads, enter the **install commit** command. On startup, the designated shelf controller (DSC) of the secure domain router (SDR) loads the committed software set.

- To commit the active software set from administration EXEC mode, use the **install commit** command with the **sdr Owner** keywords. Alternatively, use the **install commit** command without keywords or arguments.

Note If the system is restarted before the active software set is saved with the **install commit** command, the previously committed software set is used.

Rolling Back to a Previous Installation Operation

Although the term *commit* sounds final, the Cisco IOS XR software provides the flexibility to roll back the selected package set to previously saved package sets. Each time a package is activated or deactivated, a rollback point is created that defines the package set that is active after the package activation or deactivation. The software also creates a rollback point for the last committed package set. If you find that you prefer a previous package set over the currently active package set, you can use the **install rollback** command to make a previously active package set active again.

Preparing for Software Installation Operations

This section includes instructions to prepare for software installation operations.

Note Activation is performed only after the automatic package compatibility and API version compatibility checks have been passed. If a conflict is found, an on-screen error message is displayed.

Before you begin

Before adding or activating Cisco IOS XR software:

- Update the ROM Monitor software, if necessary.
- Determine if a software change is required.
- Verify that the new package is supported on your system. Some software packages require that other packages or package versions be activated, and some packages only support specific cards.
- Review the release notes for important information related to that release and to help determine the package compatibility with your router configuration.
- Verify that the system is stable and prepared for the software changes.

SUMMARY STEPS

1. **admin**
2. **show diag**
3. Update the ROMMON software if necessary.
4. **show install active**
5. **show install pie-info** *device:package* [**brief** | **detail** | **verbose**]
6. **verify packages**
7. **exit**
8. (Optional) **show system verify start**
9. (Optional) **show system verify** [**detail** | **report**]
10. **show clock**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 2	**show diag** **Example:** `RP/0/RSP0/CPU0:router(admin)# show diag`	Displays the ROMMON software version for all cards in the system. Verify that the correct ROMMON software version is installed before upgrading a Cisco IOS XR software package. **Note** See *Related Topics* for information regarding the required ROM Monitor (ROMMON) software version.
Step 3	Update the ROMMON software if necessary.	Updates the ROMMON software. For instructions, see *ROM Monitor Configuration Guide for Cisco ASR 9000 Routers*.

	Command or Action	Purpose
Step 4	**show install active** **Example:** `RP/0/RSP0/CPU0:router(admin)# show install active`	Displays the active software for the owner SDR. Use this command to determine what software should be added, upgraded or downgraded on the router, and to compare to the active software report after installation operations are complete. **Note** You can also display the active packages for a specific node, and view results in detailed or summary mode. See the *Software Package Management Commands on the Cisco ASR 9000 Series Router* module of *System Management Command Reference for Cisco ASR 9000 Series Routers* for more information.
Step 5	**show install pie-info** *device:package* [**brief** | **detail** | **verbose**] **Example:** `RP/0/RSP0/CPU0:router(admin)# show install pie-info disk1:/asr9k-mcast-p.pie-3.8.30`	Displays information imbedded in the package. The following keywords provide three levels of information: • **brief** (default)—Displays the expiration date of the file, the size, and the installed package name. The expiration date is used for certifying the package. • **detail**—Displays the package components, the compatible cards, the expiration date, file size, and the installed package name. • **verbose**—Displays information from the **detail** display and sub-component information. **Note** Always review the release notes for the software package for important information related to that release and to help determine the package compatibility with your router configuration.
Step 6	**verify packages** **Example:** `RP/0/RSP0/CPU0:router(admin)# install verify packages`	Verifies that there are no corrupted software files. The consistency of a previously installed software set is verified against the package file from which it originated. This command can be used as a debugging tool to verify the validity of the files that constitute the packages, to determine if there are any corrupted files. This command also checks for corruptions of installation state files and MBI image files. This command is particularly useful when issued after the activation of a package or upgrading the Cisco IOS XR software to a major release. **Note** The **install verify packages** command can take up to two minutes per package to process.
Step 7	**exit** **Example:**	Exits administration EXEC mode and returns to EXEC mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(admin)# exit`	
Step 8	(Optional) **show system verify start** **Example:** `RP/0/RSP0/CPU0:router# show system verify start`	Starts the system status check.
Step 9	(Optional) **show system verify** [**detail** | **report**] **Example:** `RP/0/RSP0/CPU0:router# show system verify`	Displays system status information. A variety of information is displayed including the memory and CPU usage, process status, protocol status, and other status information. Use this information to verify that the system is stable. Enter this command in EXEC mode. • **detail**—Displays additional information at the card and processor level, including actual numbers. • **report**—Displays the same information as the default **show system verify** command **Note** Although most of the output should display the status "OK," some processes may show other output, such as "Warning." This does not specifically indicate a problem. Contact your Cisco technical support representative for more information on the output of this command.
Step 10	**show clock** **Example:** `RP/0/RSP0/CPU0:router# show clock`	Verifies that the system clock is correct. Software operations use certificates based on router clock times.

Related Topics

Activation and Deactivation Prerequisites, on page 153

Examples

Verifying That the ROM Monitor Version Is Correct: Example

In the following example, the ROM Monitor software version is displayed in the "ROMMON:" field for each card.

Note For instructions to upgrade the ROM Monitor software, see *ROM Monitor Configuration Guide for Cisco ASR 9000 Routers*.

```
RP/0/RSP0/CPU0:router# admin
RP/0/RSP0/CPU0:router(admin)# show diag

Mon Jun 22 12:55:10.554 PST

NODE module 0/RSP0/CPU0 :

  MAIN:  board type 0x100302
  S/N:   FOC1230803H
  Top Assy. Number:   68-3160-04
  PID:   A2K-RSP-4G-HDD=
  UDI_VID:   VP4
  HwRev: V4.8
  New Deviation Number: 0
  CLEI:  IPUCARJBAA
  Board State : IOS XR RUN
  PLD:    Motherboard: N/A, Processor: 0x8004 (rev: 2.2), Power: N/A
  MONLIB: QNXFFS Monlib Version 3.2
  ROMMON: Version 1.0(20081208:173612) [ASR9K ROMMON]
  Board FPGA/CPLD/ASIC Hardware Revision:
    Compact Flash : V1.0
     XbarSwitch0  : V1.3
     XbarSwitch1  : V1.3
     XbarArbiter  : V1.0
     XbarInterface  : V0.0
     IntCtrl  : V1.14
     ClkCtrl  : V1.13
     PuntFPGA  : V1.5
     HD  : V3.0
     USB0  : V77.20
     USB1  : V77.20
     CPUCtrl  : V1.17
     UTI  : V1.6
     LIU  : V1.0
     MLANSwitch  : V0.0
     EOBCSwitch  : V2.0
     CBC (active partition)  : v1.2
     CBC (inactive partition)  : v1.1

NODE fantray 0/FT0/SP :

  MAIN:  board type 0x900211
  S/N:
  Top Assy. Number:   32-0000-00
  PID:
  UDI_VID:
  HwRev: V32.0
  New Deviation Number: 0
  CLEI:
  PLD:    Motherboard: N/A, Processor: N/A, Power: N/A
  ROMMON:
  Board FPGA/CPLD/ASIC Hardware Revision:
     CBC (active partition)  : v4.0
     CBC (inactive partition)  : v0.13

NODE fantray 0/FT1/SP :

  MAIN:  board type 0x900211
  S/N:
  Top Assy. Number:   32-0000-00
  PID:
  UDI_VID:
  HwRev: V32.0
```

```
  New Deviation Number: 0
  CLEI:
  PLD:    Motherboard: N/A, Processor: N/A, Power: N/A
  ROMMON:
  Board FPGA/CPLD/ASIC Hardware Revision:
     CBC (active partition)  : v4.0
     CBC (inactive partition)  : v0.13

NODE module 0/1/CPU0 :

   MAIN:  board type 0x20207
  S/N:   FOC123081J6
  Top Assy. Number:   68-3182-03
  PID:   A9K-40GE-B
  UDI_VID:   V1D
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:
  Board State : IOS XR RUN
  PLD:    Motherboard: N/A, Processor: 0x8004 (rev: 2.2), Power: N/A
  ROMMON: Version 1.0(20081208:174521) [ASR9K ROMMON]
  Board FPGA/CPLD/ASIC Hardware Revision:
     NP0  : V3.194
     NP1  : V3.194
     NP2  : V3.194
     NP3  : V3.194
     XbarInterface  : V18.4
     Bridge0  : V0.38
     Bridge1  : V0.38
     CPUCtrl  : V0.15
     USB  : V77.20
     PortCtrl  : V0.8
     PHYCtrl  : V0.6
     40 Port Gigabit Ethernet Daughter board  : V0.0
     CBC (active partition)  : v2.2
     CBC (inactive partition)  : v2.1

NODE module 0/4/CPU0 :

  MAIN:  board type 0x2020a
  S/N:   FOC123081JA
  Top Assy. Number:   68-3183-02
  PID:   A9K-8T/4-B
  UDI_VID:   V1D
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:  IPU3AE0CAA
  Board State : IOS XR RUN
  PLD:    Motherboard: N/A, Processor: 0x8004 (rev: 2.2), Power: N/A
  ROMMON: Version 1.0(20081208:174521) [ASR9K ROMMON]
  Board FPGA/CPLD/ASIC Hardware Revision:
     NP0  : V3.194
     NP1  : V3.194
     NP2  : V3.194
     NP3  : V3.194
     XbarInterface  : V18.4
     Bridge0  : V0.38
     Bridge1  : V0.38
     CPUCtrl  : V0.15
     USB  : V77.20
     PortCtrl  : V0.10
     PHYCtrl  : V0.7
     PHY0  : V0.16
     PHY1  : V0.16
```

```
     PHY2  : V0.16
     PHY3  : V0.16
     PHY4  : V0.16
     PHY5  : V0.16
     PHY6  : V0.16
     PHY7  : V0.16
     8 Port Ten Gigabit Ethernet Daughter board  : V0.0
     CBC (active partition)  : v2.2
     CBC (inactive partition)  : v2.1

NODE module 0/6/CPU0 :

  MAIN:  board type 0x20208
  S/N:   FHH12250033
  Top Assy. Number:   68-3184-02
  PID:   A9K-4T-B
  UDI_VID:   V1D
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:
  Board State : IOS XR RUN
  PLD:    Motherboard: N/A, Processor: 0x8004 (rev: 2.2), Power: N/A
  ROMMON: Version 1.0(20081208:174521) [ASR9K ROMMON]
  Board FPGA/CPLD/ASIC Hardware Revision:
     NP0  : V3.194
     NP1  : V3.194
     NP2  : V3.194
     NP3  : V3.194
     XbarInterface  : V18.4
     Bridge0  : V0.38
     Bridge1  : V0.38
     CPUCtrl  : V0.15
     USB  : V77.20
     PHY0  : V0.16
     PHY1  : V0.16
     PHY2  : V0.16
     PHY3  : V0.16
     PortCtrl  : V0.10
      PHYCtrl  : V0.7
     4 Port Ten Gigabit Ethernet Daughter board  : V0.0
     CBC (active partition)  : v2.2
     CBC (inactive partition)  : v2.1

NODE power-module 0/PM0/SP :

  MAIN:  board type 0xf00188
  S/N:
  Top Assy. Number:   341-00032-01
  PID:   A9K-3KW-AC
  UDI_VID:   V00
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:  ACACACACAC
  PLD:    Motherboard: N/A, Processor: N/A, Power: N/A
  ROMMON:
  Board FPGA/CPLD/ASIC Hardware Revision:

NODE power-module 0/PM1/SP :

  MAIN:  board type 0xf00188
  S/N:
  Top Assy. Number:   341-00032-01
  PID:   A9K-3KW-AC
  UDI_VID:   V00
```

```
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:  ACACACACAC
  PLD:    Motherboard: N/A, Processor: N/A, Power: N/A
  ROMMON:
  Board FPGA/CPLD/ASIC Hardware Revision:

NODE power-module 0/PM2/SP :

  MAIN:  board type 0xf00188
  S/N:
  Top Assy. Number:   341-00032-01
  PID:   A9K-3KW-AC
  UDI_VID:   V00
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:  ACACACACAC
  PLD:    Motherboard: N/A, Processor: N/A, Power: N/A
  ROMMON:
  Board FPGA/CPLD/ASIC Hardware Revision:

Rack 0 - ASR-9010 Chassis, Includes Accessories
  RACK NUM: 0
  S/N:
  PID:   ASR-9010 Backplane
  VID:   0.1
  Desc:  ASR-9010 Chassis, Includes Accessories
  CLEI:  NOCLEI
  Top Assy. Number:   68-1234-56
```

Displaying the Active Software for the Entire System: Example

The following example displays the active packages for the entire system. Use this information to determine if a software change is required:

```
RP/0/RSP0/CPU0:router(admin)# show install active summary

Mon Jun 22 13:01:46.438 PST
Default Profile:
  SDRs:
    Owner
  Active Packages:
    disk0:comp-asr9k-mini-3.9.0.12I
    disk0:asr9k-fpd-3.9.0.12I
    disk0:asr9k-k9sec-3.9.0.12I
    disk0:asr9k-mcast-3.9.0.12I
    disk0:asr9k-mgbl-3.9.0.12I
    disk0:asr9k-mpls-3.9.0.12I
```

Displaying Information About the Contents of a PIE File: Example

In the following example, information is displayed about the manageability PIE. This command displays the expiry date of the package, the cards supported by the package, and other details. Use

this information to verify the compatibility of the package with your system and other software packages.

Note A software activation is performed only after the automatic package compatibility and API version compatibility checks have been passed. If a conflict is found, an on-screen error message is displayed.

```
RP/0/RSP0/CPU0:router(admin)# show install pie-info disk1:/
asr9k-mgbl-p.pie-3.8.0 detail

Contents of pie file '/disk1:/asr9k-mgbl-p.pie-3.8.0':
    Expiry date       : Jan 19, 2007 02:55:56 UTC
    Uncompressed size : 17892613

    asr9k-mgbl-3.8.0
        asr9k-mgbl V3.8.0[00]  Manageability Package
        Vendor : Cisco Systems
        Desc   : Manageability Package
        Build  : Built on Wed May 10 08:04:58 UTC 2006
        Source : By edde-bld1 in /vws/aga/production/3.8.0/asr9k/workspace for c28
        Card(s): RP, DRP, DRPSC
        Restart information:
          Default:
            parallel impacted processes restart
        Components in package asr9k-mgbl-3.8.0, package  asr9k-mgbl:
            manageability-cwi V[r33x/2]  Craft Web Interface related binaries ae
            asr9k-feature-ipsla V[r33x/1]  IPSLA time stamping feature
            doc-asr9k-mgbl V[r33x/2]  Contains the man page documentation for asr9ks
 --More--
```

Verifying That There Are No Corrupted Software Files: Example

The following sample output verifies the consistency of the currently active software against the file from which it originated:

```
RP/0/RSP0/CPU0:router(admin)# install verify packages

Mon Jun 22 13:19:08.590 PST
Install operation 3 '(admin) install verify packages' started by user 'user'
via CLI at 13:19:08 DST Mon Jun 22 2009.
The install operation will continue asynchronously.
RP/0/RSP0/CPU0:router(admin)#Info:
This operation can take up to 2 minutes per package being verified.
Info:     Please be patient.

Info:     0/6/CPU0 [LC] [SDR: Owner]
Info:         meta-data: [SUCCESS] Verification Successful.
Info:         /install/asr9k-scfclient-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-os-mbi-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-mpls-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-mcast-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-lc-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-fwdg-3.9.0.12I: [SUCCESS] Verification Successful.
```

```
Info:         /install/asr9k-fpd-3.9.0.12I: [ERROR] Detected anomalies.
Info:         /install/asr9k-diags-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-base-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-admin-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:     0/1/CPU0 [LC] [SDR: Owner]
Info:         meta-data: [SUCCESS] Verification Successful.
Info:         /install/asr9k-scfclient-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-os-mbi-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-mpls-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-mcast-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-lc-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-fwdg-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-fpd-3.9.0.12I: [ERROR] Detected anomalies.
Info:         /install/asr9k-diags-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-base-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-admin-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:     0/4/CPU0 [LC] [SDR: Owner]
Info:         meta-data: [SUCCESS] Verification Successful.
Info:         /install/asr9k-scfclient-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-os-mbi-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-mpls-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-mcast-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-lc-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-fwdg-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-fpd-3.9.0.12I: [ERROR] Detected anomalies.
Info:         /install/asr9k-diags-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-base-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-admin-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:     0/RSP0/CPU0 [RP] [SDR: Owner]
Info:         meta-data: [SUCCESS] Verification Successful.
Info:         /install/asr9k-fpd-3.9.0.12I: [ERROR] Detected anomalies.
Info:         /install/asr9k-mpls-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-mgbl-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-mcast-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-k9sec-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-os-mbi-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-base-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-admin-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-fwdg-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-lc-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-rout-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-diags-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-scfclient-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:     Verification Summary:
Info:         0/6/CPU0: ERROR. Anomalies found.
Info:         0/1/CPU0: ERROR. Anomalies found.
```

```
Info:         0/4/CPU0: ERROR. Anomalies found.
Info:         0/RSP0/CPU0: ERROR. Anomalies found.
Info:         Anomalies found on the primary RP.
Info:         No standby RP is present.
Info:         Please contact your technical services representative to repair
Info:     the system.
Install operation 3 completed successfully at 13:21:29 DST Mon Jun 22 2009.
```

Verifying the Current System Status: Example

The following example shows how to prepare for system verification:

```
RP/0/RSP0/CPU0:router# show system verify start

  Storing initial router status ...
  done.
```

The following example shows output from running the **show system verify** command.

Note Although most of the output should display the status "OK," some processes may show other output, such as "Warning." This does not specifically indicate a problem. Contact your Cisco technical support representative for more information on the output of this command.

```
RP/0/RSP0/CPU0:router# show system verify

  Getting current router status ...
  System Verification Report
  ==========================
  - Verifying Memory Usage
  - Verified Memory Usage                                 : [OK]
  - Verifying CPU Usage
  - Verified CPU Usage                                    : [OK]

  - Verifying Blocked Processes
  - Verified Blocked Processes                            : [OK]
  - Verifying Aborted Processes
  - Verified Aborted Processes                            : [OK]
  - Verifying Crashed Processes
  - Verified Crashed Processes                            : [OK]

  - Verifying LC Status
  - Verified LC Status                                    : [OK]
  - Verifying QNET Status
  Unable to get current LC status info
  - Verified QNET Status                                  : [FAIL]

  - Verifying GSP Fabric Status
  - Verified GSP Fabric Status                            : [OK]
  - Verifying GSP Ethernet Status
    gsp WARNING messages for router
    Current set of gsp ping nodes does not match initial set of nodes
  - Verified GSP Ethernet Status                          : [WARNING]

  - Verifying POS interface Status
  - Verified POS interface Status                         : [OK]
```

```
- Verifying TenGigE interface Status
- Verified TenGigE interface Status                    : [OK]

- Verifying TCP statistics
- Verified TCP statistics                              : [OK]
- Verifying UDP statistics
  tcp_udp_raw WARNING messages for router
  UDP Packets sent has not increased during this period.
- Verified UDP statistics                              : [WARNING]
- Verifying RAW statistics
- Verified RAW statistics                              : [OK]

- Verifying RIB Status
- Verified RIB Status                                  : [OK]
- Verifying CEF Status
- Verified CEF Status                                  : [OK]
- Verifying CEF Consistency Status
- Verified CEF Consistency Status                      : [OK]
- Verifying BGP Status
- Verified BGP Status                                  : [OK]
- Verifying ISIS Status
- Verified ISIS Status                                 : [OK]
- Verifying OSPF Status
- Verified OSPF Status                                 : [OK]

- Verifying Syslog Messages
- Verified Syslog Messages                             : [OK]

System may not be stable. Please look into WARNING messages.
```

Verifying That the System Clock Is Correct: Example

The following example displays the current system clock setting:

```
RP/0/RSP0/CPU0:router# show clock

02:14:51.474 PST Wed Jan 28 2009
```

Upgrading to Cisco IOS XR Software Release 4.0

In Cisco IOS XR Software Release 4.0, the software packages were reorganized into functionally well-defined and independently-releasable packages. For this reason, when you upgrade from a software release prior to Release 4.0, you must perform the following procedure in order to synchronize all of the software packages according to the reorganized structure. General information regarding the the addition and activation of software packages is not covered in this procedure.

The main difference between the standard upgrade procedure and the procedure required to upgrade from Release 3.x to 4.x is that the later requires the addition of one additional software package, known as the *upgrade package* (asr9k-upgrade-p.pie).

Before you begin

Before performing this procedure, see the adding and activating software package procedures described in this module.

SUMMARY STEPS

1. **admin**
2. **install add tftp://** *hostname_or_ipaddress* **/** *directory-path* **/** *mandatory-bundle-pie*
3. **install add tftp://** *hostname_or_ipaddress* **/** *directory-path* **/ asr9k-upgrade-p.pie**
4. **install activate** *device***:***mandatory-bundle-pie device***:***upgrade-package*
5. **install deactivate** *device***:***upgrade-package*
6. (Optional) **install commit**
7. **install remove** *device***:***upgrade-package*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	Required: **admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode. • From administration EXEC mode, you can perform installation operations for the entire system. To enter administration EXEC mode, you must be logged in to the owner SDR and have root-system access privileges. • This command is not required. **Note** Some **show install** commands can be entered in EXEC mode on an SDR.
Step 2	**install add tftp://** *hostname_or_ipaddress* **/** *directory-path* **/** *mandatory-bundle-pie* **Example:** `RP/0/RSP0/CPU0:router(admin)# install add tftp://10.1.1.1/auto/tftpboot/usr/400/asr9k-mini-p.pie`	Unpacks the mandatory bundle PIE file from a network server and adds the package file to the boot device of the router. **Note** Refer to the standard procedure to add and activate packages to see other options of PIE file locations and a description of the various arguments for the **install add** command.
Step 3	**install add tftp://** *hostname_or_ipaddress* **/** *directory-path* **/ asr9k-upgrade-p.pie** **Example:** `RP/0/RSP0/CPU0:router(admin)# install add tftp://10.1.1.1/auto/tftpboot/usr/400/asr9k-upgrade-p.pie`	Unpacks the upgrade PIE file from a network server and adds the package file to the boot device of the router.
Step 4	**install activate** *device***:***mandatory-bundle-pie device***:***upgrade-package* **Example:** `RP/0/RSP0/CPU0:router(admin)# install activate`	Activates the package that was added to the router together with the upgrade package. **Note** The bundle of mandatory packages and the upgrade bundle are activated together to perform the successful upgrade from release 3.x to 4.x.

	Command or Action	Purpose
	disk0:asr9k-mini-p-4.0.0 disk0:asr9k-upgrade-p-4.0.0	
Step 5	**install deactivate** *device*:*upgrade-package* **Example:** RP/0/RSP0/CPU0:router(admin)# install deactivate disk0:asr9k-upgrade-p-4.0.0	Deactivates the upgrade package on the router . For specific information regarding the deactivation and removal of software packages, refer to the general procedure.
Step 6	(Optional) **install commit** **Example:** RP/0/RSP0/CPU0:router(admin)# install commit	Commits the current set of packages so that these packages are used if the router is restarted. Packages can be removed only if the deactivation operation is committed.
Step 7	Required: **install remove** *device*:*upgrade-package* **Example:** RP/0/RSP0/CPU0:router(admin)# install remove disk0:asr9k-upgrade-p-4.0.0	Removes the inactive upgrade package.

Example

The following example illustrates the upgrade operation:

```
RP/0/RSP0/CPU0:router(admin)# install add /tftp://223.255.254.254/auto/tftpboot/users/user/
asr9k-mini-p.pie

Fri Jul  9 03:53:11.052 UTCRP/0/RP1/CPU0:Jul  9 03:53:12.053 :
instdir[235]: %INSTALL-INSTMGR-6-INSTALL_OPERATION_STARTED :
Install operation 4 '(admin) install add
/tftp://223.255.254.254/auto/tftpboot/users/user/asr9k-mini-p.pie'
started by user 'lab'
Install operation 4 '(admin) install add
/tftp://223.255.254.254/auto/tftpboot/users/user/asr9k-mini-p.pie'
started by user 'lab' via CLI at 03:53:12 UTC Fri Jul 09 2010.
The install operation will continue asynchronously.
RP/0/RSP0/CPU0:router(admin)#
Info:    The following package is now available to be activated:
Info:    disk0:asr9k-mini-p-4.0.0
Info:    The package can be activated across the entire router.
Info:    RP/0/RP1/CPU0:Jul  9 04:32:26.152 : instdir[235]:
%INSTALL-INSTMGR-6-INSTALL_OPERATION_COMPLETED_SUCCESSFULLY :
Info:    Install operation 4 completed successfully
Info:    Install operation 4 completed successfully at 04:32:26 UTC Fri Jul 09 2010.
RP/0/RSP0/CPU0:router(admin)# install add /tftp://223.255.254.254/auto/tftpboot/users/user/
asr9k-mpls-p.pie

Fri Jul  9 05:07:52.237 UTCRP/0/RP1/CPU0:Jul  9 05:07:53.710 : instdir[235]:
%INSTALL-INSTMGR-6-INSTALL_OPERATION_STARTED :
Info:    Install operation 5 '(admin) install add
Info:    /tftp://223.255.254.254/auto/tftpboot/users/user/asr9k-mpls-p.pie'
Info:    started by user 'lab'
Info:    Install operation 5 '(admin) install add
Info:    /tftp://223.255.254.254/auto/tftpboot/users/user/asr9k-mpls-p.pie'
```

```
Info:    started by user 'lab' via CLI at 05:07:53 UTC Fri Jul 09 2010.
Info:    The install operation will continue asynchronously.
RP/0/RSP0/CPU0:router(admin)#
Info:    RP/0/RP1/CPU0:Jul  9 05:09:08.854 : instdir[235]:
%INSTALL-INSTMGR-6-INSTALL_OPERATION_COMPLETED_SUCCESSFULLY :
Install operation 5 completed successfully
Info:    The following package is now available to be activated:
Info:    disk0:asr9k-mpls-p-4.0.0
Info:    The package can be activated across the entire router.
Info:    Install operation 5 completed successfully at 05:09:08 UTC Fri Jul 09 2010.
RP/0/RSP0/CPU0:router# install add /tftp://223.255.254.254/auto/tftpboot/users/user/
asr9k-upgrade-p.pie

Fri Jul  9 05:10:31.133 UTCRP/0/RP1/CPU0:Jul  9 05:10:32.156 : instdir[235]:
%INSTALL-INSTMGR-6-INSTALL_OPERATION_STARTED :
Info:    Install operation 6 '(admin) install add
Info:    /tftp://223.255.254.254/auto/tftpboot/users/user/asr9k-upgrade-p.pie'
Info:    started by user 'lab'
Info:    Install operation 6 '(admin) install add
Info:    /tftp://223.255.254.254/auto/tftpboot/users/user/asr9k-upgrade-p.pie'
Info:    started by user 'lab' via CLI at 05:10:32 UTC Fri Jul 09 2010.
Info:    The install operation will continue asynchronously.
RP/0/RSP0/CPU0:router(admin)#RP/0/RP1/CPU0:
   Jul  9 05:11:55.634 : instdir[235]:
%INSTALL-INSTMGR-6-INSTALL_OPERATION_COMPLETED_SUCCESSFULLY :
Info:    Install operation 6 completed successfully
Info:    The following package is now available to be activated:
Info:    disk0:asr9k-upgrade-p-4.0.0
Info:    The package can be activated across the entire router.
Info:    Install operation 6 completed successfully at 05:11:55 UTC Fri Jul 09 2010.
RP/0/RSP0/CPU0:router(admin)# install activate disk0:asr9k-mini-p-4.0.0
disk0:asr9k-upgrade-p-4.0.0 disk0:asr9k-mpls-p-4.0.0

Fri Jul  9 05:23:23.150 UTC
Install operation 7 '(admin) install activate disk0:asr9k-mini-p-4.0.0
Info:    disk0:asr9k-upgrade-p-4.0.0 disk0:asr9k-mpls-p-4.0.0'
Info:    started by user 'lab'RP/0/RP1/CPU0:Jul  9 05:23:24.161 : instdir[235]:
%INSTALL-INSTMGR-6-INSTALL_OPERATION_STARTED :
Info:    Install operation 7 '(admin) install activate disk0:asr9k-mini-p-4.0.0
Info:    disk0:asr9k-upgrade-p-4.0.0 disk0:asr9k-mpls-p-4.0.0'
Info:    started by user 'lab' via CLI at 05:23:24 UTC Fri Jul 09 2010.\ 1% complete:
Info:    The operation can still be aborted (ctrl-c for options)
Info:    This operation will reload the following nodes in parallel:
Info:    0/RP1/CPU0 (HRP) (SDR: Owner)
Info:    0/SM0/SP (Fabric-SP) (Admin Resource)Proceed with this install operation (y/n)?
[y]|
Info:    1% complete: The operation can still be aborted (ctrl-c for options)
Info:    Install Method: Parallel Reload/ 1% complete: The operation can still be aborted
(ctrl-c for options)
Info:    The install operation will continue asynchronously.
RP/0/RSP0/CPU0:router(admin)#SP/0/SM0/SP:
   Jul  9 05:36:41.152 : insthelper[62]: %INSTALL-INSTHELPER-6-RELOAD_NODE_INFO :
Info:    As part of install operation 7 this node (0/SM0/SP) will now reload.
Info:    The changes made to software configurations will not be persistent
Info:    across system reloads. Use the command '(admin) install commit' to
Info:    make changes persistent.
Info:    Please verify that the system is consistent following the software
RP/0/RP1/CPU0:Jul  9 05:36:43.962 : instdir[235]:
%INSTALL-INSTMGR-6-INSTALL_OPERATION_COMPLETED_SUCCESSFULLY :
Info:    Install operation 7 completed successfully
Info:    change using the following commands:
Info:    show system verify
Info:    install verify packages
Info:    Install operation 7 completed successfully at 05:36:43 UTC Fri Jul 09 2010.
```

```
rebooting .........................Initializing DDR SDRAM...found 4096 MB
Initializing ECC on bank 0Initializing ECC on bank 1
Initializing ECC on bank 2
Initializing ECC on bank 3
Turning off data cache, using DDR for first time
Initializing NVRAM...Testing a portion of DDR SDRAM ...done
Reading ID EEPROMs ............................
Initializing SQUID ...
Initializing PCI ...PCI0 device[1]: Vendor ID 0x10eePCI0 device[1]: Device ID 0x300ePCI1
device[1]:
Device ID 0x1100PCI1 device[1]: Vendor ID 0x1013PCI1 device[2]: Device ID 0x680PCI1 device[2]:

Vendor ID 0x1095PCI1 device[3]: Device ID 0x5618PCI1 device[3]: Vendor ID 0x14e4Configuring
 MPPs ...
Configuring PCMCIA slots ...System Bootstrap, Version 1.53(20090311:225342) [CRS-1 ROMMON],

Copyright (c) 1994-2009 by Cisco Systems, Inc.
Acquiring backplane mastership ... successful
Preparing for fan initialization.............. ready
Setting fan speed to 4000 RPMs  successfulReading backplane EEPROM ...
Released backplane mastership ...Board type is 0x100002 (1048578)
Switch 0 initialized
Switch 0 Port fe1: link up (100Mb Full Duplex Copper)
Enabling watchdogG4(7457-NonSMP-MV64360 Rev 3) platform with 4096 MB of main memory....

CARD_RACK_NUMBER: 0    CARD_SLOT_NUMBER: 1        CPU_INSTANCE: 1
RACK_SERIAL_NUMBER: TBC08052402
MBI Validation starts ... using Control Plane Ethernet.
DEBUG : Driving up signal strength for Intel LXT971
Our MAC address is 0005.9a3e.89da
Interface link changed state to UP.
Interface link state up.
MBI validation sending request.
HIT CTRL-C to abort
MBI validation sending request.
HIT CTRL-C to abort
MBI validation sending request.
HIT CTRL-C to abort
MBI validation sending request.
HIT CTRL-C to abort
MBI validation sending request.
HIT CTRL-C to abort
No MBI confirmation received from dSCboot: booting from
bootflash:disk0/asr9k-os-mbi-4.0.0/mbiasr9k-rp.vm
.................................................................................
################################################################################

Restricted Rights LegendUse, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph(c) of the Commercial Computer Software
 - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph(c) (1) (ii) of the Rights in Technical
 Data and Computer
Software clause at DFARS sec. 252.227-7013.
cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706
Cisco IOS XR Software for the Cisco XR Router, Version 4.0.0 Copyright (c) 2010 by Cisco
Systems, Inc.
Jul 09 05:39:21.334 : Install (Node Preparation): Booting with software activated by previous
 install
operation,errno=2
RP/0/RP1/CPU0Jul  9 05:44:45.941: syslogd_helper: [89]: dsc_event_handler: Got SysMgr dSC
event : 1
RP/0/RP1/CPU0:Jul  9 05:45:11.354 : shelfmgr[306]: %PLATFORM-SHELFMGR-3-POWERDOWN_RESET :
```

```
Node 0/2/SP is powered off due to admin power off request ios con0/RP1/CPU0 is now available
Press RETURN to get started.
RP/0/RP1/CPU0:Jul  9 05:45:27.453 : instdir[216]:
%INSTALL-INSTMGR-4-ACTIVE_SOFTWARE_COMMITTED_INFO :
The currently active software is not committed. If the system reboots then the committed
software will be used.
Use 'install commit' to commit the active software. SYSTEM CONFIGURATION IN PROCESS
The startup configuration for this device is presently loading.
This may take a few minutes. You will be notified upon completion.
Please do not attempt to reconfigure the device until this process is complete.
User Access VerificationUsername: labPassword:
RP/0/RSP0/CPU0:router# admin
Fri Jul  9 05:45:55.941 UTC
RP/0/RSP0/CPU0:router(admin)# show platform

Fri Jul  9 05:45:59.805 UTCNode            Type                 PLIM             State
       Config State
-------------------------------------------------------------------------------------------
0/2/SP          MSC(SP)              N/A              UNPOWERED       NPWR,NSHUT,MON
0/RP1/CPU0      RP(Active)           N/A              IOS XR RUN      PWR,NSHUT,MON
0/SM0/SP        FC-40G/S(SP)         N/A              MBI-RUNNING     PWR,NSHUT,MON
0/SM1/*         UNKNOWN              N/A              PRESENT         PWR,NSHUT,MON

RP/0/RP1/CPU0:ios(admin)#
RP/0/RP1/CPU0:Jul  9 05:46:08.411 : instdir_lr[217]:
%INSTALL-INSTMGR-4-ACTIVE_SOFTWARE_COMMITTED_INFO :
The currently active software is not committed. If the system reboots then the committed
software will be used.
Use 'install commit' to commit the active software.
RP/0/RP1/CPU0:Jul  9 05:50:40.918 : placed[283]: LR-PLANE-READY DECLARATIONSYSTEM
CONFIGURATION COMPLETED
RP/0/RP1/CPU0:Jul  9 05:50:57.293 : ifmgr[213]: %PKT_INFRA-LINK-3-UPDOWN :
Interface MgmtEth0/RP1/CPU0/0, changed state to Down
RP/0/RP1/CPU0:Jul  9 05:50:57.313 : ifmgr[213]: %PKT_INFRA-LINK-3-UPDOWN :
Interface MgmtEth0/RP1/CPU0/0, changed state to Up
RP/0/RSP0/CPU0:router(admin)# show platform

Fri Jul  9 05:59:36.266 UTC
Node            Type                 PLIM             State           Config State
-------------------------------------------------------------------------------------------
0/2/SP          MSC(SP)              N/A              UNPOWERED       NPWR,NSHUT,MON
0/RP1/CPU0      RP(Active)           N/A              IOS XR RUN      PWR,NSHUT,MON
0/SM0/SP        FC-40G/S(SP)         N/A              IOS XR RUN      PWR,NSHUT,MON
0/SM1/*         UNKNOWN              N/A              PRESENT         PWR,NSHUT,MON

RP/0/RSP0/CPU0:router(admin)# install commit

Fri Jul  9 05:59:41.851 UTC
Install operation 8 '(admin) install commit' started by user 'lab' via CLI at
05:59:43 UTC Fri Jul 09 2010./
20% complete: The operation can no longer be aborted (ctrl-c for options)-
20% complete: The operation can no longer be aborted (ctrl-c for options)\
100% complete:
The operation can no longer be aborted (ctrl-c for options)
RP/0/RP1/CPU0:Jul  9 05:59:46.402 : instdir[216]:
%INSTALL-INSTMGR-4-ACTIVE_SOFTWARE_COMMITTED_INFO :
The currently active software is now the same as the committed software.
Install operation 8 completed successfully at 05:59:46 UTC Fri Jul 09 2010.
RP/0/RSP0/CPU0:router(admin)# install deactivate disk0:
asr9k-upgrade-p-4.0.0

Fri Jul  9 05:59:58.082 UTC
Install operation 9 '(admin) install deactivate disk0:asr9k-upgrade-p-4.0.0'started
   by user 'lab' via CLI at 05:59:59 UTC
```

```
Fri Jul 09 2010.
1% complete: The operation can still be aborted (ctrl-c for options)-
1% complete: The operation can still be aborted (ctrl-c for options)
Info:     Install Method: Parallel Process Restart\
1% complete: The operation can still be aborted (ctrl-c for options)
The install operation will continue asynchronously.
RP/0/RSP0/CPU0:router(admin)#
Info:     The changes made to software configurations will not be persistent
Info:     across system reloads. Use the command '(admin) install commit' to
Info:     make changes persistent.
Info:     Please verify that the system is consistent following the software
Info:     change using the following commands:
Info:         show system verify
Info:         install verify packages
RP/0/RP1/CPU0:Jul  9 06:01:45.662 : instdir[216]:
%INSTALL-INSTMGR-4-ACTIVE_SOFTWARE_COMMITTED_INFO :
The currently active software is not committed. If the system reboots then the committed
software will be used.
Use 'install commit' to commit the active software.
Install operation 9 completed successfully at 06:01:45 UTC Fri Jul 09 2010.
RP/0/RSP0/CPU0:router(admin)# install commit

Fri Jul  9 06:01:53.583 UTC
Install operation 10 '(admin) install commit' started by user 'lab' via CLI at06:01:54 UTC
 Fri Jul 09 2010./
20% complete: The operation can no longer be aborted (ctrl-c for options)-
20% complete: The operation can no longer be aborted (ctrl-c for options)\
100% complete: The operation can no longer be aborted (ctrl-c for options)
RP/0/RP1/CPU0:Jul  9 06:01:57.807 : instdir[216]:
%INSTALL-INSTMGR-4-ACTIVE_SOFTWARE_COMMITTED_INFO :
The currently active software is now the same as the committed software.
Install operation 10 completed successfully at 06:01:57 UTC Fri Jul 09 2010.
RP/0/RSP0/CPU0:router(admin)#
RP/0/RSP0/CPU0:router(admin)#
RP/0/RSP0/CPU0:router(admin)# install remove disk0:
asr9k-upgrade-p-4.0.0

Fri Jul  9 06:04:57.676 UTC
Install operation 11 '(admin) install remove disk0:asr9k-upgrade-p-4.0.0'started
   by user 'lab' via CLI at 06:04:58 UTC
Fri Jul 09 2010./
1% complete: The operation can no longer be aborted (ctrl-c for options)
Info:     This operation will remove the following packages:
Info:         disk0:asr9k-fpd-4.0.0
Info:         disk0:asr9k-doc-4.0.0
Info:         disk0:asr9k-k9sec-4.0.0
Info:         disk0:asr9k-sbc-4.0.0
Info:         disk0:asr9k-diags-4.0.0
Info:         disk0:asr9k-mgbl-4.0.0
Info:         disk0:asr9k-mcast-4.0.0
Info:         disk0:asr9k-mpls-4.0.0
Info:         disk0:asr9k-rout-4.0.0
Info:         disk0:asr9k-fwdg-4.0.0
Info:         disk0:asr9k-lc-4.0.0
Info:         disk0:asr9k-admin-4.0.0
Info:         disk0:asr9k-upgrade-p-4.0.0-
1% complete: The operation can no longer be aborted (ctrl-c for options)
Info:     After this install remove the following install rollback point will
Info:     no longer be reachable, as the required packages will not be present:
Info:         7\
1% complete: The operation can no longer be aborted (ctrl-c for options)
Proceed with removing these packages? [confirm]|
1% complete: The operation can no longer be aborted (ctrl-c for options)
The install operation will continue asynchronously.
```

```
RP/0/RSP0/CPU0:router(admin)#SP/0/SM0/SP:Jul
    9 06:05:03.902 : envmon[117]: %PLATFORM-ENVMON-4-ALARM : MINOR_HI alarm
cleared by host__temp__Inlet0
Install operation 11 completed successfully at 06:05:33 UTC
Fri Jul 09 2010.
RP/0/RSP0/CPU0:router(admin)#
RP/0/RSP0/CPU0:router(admin)# show install act
Fri Jul  9 06:08:11.372 UTC
Secure Domain Router: Owner  Node 0/RP1/CPU0 [HRP] [SDR: Owner]
Boot Device: disk0:    Boot Image: /disk0/asr9k-os-mbi-4.0.0/mbiasr9k-rp.vm
Active Packages:       disk0:asr9k-mpls-p-4.0.0      disk0:asr9k-mini-p-4.0.0
Admin Resources:  Node 0/SM0/SP [Fabric-SP] [Admin Resource]
Boot Device: bootflash:    Boot Image: /disk0/asr9k-os-mbi-4.0.0/sp/mbiasr9k-sp.vm
Active Packages:       disk0:asr9k-mini-p-4.0.0
RP/0/RSP0/CPU0:router(admin)#
```

Related Topics

Activation and Deactivation Prerequisites, on page 153
Adding and activating packages, on page 157
Deactivating and Removing Cisco IOS XR Software Packages, on page 168

Rolling Back to a Previous Software Set

Cisco IOS XR software allows you to roll back one or more SDRs to a previous committed or uncommitted software set. Use the **show install rollback ?** command to view the available rollback points and use the **install rollback to** command to roll back the SDR to a previous software set. You can also use the **install rollback to committed** command to roll back to the most recent committed software set.

Note Rollback operations can be performed by running the command in administration EXEC or EXEC mode.

Note If type 8,9, or 10 is the secret key configured, then before downgrading to 6.6.3 and earlier versions, perform either of the following methods:

- Type a combination of secret type and encrypted key instead of plain text for the password. Example:

  ```
  username root
  group root-lr
  group cisco-support
  secret 10
  $6$Mwaqg/jdBPOn4g/.$PrJP2KjsCbL6bZqmYOej5Ay67S/sSWJNlkiYhCTc/B/35E1kJBqffmBtn.ddQEH0O2CU7V.ZEMmqIg7uE8cfz0
  ```

  This is because 6.6.3 and earlier versions do not support type 8,9, or 10 key type.

- Ensure that there are secret type 5 users on the system.

Displaying Rollback Points

A rollback point is created every time a software package is activated, deactivated, or committed. Use the **show install rollback ?** command to display the eligible rollback points.

```
RP/0/RSP0/CPU0:router# admin
```

```
RP/0/RSP0/CPU0:router(admin)# show install rollback ?

    0  ID of the rollback point to show package information for
    2  ID of the rollback point to show package information for
```

In this example, the rollback points are 0 and 2. The rollback point with the highest number is the current software point. For example, if the last installation operation was operation 3 (activating the MPLS package) then the highest rollback point is 3, which is the same as the current software (MPLS package activated).

To easily identify specific rollback points, you can assign a label or description to a rollback point using the **install label** command.

You can enter the command in either administration EXEC mode or EXEC mode.

Displaying the Active Packages Associated with a Rollback Point

To display the active packages associated with a rollback point, use the **show install rollback** command with the *point-id* argument. This command displays the packages that are active if you roll back one or more SDRs to that installation point. For example, the **show install rollback 2** command displays the packages that are active if you roll back to rollback point 2.

```
RP/0/RSP0/CPU0:router(admin)# show install rollback 0

Tue Jun 23 06:25:06.493 PST
ID: 0, Label:
Timestamp: 23:11:20 UTC Sat Oct 28 2000

Secure Domain Router: Owner

  Node 0/RSP0/CPU0 [RP] [SDR: Owner]
    Boot Device: disk0:
    Boot Image: /disk0/asr9k-os-mbi-3.9.0.12I/mbiasr9k-rp.vm
    Rollback Packages:
      disk0:comp-asr9k-mini-3.9.0.12I

  Node 0/1/CPU0 [LC] [SDR: Owner]
    Boot Device: mem:
    Boot Image: /disk0/asr9k-os-mbi-3.9.0.12I/lc/mbiasr9k-lc.vm
    Rollback Packages:
      disk0:comp-asr9k-mini-3.9.0.12I

  Node 0/4/CPU0 [LC] [SDR: Owner]
    Boot Device: mem:
    Boot Image: /disk0/asr9k-os-mbi-3.9.0.12I/lc/mbiasr9k-lc.vm
    Rollback Packages:
      disk0:comp-asr9k-mini-3.9.0.12I

  Node 0/6/CPU0 [LC] [SDR: Owner]
    Boot Device: mem:
    Boot Image: /disk0/asr9k-os-mbi-3.9.0.12I/lc/mbiasr9k-lc.vm
    Rollback Packages:
      disk0:comp-asr9k-mini-3.9.0.12I
```

You can enter the command in either administration EXEC mode or EXEC mode.

Note For more information on the command options, see the *Software Package Management Commands on Cisco IOS XR Software* module of *System Management Command Reference for Cisco ASR 9000 Series Routers*.

Rolling Back to a Specific Rollback Point

You can roll back to a specific rollback point, including a noncommitted software set:

- If you roll back to the most recent noncommitted rollback point (with the highest number), you do not need to reload the router.
- You can repeat the rollback process one rollback point at a time without reloading if you always choose the most recent rollback point.
- If you choose a rollback point that is older than the most recent point, the impacted nodes reload, interrupting data traffic on those nodes. Before the reload occurs, you are prompted to confirm the install rollback operation.

In the following example, the system is rolled back to noncommitted rollback point 8:

```
RP/0/RSP0/CPU0:router(admin)# install rollback to 8

  Install operation 10 'install rollback to 8' started by user 'cisco' at 07:49:26
  UTC Mon Nov 14 2009.
  The install operation will continue asynchronously.
  Info:     The changes made to software configurations will not be persistent
  Info:     across system reloads. Use the command 'admin install commit' to make
  Info:     changes persistent.
  Info:     Please verify that the system is consistent following the software
  Info:     change using the following commands:
  Info:         show system verify
  Info:         install verify packages

  The currently active software is the same as the committed software.

  Install operation 10 completed successfully at 07:51:24 UTC Mon Nov 14 2009.
```

Rolling Back to the Last Committed Package Set

Use the **install rollback to committed** command to roll back to the last committed package set.

In the following example, the owner SDR is rolled back to the last committed package set:

```
RP/0/RSP0/CPU0:router(admin)# install rollback to committed

  Install operation 27 'install rollback to committed' started by user 'lab' at
  16:41:38 UTC Sat Nov 19 2009.
  Info:     The rollback to committed software will require a reload of impacted
  Info:     nodes because it is over multiple activation & deactivation
  Info:     operations.
  Info:     This operation will reload the following node:
  Info:         0/RP1/CPU0 (RP) (SDR: Owner)
  Info:     This operation will reload all RPs in the Owner SDR, and thereby
  Info:     indirectly cause every node in the router to reload.

  Proceed with this install operation? [confirm]

  Updating Commit Database.  Please wait...[OK]
```

```
Info:     The changes made to software configurations will not be persistent
Info:     across system reloads. Use the command 'admin install commit' to make
Info:     changes persistent.
Info:     Please verify that the system is consistent following the software
Info:     change using the following commands:
Info:         show system verify
Info:         install verify packages
Install operation 27 completed successfully at 16:42:23 UTC Sat Nov 19 2009.
```

You can enter the command in either administration EXEC mode or EXEC mode.

Resetting Router to Factory Settings

The logical volumes and ROMMON variables of CPU boards on a router can be reset to factory settings using zapdisk feature. After enabling the zapdisk feature on the router, the CPU boards are reset to factory settings in the next reimage of the boards. During the reimage process, all logical volumes of the CPU boards including the files saved in harddisk: are cleaned up, and ROMMON variables of the CPU boards are reset to factory settings.

Procedure

Step 1 **admin**

Example:

```
Router# admin
```

Enters the System Admin EXEC mode.

Step 2 **zapdisk set**

Example:

```
sysadmin-vm:0_RP0# zapdisk set
Fri Jul  21 22:32:29.242 UTC
result Zapdisk set command success
```

Enables zapdisk feature.

Note
To disable the zapdisk feature, run the **zapdisk unset** command:

```
sysadmin-vm:0_RP0# zapdisk unset
Fri Jul  21 22:32:29.242 UTC
result Zapdisk unset command success
```

Step 3 **run**

Example:

```
sysadmin-vm:0_RP0# run
 [sysadmin-vm:0_RP0:~]$/opt/cisco/calvados/bin/nvram_dump -a
 PS1=rommon ! >
 ZAPDISK_CARD=1
```

Verifies status of zapdisk feature on the CPU board. `ZAPDISK_CARD=1` indicates that zapdisk feature is enabled; `ZAPDISK_CARD=0` indicates that zapdisk feature is disabled.

Red Hat Package Manager

RPM (Red Hat Package Manager) is a widely used open-source package management system for installing, updating, uninstalling, verifying, and querying software packages on Linux distributions such as Red Hat Enterprise Linux (RHEL), CentOS, and so on.

Cisco uses the RPM package management system to package required and optional files for installing and upgrading the Cisco IOS XR software.

Cisco IOS-XR 64-bit package file names and versions

Cisco IOS XR 64-bit software packages can be an ISO file, RPM file, a composite RPM SMU package or a single RPM SMU package. A *composite package* is a file that contains multiple packages.

Note Hyphens in the filename are part of the filename.

This table shows the filenames for ISO and RPM file types.

Table 11: Package file names and versions

Software Delivery Type	Filename	Example
ISO package	*platform-composite_name-major.minor.maintenance.iso*	`asr9k-mini-x64-7.3.2.iso`
RPM package	*platform-package_type-major.minor.maintenance.opt_files-release.px.rpm*	`asr9k-mpls-x64-2.0.0.0-r732.x86_64.rpm`
Composite SMU	*platform-package_type-major.minor.maintenance.opt_files-release.ddts.px.rpm*	`asr9k-fwding-64-1.0.0.1-r2412.CSCwj98521.x86_64.rpm`
Single package SMU (RPM)	*platform-package_type-major.minor.maintenance.opt_files-release.ddts.px.rpm*	`asr9k-fwding-64-1.0.0.1-r2412.CSCwj98521.x86_64.rpm`

Rolling back to a previous installation operation

Cisco IOS XR 64-bit software

The **install rollback** command is not supported on Cisco IOS XR 64-bit software.

Instead, you can perform a roll back in one of the following ways:

- If you have not committed the changes, the software rolls back to the previous version upon router reload.
- After committing the changes, execute the **install add** and **install activate** commands using the package version to which you want to roll back the software.

Filename components description

This table describes the filename components for all packages.

Table 12: Composite and single package filename components

Component	Description
platform	Identifies the platform for which the software package is designed. The platform designation is “asr9k.”
composite_name	Identifies a specific composite package. The only composite PIE file at this time is named “mini” and includes all packages described in the Cisco IOS XR Unicast Routing Core Bundle.
package_type	Identifies the type of package the file supports (*package_type* applies only to single-package PIEsfiles). Package types include: • mcast—Multicast package • mgbl—Manageability package • mpls—MPLS package • k9sec—Security package • diags—Diagnostics package • fpd—Field-programmable device package • doc—Documentation package
major	Identifies the major release of this package. • A major release occurs when there is a major architectural change to the product (for example, a major new capability is introduced). • All packages operating on the router must be at the same major release level. • A major release is the least frequent release and may require a router reboot.

minor	Identifies the minor release of this package.
	• A minor release contains one or more of the following:
	  • New features
	  • Bug fixes
	• The minor release version does not have to be identical for all software packages operating on the router, but the operating packages must be certified by Cisco as compatible with each other.
	• A minor release may require a router reboot.

Component	Description
maintenance	Identifies the maintenance release of this package. • A maintenance release contains a collection of bug fixes for a package. • The maintenance release version does not have to be identical for all software packages operating on the router, but the major and minor versions of the maintenance release must match those of the package being updated. • A maintenance release does not usually require a router reboot.
ddts	SMUs only. Identifies a DDTS[1] number that describes the problem this SMU addresses. DDTS is the method used to track known bugs and the resolutions or workarounds for those issues.

[1] distributed defect tracking system

Copying the file to a local storage device or network server

To add an optional package or upgrade or downgrade a package, you must copy the appropriate file to a local storage device or to a network file server to which the router has access.

If you need to store files on the router, we recommended storing files on the hard disk. Flash disk0: serves as the boot device for packages that have been added or activated on the system. Flash disk1: is used as a backup for disk0:.

Tip Before copying files to a local storage device, use the **dir** command to check to see if the required files are already on the device.

Software Maintenance Upgrades

A software maintenance upgrade (SMU) is a PIE or RPM file that contains fixes for a specific defect. A composite SMU is a PIE or RPM file that contains SMUs for more than one package. SMUs are added and activated using the same procedures as other PIE or RPM files. SMUs are created to respond to immediate issues and do not include new features. Typically, SMUs do not have a large impact on router operations. SMU versions are synchronized to the package major, minor, and maintenance versions they upgrade.

The affect of an SMU depends on its type:

- Process Restart SMU—Causes a process or group of processes to restart on activation.
- Reload SMU—Causes a parallel reload (of RPs and line cards).

SMUs are not an alternative to maintenance releases. They provide quick resolution of immediate issues. All bugs fixed by SMUs are integrated into the maintenance releases. For information on available SMUs, contact Cisco Technical Support, as described in *Obtaining Technical Assistance* in the monthly What's New in Cisco Product Documentation.

Note Activating a software maintenance upgrade (SMU) does not cause any earlier SMUs, or the package to which the SMU applies, to be automatically deactivated.

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.

Deactivation of fully superseded SMUs

Cisco IOS XR Software will accumulate a set of Software Maintenance Upgrades (SMUs) over time, where an older SMU gets superceded by the latest SMU. For example, if SMU A was initially delivered to you, and subsequently, as a result of a bug resolution, SMU B was delivered, then SMU A becomes the subset of SMU B and SMU A is superceded by SMU B. In this case, SMU A is redundant and can be deactivated to clean up the software package.

To deactivate all the fully superseded SMUs, use the **install deactivate superseded** command in the admin mode.

```
RP/0/RSP0/CPU0: router(admin) # install deactivate superseded
```

To display the details of the SMUs that are superseded, use the **show install superseded** command in the EXEC mode.

```
RP/0/RSP0/CPU0: router # show install superseded
Thu Feb 3 17:37:20.379 UTC
disk0:asr9k-px-4.3.0.CSCud93518-1.0.0 is fully superseded by
disk0:asr9k-px-4.3.0.CSCue23747-1.0.0
```

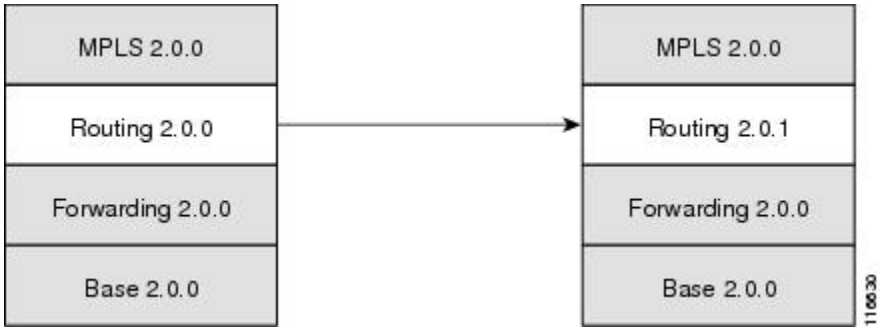
Upgrading Packages

To upgrade a package that is currently active on your router, add and activate a newer version of the same package (see Figure 12: Example of a Maintenance Release Package Upgrade, on page 149). The older version of the software package is deactivated automatically. These actions are permitted only after the package compatibility checks and API version compatibility checks have been passed.

Deactivated packages are not removed from the router. To remove inactive package files, use the **install remove** command.

Caution Upgrading or downgrading a software package can cause a process to restart or a new process to start. Use the **test** option to preview the impact of the package activation.

Figure 12: Example of a Maintenance Release Package Upgrade



Related Topics

Deactivating and Removing Cisco IOS XR Software Packages, on page 168

Downgrading Packages

To downgrade a software package, activate an older version on one or more cards for which that package is already active. The newer version of the same software package is deactivated automatically. These actions are performed only after the package compatibility checks and API version compatibility checks have been passed.

Deactivated packages are not removed from the router. To remove inactive package files, use the **install remove** command. See the *Related Topics* section for links to more information.

Note If type 8,9, or 10 is the secret key configured, then before downgrading to 6.6.3 and earlier versions, perform either of the following methods:

- Type a combination of secret type and encrypted key instead of plain text for the password. Example:

```
username root
group root-lr
group cisco-support
secret 10
$6$Mwaqg/jdBPOn4g/.$PrJP2KjsCbL6bZqmYOej5Ay67S/sSWJNlkiYhCTc/B/35E1kJBqffmBtn.ddQEH0O2CU7V.ZEMmqIg7uE8cfz0
```

  This is because 6.6.3 and earlier versions do not support type 8,9, or 10 key type.

- Ensure that there are secret type 5 users on the system.

Related Topics

Deactivating and Removing Cisco IOS XR Software Packages, on page 168

Impact of Package Version Changes

Each package version change has a different impact on the operation of the router, depending on the type of package and whether the upgrade is for a major, minor, or maintenance release. The following resources can provide more information on the impact of a package version change:

- See *Related Topics* for more information on the typical impact for major, minor, and maintenance releases.

- For specific information about the impact of an upgrade, consult the release notes for the package release, and test the impact of the package activation by adding the test option to the **install activate** command.
- The Cisco IOS XR Software Selector tool also contains information on package version compatibility.

Related Topics

PIE Filenames and Version Numbers, on page 119
Obtaining and Placing Cisco IOS XR Software, on page 154

Impact of Package Activation and Deactivation

Activation or deactivation of a package can have an immediate impact on the system. The system can be affected in the following ways:

- When a new package is activated, any new CLI commands for the package are added to the router. The router need not be restarted or reloaded.
- When a package is deactivated, the commands associated with the features being deactivated are removed from the router. The commands are no longer available to the user.
- During a software package deactivation, upgrade, or downgrade, any incompatible configurations are removed from the running configuration of the router, and saved to a file. Messages for incompatible configurations are displayed. Incompatible configurations are those configurations that are not supported by the new version of the software package.

Note You must address any issues that result from the revised configuration and reapply the configuration, if necessary.

- New processes may be started.
- Running processes may be stopped or restarted.
- All processes in the cards may be restarted. Restarting processes in the cards is equivalent to a soft reset.
- The cards may reload.
- No impact: no processes in the card may be affected.

Tip When activating and deactivating packages, use the **test** option to test the effects of a command without impacting the running system. After the activation or deactivation process completes, enter the **show install log** command to display the process results.

Delaying the return of the CLI prompt

By default, the CLI prompt is returned to the screen before the installation operation is complete, which allows you to enter other commands that are not installation commands. If additional installation requests are attempted before the first operation is complete, they are not run.

To delay the return of the CLI prompt until an installation operation is complete, enter the **install** command with the **synchronous** keyword. For example:

```
install add disk1:package synchronous
install activate disk0:package synchronous
```

To determine if an **install** command is currently running, enter the **show install request** command.

Displaying Installation Log Information

The install log provides information on the history of the installation operations. Each time an installation operation is run, a number is assigned to that operation.

- Use the **show install log** command to display information about both successful and failed installation operations.
- The **show install log** command with no arguments displays a summary of all installation operations. Specify the *request-id* argument to display information specific to an operation. Use the **detail** or **verbose** keywords to display details for specific operation.
- Use the **detail** or **verbose** keywords to display detailed information, including file changes, nodes that could be reloaded, impact to processes, and impact to Dynamic Link Libraries (DLLs).

Tip By default, the install log stores up to 50 entries. Use the **clear install log-history** command to reset the number of entries to any value from 0 to 255.

Examples

Displaying install log Entries: Example

The following example displays information for the install requests. Use the **verbose** keyword to display detailed information, including files changes, impact to processes, and impact to DLLs.

```
RP/0/RSP0/CPU0:router(admin)# show install log verbose

Install operation 1 started by user 'labuser' at 17:48:51 UTC Sat Jun 03 2009.
install add /disk1:asr9k-diags-p.pie-PD34-06.06.07
/disk1:asr9k-k9sec-p.pie-PD34-06.06.07 /disk1:asr9k-mcast-p.pie-PD34-06.06.07
/disk1:asr9k-mgbl-p.pie-PD34-06.06.07 /disk1:asr9k-mpls-p.pie-PD34-06.06.07
Install operation 1 completed successfully at 17:51:32 UTC Sat Jun 03 2009.

Install logs:
 Install operation 1 'install add /disk1:asr9k-diags-p.pie-PD34-06.06.07
 /disk1:asr9k-k9sec-p.pie-PD34-06.06.07 /disk1:asr9k-mcast-p.pie-PD34-06.06.07
 /disk1:asr9k-mgbl-p.pie-PD34-06.06.07 /disk1:asr9k-mpls-p.pie-PD34-06.06.07'
 started by user 'labuser' at 17:48:51 UTC Sat Jun 03 2009.
 Info:     The following packages are now available to be activated:
    Info:
    Info:         disk0:asr9k-diags-3.7.2.1I
    Info:         disk0:asr9k-k9sec-3.7.2.1I
    Info:         disk0:asr9k-mcast-3.7.2.1I
    Info:         disk0:asr9k-mgbl-3.7.2.1I
    Info:         disk0:asr9k-mpls-3.7.2.1I
    Info:
  Install operation 1 completed successfully at 17:51:32 UTC Sat Jun 03 2009.
  Install operation 2 started by user 'labuser' at 18:06:32 UTC Sat Jun 03 2009.
  install activate disk0:asr9k-diags-3.7.2.1I disk0:asr94k-k9sec-3.7.2.1I
```

```
disk0:asr9k-mcast-3.7.2.1I disk0:asr9k-mgbl-3.7.2.1I disk0:asr9k-mpls-3.7.2.1I
Install operation 2 completed successfully at 18:07:48 UTC Sat Jun 03 2009.
  Summary:
    Install method: parallel
    Summary of changes on nodes 0/1/SP, 0/6/SP, 0/SM0/SP, 0/SM1/SP,
    0/SM2/SP,0/SM3/SP:
        Activated:    asr9k-diags-3.7.2.1I
        No processes affected

    Summary of changes on nodes 0/1/CPU0, 0/6/CPU0:
        Activated:    asr9k-diags-3.7.2.1I
                      asr9k-mcast-3.7.2.1I
                      asr9k-mpls-3.7.2.1I
       1 asr9k-mpls processes affected (0 updated, 1 added, 0 removed, 0 impacted)
       2 asr9k-mcast processes affected (0 updated, 2 added, 0 removed, 0 impacted)

    Summary of changes on nodes 0/RP0/CPU0, 0/RP1/CPU0:
        Activated:    asr9k-diags-3.7.2.1I
                      asr9k-k9sec-3.7.2.1I
                      asr9k-mcast-3.7.2.1I
                      asr9k-mgbl-3.7.2.1I
                      asr9k-mpls-3.7.2.1I
       6 asr9k-mgbl processes affected (0 updated, 6 added, 0 removed, 0 impacted)
       8 asr9k-mpls processes affected (0 updated, 8 added, 0 removed, 0 impacted)
       7 asr9k-k9sec processes affected (0 updated, 7 added, 0 removed, 0 impacted)
      14 asr9k-mcast processes affected (0 updated, 14 added, 0 removed, 0 impacted)

Install logs:
  Install operation 2 'install activate disk0:asr9k-diags-3.7.2.1I
    disk0:asr9k-k9sec-3.7.2.1I disk0:asr9k-mcast-3.7.2.1I disk0:asr9k-mgbl-3.7.2.1I
    disk0:asr9k-mpls-3.7.2.1I' started by user 'labuser' at
    18:06:32 UTC Sat Jun 03 2009.
    Info:     The changes made to software configurations will not be
    Info:     persistent across system reloads. Use the command 'admin install
    Info:     commit' to make changes persistent.
    Info:     Please verify that the system is consistent following the
    Info:     software change using the following commands:
    Info:         show system verify
 --More--
```

The following example displays information for a specific install request. Use the **detail** keyword to display additional information, including impact to processes and nodes impacted.

```
RP/0/RSP0/CPU0:router(admin)# show install log 2 detail

  Install operation 2 started by user 'labuser' at 18:06:32 UTC Sat Jun 03 2009.
  install activate disk0:asr9k-diags-3.7.2.1I disk0:asr9k-k9sec-3.7.2.1I
  disk0:asr9k-mcast-3.7.2.1I disk0:asr9k-mgbl-3.7.2.1I disk0:asr9k-mpls-3.7.2.1I
  Install operation 2 completed successfully at 18:07:48 UTC Sat Jun 03 2006.

  Summary:
      Install method: parallel
      Summary of changes on nodes 0/1/SP, 0/6/SP, 0/SM0/SP, 0/SM1/SP,
      0/SM2/SP, 0/SM3/SP:
          Activated:    asr9k-diags-3.7.2.1I
          No processes affected

      Summary of changes on nodes 0/1/CPU0, 0/6/CPU0:
          Activated:    asr9k-diags-3.7.2.1I
                        asr9k-mcast-3.7.2.1I
                        asr9k-mpls-3.7.2.1I
           1 asr9k-mpls processes affected (0 updated, 1 added, 0 removed, 0 impacted)
```

```
          2 asr9k-mcast processes affected (0 updated, 2 added, 0 removed, 0 impacted)

    Summary of changes on nodes 0/RP0/CPU0, 0/RP1/CPU0:
        Activated:    asr9k-diags-3.7.2.1I
                      asr9k-k9sec-3.7.2.1I
                      asr9k-mcast-3.7.2.1I
                      asr9k-mgbl-3.7.2.1I
                      asr9k-mpls-3.7.2.1I
         6 asr9k-mgbl processes affected (0 updated, 6 added, 0 removed, 0 impacted)
         8 asr9k-mpls processes affected (0 updated, 8 added, 0 removed, 0 impacted)
         7 asr9k-k9sec processes affected (0 updated, 7 added, 0 removed, 0 impacted)
        14 asr9k-mcast processes affected (0 updated, 14 added, 0 removed, 0 impacted)

Install logs:
    Install operation 2 'install activate disk0:asr9k-diags-3.7.2.1I
    disk0:asr9k-k9sec-3.7.2.1I disk0:asr9k-mcast-3.7.2.1I disk0:asr9k-mgbl-3.7.2.1I
    disk0:asr9k-mpls-3.7.2.1I' started by user 'labuser' at 18:06:32 UTC
    Sat Jun 03 2006.
    Info:     The changes made to software configurations will not be
    Info:     persistent across system reloads. Use the command 'admin install
    Info:     commit' to make changes persistent.
    Info:     Please verify that the system is consistent following the
    Info:     software change using the following commands:
    Info:         show system verify
    Info:         install verify packages
    Install operation 2 completed successfully at 18:07:48 UTC Sat Jun 03 2006.
```

Package Management Procedures

Note Review the concepts about package management before performing the tasks described in this module.

Activation and Deactivation Prerequisites

These prerequisites must be met for a package to be activated or deactivated:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- Verify that all cards are installed and operating properly. For example, do not activate or deactivate packages while cards are booting, while cards are being upgraded or replaced, or when you anticipate an automatic switchover activity.
- If a ROM Monitor upgrade is required for the software package, the upgrade must be completed before the package is activated. For ROM Monitor upgrade information and procedures, see *ROM Monitor Configuration Guide for Cisco ASR 9000 Routers*.
- Check the sanity of the configuration file system and recover from any internal inconsistencies by using the **cfs check** command.

```
RP/0/RSP0/CPU0:router# cfs check

Tue Sep 20 07:22:03.374 DST
```

```
Creating any missing directories in Configuration File system...OK
Initializing Configuration Version Manager...OK
Syncing commit database with running configuration...OK
```

- Clear any inconsistency alarms and remove any failed configurations using the **clear configuration inconsistency** command.

  An inconsistency alarm is set when there is a failure to restore the configuration; this can occur during router startup, or when a line card or route switch processor (RSP) card is inserted or removed. If an inconsistency alarm is set, a message similar to the one in this example is displayed:

  ```
  RP/0/0/CPU0:May 26 11:58:40.662 : cfgmgr-rp[130]: %MGBL-CONFIGCLI-3
     BATCH_CONFIG_FAIL : 28 config(s) failed during startup. To view
     failed config(s) use the command - "show configuration failed startup"
  ```

  When the inconsistency alarm is set, all configuration commit operations fail until the alarm is cleared.

- Although more than one version of a software package can be added to a storage device, only one version of a package can be active for any card.
- Some packages require the activation or deactivation of other packages.
- The package being activated must be compatible with the current active software set.
- Package activation from EXEC mode is supported for specific packages and upgrades, such as optional packages and SMUs. Packages that do not support EXEC mode activation can only be activated for the entire router from administration EXEC mode.

Activation is performed only after the package compatibility checks and API version compatibility checks have been passed. If a conflict is found, an on-screen error message is displayed.

While a software package is being activated, other requests are not allowed to run on any of the impacted nodes. Package activation is completed when a message similar to this one appears:

```
Install operation 2 completed successfully at 20:30:29 UTC Mon Nov 14 2005.
```

Each CLI install request is assigned a request ID, which can be used later to review the events.

Obtaining and Placing Cisco IOS XR Software

This section contains information to locate the available software packages and to transfer them either to a local storage device or to a network server. When this is done, the package or packages can be added and activated on the router.

There are two primary ways to obtain packages in Cisco IOS XR software:

- Request the software from Cisco on a flash disk that you can insert into the removable flash disk slot (usually flash disk1:). Flash disk1: is optional. When it is installed, flash disk1: can be used to store PIE files, which can then be used to add new software to the boot device (usually flash disk0:).
- Download the Cisco IOS XR software packages to a local storage device of the DSC, such as flash disk1:, or to a remote server, such as a tftp or rcp server.

The boot device is the local disk on the DSC where Cisco IOS XR software is added and activated. PIE files should not be stored on this boot device. The default boot device is disk0:. All PIE files should be stored on flash disk1:.

Transferring installation files from a network file server to a local storage device

If the Cisco IOS XR software PIE or RPM files are located on a remote TFTP, FTP, SFTP, or rcp server, you can copy the files to a local storage device such as disk1:. When the PIE or RPM files are located on a local storage device, the software packages can be added and activated on the router from that storage device. Table 13: Download Protocols Supported by Cisco IOS XR Software, on page 155 describes the supported server protocols, and the CLI syntax used copy files from each server type to the local storage device.

Tip Cisco IOS XR software PIE or RPM files can also be added to the router boot device directly from the remote server.

Note Consult your system administrator for the location and availability of your network server.

Table 13: Download Protocols Supported by Cisco IOS XR Software

Name	Description
Trivial File Transfer Protocol	TFTP allows files to be transferred from one computer to another over a network, usually without the use of client authentication (for example, username and password). It is a simplified version of FTP. **Note** Some Cisco IOS XR software images may be larger than 32 MB, and the TFTP services provided by some vendors may not support a file this large. If you do not have access to a TFTP server that supports files larger than 32 MB, download the software image using FTP or rcp.
File Transfer Protocol	FTP is part of the TCP/IP protocol stack and requires a username and password.
Remote Copy Protocol	The rcp protocol uses TCP to ensure the reliable delivery of data, and rcp downloads require a usernames.
SSH File Transfer Protocol	SFTP is part of the SSHv2 feature in the Security package and provides for secure file transfers. For more information, see the *System Security Configuration Guide for Cisco ASR 9000 Series Routers*.

The router commands listed in Table 14: Commands for Copying Package Files to the Router, on page 156 show how to copy package files to the router using three types of file transfer protocols.

Table 14: Commands for Copying Package Files to the Router

Server Type	Command and Examples
TFTP	The following command syntax is used: **copy tftp:**// *hostname_or_ipaddress* / *directory-path* / *filename* **disk1:** Example: `RP/0/RSP0/CPU0:router# copy tftp://10.1.1.1/images/comp-asr9k-mini.pie disk1:`
FTP	The following command syntax is used: **copy ftp:**// *username* **:** *password* **@** *hostname_or_ipaddress* / *directory-path* / *filename* **disk1:** Example: `RP/0/RSP0/CPU0:router# copy ftp://john:secret@10.1.1.1/images/comp-asr9k-mini.pie disk1:`
rcp	The following command syntax is used: **copy rcp:**// *username* **@** *hostname_or_ipaddress* / *directory-path* / *filename* **disk1:** Example: `RP/0/RSP0/CPU0:router# copy rcp://john@10.1.1.1/images/comp-asr9k-mini.pie disk1:`

Table 15: Command Variables for Copying and Adding Packages from a Network Server, on page 156 describes the command variables for copying packages from a network server.

Table 15: Command Variables for Copying and Adding Packages from a Network Server

Variable	Description
hostname_or_ipaddress	Host name or IP address of the server that stores the source file.
filename	Name of the PIE or RPM file (package). See the Overview of Cisco IOS XR Software Packages, on page 117 for descriptions of the available packages.
username	Required for FTP and rcp only and must be a valid username on the FTP or rcp server.
password	Required for FTP only. If a password is not provided, the networking device accepts anonymous FTP.

Variable	Description
directory-path	The specified directory should be a directory under the home directory of the user. In the rcp and FTP examples in Table 14: Commands for Copying Package Files to the Router, on page 156, the file being downloaded is in a subdirectory called "images" in the home directory of the user "john." **Note** For FTP and rcp services, *directory-path* is the directory relative to the *username* home directory. If you want to specify an absolute path for the directory, you must add a "/" following the server address.

When the installation files have been transferred to a network file server or the router, you are ready to activate or upgrade the software.

Note Files with the `vm` extension are bootable installation files used only to replace all current Cisco IOS XR software. These files are installed from ROM monitor mode and cause significant router downtime. We recommend installing or upgrading software packages using PIE or RPM files only, as described in this chapter. See *ROM Monitor Configuration Guide for Cisco ASR 9000 Routers* for information on installing from `vm` files.

Related Topics

Adding and activating packages, on page 157
Overview of Cisco IOS XR Software Packages, on page 117

Adding and activating packages

The procedure in this section describes how to upgrade or add Cisco IOS XR software PIE or RPM files that are stored on a local storage device, such as a flash disk, or on a remote TFTP, FTP, SFTP, or rcp server. The PIE or RPM software file can include any of the following:

- The Cisco IOS XR Unicast Routing Core Bundle (six packages in one composite PIE or RPM file)
- Any of the optional packages (one package per PIE or RPM file)
- Software maintenance upgrades (SMUs)

When you need to add and activate two or more of the preceding package types, you should add and activate them in the order listed above.

Note When adding and activating two or more packages, optional packages can be activated together. Also, if the operation is a reload, multiple packages can be activated together. For example, five reload SMUs can be activated together or the Cisco IOS XR Unicast Routing Core Bundle plus the SMUs and optional packages can be activated together.

For a description of the software management process, see the *Related Topics* section.

These instructions are also used to downgrade software packages.

Note By default, installation operations are performed asynchronously: the CLI prompt is returned before the operation is complete, allowing the operator to continue work while the installation is completed in the background. Use the **synchronous** keyword at the end of install commands to delay the return of the CLI prompt until an installation operation is complete. See the *Related Topics* section for more information.

Before you begin

Before upgrading or adding packages, verify that these prerequisites have been met:

- Verify that the ROMMON version is correct. For instructions on upgrading ROM Monitor, see *ROM Monitor Configuration Guide for Cisco ASR 9000 Routers*.
- All packages to be upgraded or added are present on a local storage device (for example a flash disk), or a network file server.
- Prerequisites for the activation of packages are met as described in the Prerequisites section.
- Complete the procedures described in the Preparing for Software Installation Operations, on page 123 section.

SUMMARY STEPS

1. Connect to the console port and log in.
2. (Optional) **dir** *flash-disk* **:**
3. **admin**
4. **install add** [**source** *source-path* | **tar**] file [**activate**]
5. (Optional) **show install inactive summary**
6. **install activate** {**id** *add-id* | *device package*} [**test**] [**location** *node-id*] [**pause sw-change**] [**sdr** *sdr-name*] [**prompt-level** {**all** | **none**}] [**auto-abort-timer** *time*
7. Repeat Step 4, on page 159 through Step 6, on page 160 until all packages are activated.
8. (Optional) **show install active summary**
9. (Optional) **install verify packages**
10. (Optional) **exit**
11. (Optional) **show system verify start**
12. **admin**
13. (Optional) **install commit**
14. Upgrade the field-programmable device (FPD) software, if necessary.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	Connect to the console port and log in.	Establishes a CLI management session with the SDR. Connect to the console port for the active DSC.

	Command or Action	Purpose
		For more information on console connections, see *Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*.
Step 2	(Optional) **dir** *flash-disk* **:** **Example:** `RP/0/RSP0/CPU0:router# dir disk1:`	Displays the package files that are available for package upgrades and additions. **Note** Only PIE or RPM files can be added and activated using this procedure.
Step 3	Required: **admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode. This step is not applicable for Cisco IOS XR 64-bit software. **Note** Some **show install** commands can be entered in EXEC mode on an SDR.

Step 4	**install add** [**source** *source-path* | **tar**] file [**activate**]
	Example:
	`RP/0/RSP0/CPU0:router(admin)# install add disk1:asr9k-mgbl.pie-3.8.30.1i`
	or
	`RP/0/RSP0/CPU0:router(admin)# install add source tftp://10.1.1.1/images/ asr9k-k9sec-p.pie asr9k-mpls-p.pie asr9k-mcast-p.pie`
	or
	`RP/0/RSP0/CPU0:router(admin)# install add ftp://john:secret@10.1.1.1/images/asr9k-k9sec-p.pie`
	or
	`RP/0/RSP0/CPU0:router(admin)# install add tar rcp://john@10.1.1.1/images/asr9k-iosxr-3.6.0.tar`
		Unpacks a PIE or RPM file from local storage device or network server and adds the package files to the boot device of the router. The boot device is located on the DSC.
		• If the **source** keyword is used, the *source-path* specifies the directory path that is used for multiple filenames in the same directory.
		• If the **tar** keyword is used, all PIE or RPM files contained in the tar file are unpacked.
		The *file* argument can take any of these formats:
		• *device filename*
		• **tftp:**//*hostname_or_ipaddress /directory-path /filename*
		• **ftp:**// *username***:***password***@***hostname_or_ipaddress /directory-path/filename*
		• **rcp:**//*username***@***hostname_or_ipaddress/ directory-path/filename*
		These are descriptions for each of the terms used here:
		• *device*—Name of the local storage device where the PIE or RPM file is stored, such as **disk1:**.
		• *filename*—Name of the PIE or RPM file you want to add. If the **tar** keyword is used, the *file* argument is the name of a tar file containing one or more PIE or RPM files, or directories containing PIE or RPM files.
		• **tftp://**—Unpacks the PIE or RPM file from a network server using Trivial File Transfer Protocol.

	Command or Action	Purpose
		• **ftp://**—Unpacks the PIE or RPM file from a network server using File Transfer Protocol. • **rcp://**—Unpacks the PIE or RPM file from a network server using Remote Copy Protocol • *hostname_or_ipaddress*—Host name or IP address of the network file server. • *directory-path*—Network file server path that leads to the PIE or RPM file to be added. • *username*—Username of user that has access privileges to the directory in which the PIE or RPM file is stored. • *password*—Password associated with the username of user that has access privileges to the directory in which the PIE or RPM file is stored. • **activate**—Automatically activates the software package after it is successfully added. **Note** Multiple versions of a software package can be added to the storage device without impacting the running configuration, but only one version of a package can be activated for a card.
Step 5	(Optional) **show install inactive summary** **Example:** `RP/0/RSP0/CPU0:router(admin)# show install inactive summary`	Displays the inactive packages on the router. Verify that the package added in the previous step appears in the display.
Step 6	**install activate** {**id** *add-id* | *device package*} [**test**] [**location** *node-id*] [**pause sw-change**] [**sdr** *sdr-name*] [**prompt-level** {**all** | **none**}] [**auto-abort-timer** *time* **Example:** `RP/0/RSP0/CPU0:router(admin)# install activate disk0:asr9k-mini-px-4.3.99`	Activates a package that was added to the router. (Skip this step if the package was activated earlier with the **install add** command.) • **id** *add-id*—Specifies the package using the operation ID of the **install add** operation in which you added the package. The operation ID is provided in the output of the **install add** command. You can also use **show install log** to display installation operation IDs. • *device:package*—Specifies the package by name. Replace the *device:package* argument with the name of the boot device and inactive package, which can be displayed as described in the previous step. **Note** Press **?** after a partial package name to display all possible matches available for activation. If there is

	Command or Action	Purpose
		only one match, press [TAB] to fill in the rest of the package name. • **location** *node-id*—Activates a package for a specific card (node). To display a list of node IDs for the entire system, enter the **show platform** command in administration EXEC mode. A package cannot be activated on a single node unless some version of the package being activated is already active on all nodes. **Note** By default, packages are activated for all cards supported by that package. • **pause sw-change**—Pauses the operation after preparatory checks and before the configuration is locked for the actual activation. This action enables you to hold the operation while you perform configuration changes, and proceed with the activation whenever you choose. This operation is useful, for example, if your workflow involves configuring a router out of the network during software installation and you want to minimize the time that the router is out of the network. Follow onscreen instructions to control the pausing and completion of the operation. • **prompt-level**—Use a prompt-level of **all** to view all stages of the installation process and to specify whether to continue, or not. • **auto-abort-timer**—Specifies an abort timer value, in minutes, which when expired loads the last committed loadpath. The default is 60.The timer is disabled by default.After the installation, if the activated software is working correctly, use the **install commit** command to cancel the timer and commit the new loadpath. **Note** The package being activated must be compatible with the currently active software to operate. When an activation is attempted, the system runs an automatic compatibility check to ensure that the package is compatible with the other active software on the router. The activation is permitted only after all compatibility checks have been passed. **Tip** When activating packages, use the **test** option to test the effects of a command without impacting the running

	Command or Action	Purpose
		system. After the activation process finishes, enter the **show install log** command to display the process results.
Step 7	Repeat Step 4, on page 159 through Step 6, on page 160 until all packages are activated.	Activates additional packages as required.
Step 8	(Optional) **show install active summary** **Example:** `RP/0/RSP0/CPU0:router(admin)# show install active`	Displays all active packages. Use this display to determine if the correct packages are active:
Step 9	(Optional) **install verify packages** **Example:** `RP/0/RSP0/CPU0:router(admin)# install verify packages`	Verifies the consistency of a installed software set with the package file from which it originated. This command can be used as a debugging tool to verify the validity of the files that constitute the packages, to determine whether there are any corrupted files. This command also checks for corruptions of installation state files and MBI image files. This command is particularly useful when issued after the activation of a package or upgrading the Cisco IOS XR software to a major release. **Note** The **install verify packages** command can take up to two minutes for each package to process.
Step 10	(Optional) **exit** **Example:** `RP/0/RSP0/CPU0:router(admin)# exit`	Exits administration EXEC mode and returns to EXEC mode. Use this command only if you performed the installation operations in administration EXEC mode.
Step 11	(Optional) **show system verify start** **Example:** `RP/0/RSP0/CPU0:router# show system verify start`	Starts the system status check.
Step 12	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.

Step 13	(Optional) **install commit**
	Example:
	`RP/0/RSP0/CPU0:router# dir disk1:`
	`RP/0/RSP0/CPU0:router(admin)# install commit`
		Commits the current set of packages for an SDR or for all SDRs so that these packages are used if the router is restarted.
		• This command can be used from either administration EXEC or EXEC mode.
		For more information, see the *Related Topics* section.

	Command or Action	Purpose
Step 14	Upgrade the field-programmable device (FPD) software, if necessary.	Whenever a Cisco IOS XR software image that supports SPAs and SIPs is released, a companion SPA or SIP FPD image is bundled with the Cisco IOS XR software release. However, the FPD image is not automatically upgraded. You must manually upgrade the FPD image running on the SPA or SIP when you upgrade the Cisco IOS XR software image. FPD versions must be compatible with the Cisco IOS XR software that is running on the router. For information on FPDs, including instructions to upgrade FPD images, see the *Upgrading FPD Cisco IOS XR Software* section.

Related Topics

Obtaining and Placing Cisco IOS XR Software, on page 154
Activation and Deactivation Prerequisites, on page 153
Preparing for Software Installation Operations, on page 123
Downgrading Packages, on page 149
PIE Filenames and Version Numbers, on page 119
Committing the Active Package Set, on page 165

Examples

Adding a Package: Example

The following example shows how to add the contents of a PIE file on disk1: to the boot device. Because the software package is added to the boot device by default, it is not necessary to specify the destination device in the CLI.

```
RP/0/RSP0/CPU0:router(admin)# install add disk1:asr9k-mpls-p.pie-3.7.2 synchronous

Install operation 4 'install add /disk1:asr9k synchronous' started by user
'cisco' at 18:10:18 UTC Sat Apr 08 2009.
Info:     The following package is now available to be activated:
Info:
Info:         disk0:asr9k-mpls-3.7.2
Info:
Install operation 4 completed successfully at 18:14:11 UTC Sat Apr 08 2009.
```

The following example shows how to add the contents of a PIE file on a TFTP server to the boot device:

```
RP/0/RSP0/CPU0:router(admin)# install add tftp://209.165.201.1/
asr9k-mpls.pie synchronous

Install operation 4 '(admin) install add /tftp://209.165.201.1/asr9k-mpls.pie synchronous'

  started by user 'cisco' at 18:16:18 UTC Thu Jan 03 2009.
Info:     The following package is now available to be activated:
Info:
Info:         disk0:asr9k-mpls-3.7.2
```

```
Info:
Install operation 4 completed successfully at 18:19:10 UTC Thu Jan 03 2009.
```

Activating a Package: Example

The following example shows the activation of the MPLS package. The package is activated on the boot device disk0:.

```
RP/0/RSP0/CPU0:router(admin)# install activate disk0:
asr9k-mpls-3.7.2 synchronous

Install operation 15 'install activate disk0:asr9k-mpls-3.7.2 synchronous'
started by user 'lab' at 19:15:33 UTC Sat Apr 08 2009.
Info:     The changes made to software configurations will not be persistent
Info:     across system reloads. Use the command 'admin install commit' to make
Info:     changes persistent.
Info:     Please verify that the system is consistent following the software
Info:     change using the following commands:
Info:         show system verify
Info:         install verify packages
Install operation 5 completed successfully at 19:16:18 UTC Sat Apr 08 2009.
```

Activating a Package by Specifying an Operation ID: Example

The following example shows the activation of the MPLS package using the operation ID of the **install add** operation that added the package:

```
RP/0/RSP0/CPU0:router(admin)# install activate id 4

Install operation 5 '(admin) install activate id 4' started by user 'lab' via
CLI at 18:20:17 UTC Thu Jan 03 2009.
Info:     This operation will activate the following package:
Info:         disk0:asr9k-mpls-3.7.2
Info:     Install Method: Parallel Process Restart
The install operation will continue asynchronously.
Info:     The changes made to software configurations will not be persistent
Info:     across system reloads. Use the command '(admin) install commit' to
Info:     make changes persistent.
Info:     Please verify that the system is consistent following the software
Info:     change using the following commands:
Info:         show system verify
Info:         install verify packages
Install operation 5 completed successfully at 18:21:30 UTC Thu Jan 03 2009.
```

Adding and Activating a Package from an FTP File Server with One Command: Example

To add and activate a package with a single command, enter the **install add** command with the **activate** keyword. In the following example, the Manageability PIE located on disk1: is verified, unpacked, and added to the boot device disk0. Because this operation is performed in administration EXEC mode, the package is activated for all SDRs in the system.

```
RP/0/RSP0/CPU0:router(admin)# install add disk1:
```

```
asr9k-mgbl-p.pie-3.7.2 activate

Install operation 4 'install add /disk1:asr9k-mgbl-3.7.2 activate' started
by user 'cisco' at 07:58:56 UTC Wed Mar 01 2009.
The install operation will continue asynchronously.
:router(admin)#Part 1 of 2 (add software): Started
Info:     The following package is now available to be activated:
Info:
Info:         disk0:asr9k-mgbl-3.7.2
Info:
Part 1 of 2 (add software): Completed successfully
Part 2 of 2 (activate software): Started
Info:     The changes made to software configurations will not be
  persistent across system reloads. Use the command 'admin install
Info:     commit' to make changes persistent.
Info:     Please verify that the system is consistent following
  the software change using the following commands:
Info:         show system verify
Info:         install verify packages
Part 2 of 2 (activate software): Completed successfully
Part 1 of 2 (add software): Completed successfully
Part 2 of 2 (activate software): Completed successfully
Install operation 4 completed successfully at 08:00:24 UTC Wed Mar 01 2009.
```

Displaying the Active Packages: Example

The following example displays a summary of the active packages on a router. Because this operation is performed in administration EXEC mode, the active packages for all SDRs are displayed.

```
RP/0/RSP0/CPU0:router(admin)# show install active summary
Mon Jun 22 23:41:19.509 PST
Default Profile:
  SDRs:
    Owner
  Active Packages:
    disk0:comp-asr9k-mini-3.9.0.12I
    disk0:asr9k-fpd-3.9.0.12I
    disk0:asr9k-k9sec-3.9.0.12I
    disk0:asr9k-mcast-3.9.0.12I
    disk0:asr9k-mgbl-3.9.0.12I
    disk0:asr9k-mpls-3.9.0.12I
```

Committing the Active Package Set

When a package is activated, it becomes part of the current running configuration. To make the package activation persistent across system-wide reloads, enter the **install commit** command. On startup, DSC of the owner SDR loads this committed software set. If the system is reloaded before the current active software is committed with the **install commit** command, the previously committed software set is used.

If the system is reloaded before the current active software is committed with the **install commit** command, the previously committed software set is used.

Tip Before committing a package set, verify that the SDR is operating correctly and is forwarding packets as expected.

Note Step 1 is applicable only to Cisco IOS XR 32-bit software.

SUMMARY STEPS

1. **admin**
2. **install commit**
3. **show install committed** [**detail** | **summary** | **verbose**] [**location** *node-id*]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 2	Required: **install commit** **Example:** `RP/0/RSP0/CPU0:router(admin)# install commit`	Commits the current set of packages on the router so that these packages are used if the router is restarted. • This command can be used from either administration EXEC or EXEC mode.
Step 3	**show install committed** [**detail** | **summary** | **verbose**] [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router(admin)# show install committed`	Displays which packages are committed. • Enter this command in administration EXEC mode to display information for the entire system. • Enter this command in EXEC mode to display information for a specific SDR only. • For more information on the command options, see *System Management Command Reference for Cisco ASR 9000 Series Routers*.

Examples

Committing the Active Package Set: Example

In the following example, the active software packages are committed on the router:

```
RP/0/RSP0/CPU0:router(admin)# install commit

Install operation 16 'install commit' started by user 'lab' at 19:18:58 UTC
Sat Apr 08 2009.
Install operation 16 completed successfully at 19:19:01 UTC Sat Apr 08 2009.
```

Displaying the Committed Package Versions: Example

In the following example, the committed packages are shown for the owner SDR:

```
RP/0/RSP0/CPU0:router(admin)# show install committed

Tue Jun 23 05:11:29.968 PST
Secure Domain Router: Owner

  Node 0/RSP0/CPU0 [RP] [SDR: Owner]
    Boot Device: disk0:
    Boot Image: /disk0/asr9k-os-mbi-3.9.0.12I/mbiasr9k-rp.vm
    Committed Packages:
      disk0:comp-asr9k-mini-3.9.0.12I
      disk0:asr9k-fpd-3.9.0.12I
      disk0:asr9k-k9sec-3.9.0.12I
      disk0:asr9k-mcast-3.9.0.12I
      disk0:asr9k-mgbl-3.9.0.12I
      disk0:asr9k-mpls-3.9.0.12I

  Node 0/1/CPU0 [LC] [SDR: Owner]
    Boot Device: mem:
    Boot Image: /disk0/asr9k-os-mbi-3.9.0.12I/lc/mbiasr9k-lc.vm
    Committed Packages:
      disk0:comp-asr9k-mini-3.9.0.12I
      disk0:asr9k-fpd-3.9.0.12I
      disk0:asr9k-mcast-3.9.0.12I
      disk0:asr9k-mpls-3.9.0.12I

  Node 0/4/CPU0 [LC] [SDR: Owner]
    Boot Device: mem:
    Boot Image: /disk0/asr9k-os-mbi-3.9.0.12I/lc/mbiasr9k-lc.vm
    Committed Packages:
      disk0:comp-asr9k-mini-3.9.0.12I
      disk0:asr9k-fpd-3.9.0.12I
      disk0:asr9k-mcast-3.9.0.12I
      disk0:asr9k-mpls-3.9.0.12I

  Node 0/6/CPU0 [LC] [SDR: Owner]
    Boot Device: mem:
    Boot Image: /disk0/asr9k-os-mbi-3.9.0.12I/lc/mbiasr9k-lc.vm
    Committed Packages:
      disk0:comp-asr9k-mini-3.9.0.12I
      disk0:asr9k-fpd-3.9.0.12I
      disk0:asr9k-mcast-3.9.0.12I
      disk0:asr9k-mpls-3.9.0.12I
```

As with the **show install active** command, the **show install committed** command may display a composite package that represents all packages in the Cisco IOS XR Unicast Routing Core Bundle.

Deactivating and Removing Cisco IOS XR Software Packages

When a package is deactivated, it is no longer active on the router, but the package files remain on the boot disk. The package files can be reactivated later, or they can be removed from the disk.

A package is deactivated using the following methods:

- When a newer version of a package is activated, the earlier version of the package is automatically deactivated. See *Related Topics* for more information.

Note Activating a software maintenance upgrade (SMU) does not cause any earlier SMUs or the package to which the SMU applies to be automatically deactivated.

- When an earlier version of a package is activated, the newer version is deactivated automatically. See *Related Topics* for more information.
- A specific package is deactivated using the **install deactivate** command. This command turns off the package features for a card or card type.

Before you begin

The following are the restrictions when deactivating and removing Cisco IOS XR Software packages:

- A package cannot be deleted if it is part of the running or committed software of the SDR.
- A package cannot be deactivated if that package is required by another active package. When a deactivation is attempted, the system runs an automatic check to ensure that the package is not required by other active packages. The deactivation is permitted only after all compatibility checks have been passed.
- Router reloads: If the deactivation requires a router reload, a confirmation prompt appears. Use the **install deactivate** command with the **prompt-level none** keywords to automatically ignore any reload confirmation prompts and proceed with the package deactivation. The router reloads if required.
- Node reloads: If a software operation requires a node reload, the configuration register for that node should be set to autoboot. If the config-register for the node is not set to autoboot, then the system automatically changes the setting and the node reloads. A message describing the change is displayed.
- 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 field-programmable gate array (FPGA) may not operate properly until the incompatibility is resolved. For information on FPDs, including instructions to upgrade FPD images, see the *Upgrading FPD Cisco IOS XR Software* module of *Interface and Hardware Component Configuration Guide for Cisco ASR 9000 Series Routers*.

SUMMARY STEPS

1. Connect to the console port and log in.
2. **admin**
3. **install deactivate** { **id** *add-id* | *device* **:** *package* } [**location** *node-id*] [**test**] [**pause sw-change**]
4. (Optional) **show install inactive summary**
5. (Optional) **install verify packages**
6. **exit**
7. (Optional) **show system verify start**

8. (Optional) **show system verify** [**detail** | **report**]
9. **admin**
10. (Optional) **install commit**
11. (Optional) **install remove** { **id** *add-id* | *device* **:** *package* | **inactive** } [**test**]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	Connect to the console port and log in.	Establishes a CLI management session with the SDR. Connect to the console port for the active DSC. For more information on console connections, see *Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*.
Step 2	Required: **admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode. • From administration EXEC mode, you can perform installation operations for the router. To enter administration EXEC mode, you must be logged in to the owner SDR and have root-system access privileges. • This command is not required.
Step 3	**install deactivate** { **id** *add-id* | *device* **:** *package* } [**location** *node-id*] [**test**] [**pause sw-change**] **Example:** `RP/0/RSP0/CPU0:router(admin)# install deactivate disk0:asr9k-diags-3.7.2`	Deactivates a package on a router. • To deactivate all packages that were added in one or more specific install add operations, or specify packages by name, use the **id** *add-id* keyword and argument. The operation ID of an install add operation is indicated in the syslog displayed during the operation and in the output of the **show install log** command. • To deactivate a package for the router, use this command in administration EXEC mode. • To deactivate a package when logged in to an SDR, use this command in EXEC mode. • Use the **location** *node-id* keyword and argument to deactivate the package for a specific node, if supported. • Use the **pause sw-change** keywords to pause the operation after preparatory checks and before the configuration is locked for the actual deactivation. This enables you to hold the operation while you perform configuration changes, and proceed with the deactivation whenever you choose. This is useful, for

	Command or Action	Purpose
		example, if your workflow involves configuring a router out of the network during software changes and you want to minimize the time that the router is out of the network. Follow the onscreen instructions to control the pausing and completion of the operation. **Note** Press **?** after a partial package name to display all possible matches available for deactivation. If there is only one match, press [TAB] to fill in the rest of the package name. When a package is deactivated for an SDR from administration EXEC mode, a notification message appears on the console for that SDR, with information on the impact of the deactivation.
Step 4	(Optional) **show install inactive summary** **Example:** `RP/0/RSP0/CPU0:router(admin)# show install inactive summary`	Displays the inactive packages on the router.
Step 5	(Optional) **install verify packages** **Example:** `RP/0/RSP0/CPU0:router(admin)# install verify packages`	Verifies the consistency of an installed software set with the package file from which it originated. This command can be used as a debugging tool to verify the validity of the files that constitute the packages, to determine if there are any corrupted files. This command also checks for corruptions of installation state files and MBI image files. This command is particularly useful when issued after the activation of a package or upgrading the Cisco IOS XR software to a major release. **Note** The **install verify packages** command can take up to two minutes per package to process.
Step 6	Required: **exit** **Example:** `RP/0/RSP0/CPU0:router(admin)# exit`	Exits administration EXEC mode and returns to EXEC mode. Use this command only if you performed the installation operations in administration EXEC mode.
Step 7	(Optional) **show system verify start** **Example:** `RP/0/RSP0/CPU0:router# show system verify start`	Starts the system status check.
Step 8	(Optional) **show system verify** [**detail** | **report**]	Displays system status information. A variety of information is displayed including the memory and CPU

	Command or Action	Purpose
	Example: `RP/0/RSP0/CPU0:router# show system verify`	usage, process status, protocol status, and other status information. Use this information to verify that the system is stable. Enter this command in EXEC mode. • **detail**—Displays additional information at the card and processor level, including actual numbers. • **report**—Displays the same information as the default **show system verify** command **Note** Although most of the output should display the status "OK," some processes may show other output, such as "Warning." This does not specifically indicate a problem. Contact your Cisco technical support representative for more information on the output of this command.
Step 9	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 10	(Optional) **install commit** **Example:** `RP/0/RSP0/CPU0:router(admin)# install commit`	Commits the current set of packages so that these packages are used if the router is restarted. Packages can be removed only if the deactivation operation is committed. **Note** This command is entered in administration EXEC or EXEC mode.
Step 11	(Optional) **install remove** { **id** *add-id* | *device* **:** *package* | **inactive** } [**test**] **Example:** `RP/0/RSP0/CPU0:router(admin)# install remove disk0:asr9k-diags-3.8.30`	Removes the inactive package. • Only inactive packages can be removed. • Packages can be removed only if they are deactivated from all cards in the router. • The package deactivation must be committed. • To remove a specific inactive package from a storage device, use the **install remove** command with the device**:** *package* arguments. • To remove all packages that were added in one or more specific **install add** operations, use the **id** *add-id* keyword and argument. The operation ID of an **install add** operation is indicated in the syslog displayed during the operation and in the output of the **show install log** command. If you specify packages according to operation ID, all the packages that were added by the specified operation must still be on the router.

	Command or Action	Purpose
		• To remove all inactive packages from all nodes in the system, use the **install remove** command with the **inactive** keyword. • You can use the **install remove** command in either administration EXEC or EXEC mode.

Related Topics

Adding and activating packages, on page 157
Committing the Active Package Set, on page 165

Examples

In the following examples, a package is deactivated from the router. The changes are committed and the inactive package is removed from the router.

Deactivating the Package: Example

```
RP/0/RSP0/CPU0:router(admin)# install deactivate disk0:asr9k
-diags-.7.2

Install operation 27 'install deactivate disk0:asr9k-diags-3.7.2' started by
user 'lab' at 23:29:37 UTC Sat Apr 15 2009.
The install operation will continue asynchronously.
Info:     The changes made to software configuration
Info:     across system reloads. Use the command 'admin install commit' to make
Info:     changes persistent.
Info:     Please verify that the system is consistent following the software
Info:     change using the following commands:
Info:         show system verify
Info:         install verify packages
Install operation 27 completed successfully at 23:30:22 UTC Sat Apr 15 2009.
```

Committing the Active Software Set: Example

```
RP/0/RSP0/CPU0:router(admin)# install commit

Install operation 29 'install commit' started by user 'lab' at 23:39:21 UTC
Sat Apr 15 20090.
Install operation 29 completed successfully at 23:39:24 UTC Sat Apr 15 2009.
```

Displaying the Inactive Packages: Example

```
RP/0/RSP0/CPU0:router(admin)# show install inactive summary

Default Profile:
  SDRs:
  Owner
  Inactive Packages:
    disk0:asr9k-diags-3.7.2
```

Removing the Inactive Package from the Router: Example

The following example shows how to remove an inactive package. In this example, the operation is run in test mode. The operation is confirmed and the package is removed.

```
RP/0/RSP0/CPU0:router(admin)# install remove disk0:asr9k-diags-3.7.2 test

Install operation 30 'install remove disk0:hfr-diags-3.7.2 test' started by
user 'lab' at 23:40:22 UTC Sat Apr 15 2009.
Warning:  No changes will occur due to 'test' option being specified. The
Warning:  following is the predicted output for this install command.
Info:     This operation will remove the following package:
Info:         disk0:asr9k-diags-3.7.2
Info:     After this install remove the following install rollback points will
Info:     no longer be reachable, as the required packages will not be present:
Info:         4, 9, 10, 14, 15, 17, 18
Proceed with removing these packages? [confirm] y

The install operation will continue asynchronously.
Install operation 30 completed successfully at 23.
```

Pausing Before Configuration Lock: Example

Note This configuration is supported only on the Cisco IOS XR 32-bit software.

The following example shows how to deactivate a package, pausing the operation before locking the configuration for the actual software deactivation. While the operation is paused, you can enter a configuration mode and perform configurations. When you want to complete the operation, enter the **install operation** *id* **complete** command, or the **install operation** *id* **attach synchronous** command.

```
RP/0/RSP0/CPU0:router(admin)# install deactivate disk0:comp-asr9k
-3.7.2.07I.CSCsr09575-1.0.0 pause sw-change

Install operation 12 '(admin) install deactivate
  disk0:comp-asr9k-3.7.2.07I.CSCsr09575-1.0.0 pause sw-change'
  started by user 'admin' via CLI at 09:06:26 BST Mon Jul 07 2009.
Info: This operation will reload the following nodes in parallel:
Info: 0/0/CPU0 (RP) (SDR: Owner)
Info: 0/1/CPU0 (LC(E3-GE-4)) (SDR: Owner)
Info: 0/5/CPU0 (LC(E3-OC3-POS-4)) (SDR: Owner)
Proceed with this install operation (y/n)? [y]
The install operation will continue asynchronously.
Info: Install Method: Parallel Reload
Info: Install operation 12 is pausing before the config lock is applied for
Info:    the software change as requested by the user.
Info: No further install operations will be allowed until the operation is resumed.
Info: Please continue the operation using one of the following steps:
Info: - run the command '(admin) install operation 12 complete'.
Info: - run the command '(admin) install operation 12 attach synchronous' and then
Info:      answer the query.
```

Additional References

The following sections provide references related to software package management on Cisco IOS XR software.

Related Documents

Related Topic	Document Title
Cisco IOS XR install commands	*Software Package Management Commands on the Cisco ASR 9000 Series Router* module of *System Management Command Reference for Cisco ASR 9000 Series Routers*
Cisco IOS XR getting started material	*Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*
Cisco IOS XR master command index	*Cisco ASR 9000 Series Aggregation Services Router Commands Master List*
Information about user groups and task IDs	*Configuring AAA Services on the Cisco ASR 9000 Series Router* module of *System Security Configuration Guide for Cisco ASR 9000 Series Routers*
ROM Monitor	*ROM Monitor Configuration Guide for Cisco ASR 9000 Routers*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

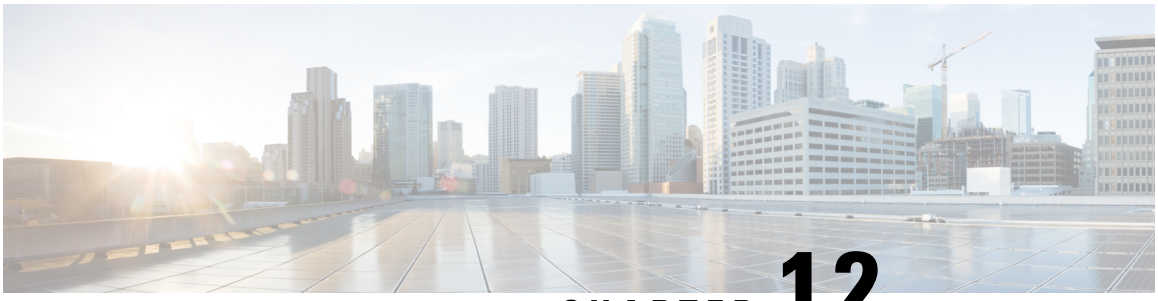
MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 12

Upgrading Field-Programmable Devices

In general terms, *field-programmable devices* (FPDs) are hardware devices implemented on router cards that support separate software upgrades. A *field-programmable gate array* (FPGA) is a type of programmable memory device that exists on most hardware components of the router. The term *FPD* has been introduced to collectively and generically describe any type of programmable hardware device on SIPs and shared port adapters (SPAs), including FPGAs. Cisco IOS XR software provides the Cisco FPD upgrade feature to manage the upgrade of FPD images on SIPs and SPAs.

This chapter describes the information that you must know to verify image versions and to perform an upgrade for SPA or SIP FPD images when incompatibilities arise.

For complete descriptions of the FPD commands listed in this module, refer to the upcoming sections. To locate documentation for other commands that might appear in the course of performing a configuration task, search online in *Cisco ASR 9000 Series Aggregation Services Router Commands Master List*.

Table 16: Feature History for Upgrading FPD Software on Cisco IOS XR Software

Release	Modification
Release 3.9.0	Support for FPD upgrades was introduced.
Release 6.3.1	Support for parallel FPD upgrade for power modules.

This module contains the following topics:

- Upgrading Field-Programmable Device, on page 178
- Prerequisites for FPD Image Upgrades, on page 178
- Overview of FPD Image Upgrade Support, on page 178
- Hitless FPD Firmware Upgrade for Optical Transceiver Modules, on page 182
- FPD upgrade service, on page 183
- YANG Data Model for Field Programmable Device, on page 185
- How to Upgrade FPD Images, on page 186
- Configuration Examples for FPD Image Upgrade, on page 189
- Troubleshooting Problems with FPD Image Upgrades, on page 194

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

- 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.

Whenever an image is released that supports SIPs and SPAs, a companion SIP and SPA FPD image is bundled. However , the FPD image is not automatically upgraded. You must manually upgrade the FPD image running on the SPA or SIP when you upgrade the Cisco IOS XR software image.

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. An FPGA incompatibility on a SPA does not necessarily affect the running of the SPA interfaces; an FPD incompatibility on a SIP disables all interfaces for all SPAs in the SIP until the incompatibility is addressed.

Use the **show hw-module fpd** command to determine if an FPD upgrade is required. A value of 'Yes' in the Upg/Dng? (upgrade/downgrade) column indicates that an upgrade or downgrade is required.

The NCS 5500 supports upgrades for FPGA devices on its SIPs and SPAs. FPGA and ROMMON software upgrades are part of an FPD image package that corresponds to a Cisco IOS XR software image. SIPs and SPAs support manual upgrades for FPGA devices using the Cisco FPD upgrade feature that is further described in this chapter.

Note

- It is mandatory to upgrade all the required FPDs before doing a reload when you are upgrading FPDs on line cards. This is because, partial FPD component upgrades might result in booting errors (in some cases).
- You must not reload any line card or the router before all FPD image upgrades are completed successfully.

Parallel Power Module Upgrade

Power modules can now be upgraded in parallel on Cisco Routers. This feature lets you perform FPD upgrades on multiple power modules simultaneously. The newer power modules (V3) take more time to upgrade separately than their previous counterparts, which increases the total time taken to upgrade a full chassis to an unacceptable limit.

Parallel upgrade process reduces the overall time required to upgrade a full chassis with many power modules. Only power modules that support FPD upgrades can be upgraded in parallel. This includes V3 AC-DC and V2 AC-DC power modules.

Note Power module upgrades are time consuming and cannot be implicitly upgraded or as a part of automatic FPD upgrades. These modules must be upgraded independent of the other fpga upgrades.

To upgrade the power modules in parallel, use **upgrade hw-module location pm-all fpd all** or **upgrade hw-module fpd all location pm-all** command in Admin mode.

To force a power module upgrade, use **upgrade hw-module fpd all force location pm-all** command in Admin mode.

Pre-requisites to perform Parallel Upgrade

- Ensure that all power connections to the power supply are energized. To verify the power supply details, use **show environment power-supply** command in Admin mode.
- Ensure power available to the power supply is equal to the rated power. For example, 6KW power module must have a 6KW power feed. If the power feed to the power supply is less, the excess power calculation will be incorrect and the chassis may run out of power during an upgrade and suffer a sudden shutdown.
- Ensure sufficient or excess power is available in the chassis before you start the upgrade process.
- Do not add or remove any component (Line cards, RPs, power connections) from the chassis during an upgrade. This may cause power failure in the system due to sudden change in power in the system.

Note

- The system upgrades the power modules in random order.
- The number of modules that can be upgraded simultaneously depends on the excess power available to the chassis.
- Ensure you initiate the parallel upgrade process only when all the pre-requisites are satisfied because the upgrade process cannot be aborted in between.

Performing Parallel Power Module Upgrade

To initiate a parallel upgrade process and upgrade all the power modules in the chassis simultaneously, use **pm-all** keyword in the **upgrade hw-module fpd** command in Admin mode.

Example

The following section illustrates parallel power module upgrade implementation:

Verification

Use **show hw-module fpd** command to verify the upgrade:

Manual Power Module Upgrade

Manual Power modules FPD upgrades are supported on Cisco ASR 9000 Series Routers and should be performed in Admin mode only. This feature lets you perform FPD upgrades on individual Power Entry Modules (PEMs) rather than initiating a Parallel Power Module Upgrade.

Only power modules that support FPD upgrades can be upgraded manually. This includes V3 AC-DC and V2 AC-DC power modules

Note Power module upgrades are time consuming and can't be implicitly upgraded or as a part of automatic FPD upgrades. These modules must be upgraded independent of the other fpga upgrades.

To determine which PEMs requires upgrade, use **show hw-module location all fpd**.

PEMs requiring upgrade are in **UPGD SKIP** status.

```
Router#show hw-module location all fpd

Auto-upgrade:Enabled
                                                                        FPD Versions
                                                                       ==============
Location   Card type             HWver FPD device       ATR Status    Running   Programd
-------------------------------------------------------------------------------------------
0/PT0      PWR-4.4KW-DC-V3       2.11  PM0-DT-Pri0MCU       UPGD SKIP   3.00      3.00
0/PT0      PWR-4.4KW-DC-V3       2.11  PM0-DT-Pri1MCU       UPGD SKIP   3.00      3.00
0/PT0      PWR-4.4KW-DC-V3       2.11  PM0-DT-Sec054vMCU    UPGD SKIP   3.00      3.00
0/PT0      PWR-4.4KW-DC-V3       2.11  PM0-DT-Sec154vMCU    UPGD SKIP   3.00      3.00
0/PT0      PWR-4.4KW-DC-V3       2.11  PM0-DT-Sec5vMCU      UPGD SKIP   3.00      3.00
```

To upgrade the power modules manually, use [**admin**] **upgrade hw-module location 0/PT**<location> **fpd** <fpd_device>.

```
Router# admin
Router# upgrade hw-module location 0/PT0 fpd PM0-DT-Pri0MCU
```

Warning Do not perform power module FPD upgrades when the system is running with a single power feed, as this may result in `UPGD FAIL` alarms indicating that one or more FPDs are not in the current state.

Automatic Line Card Reload on FPD Upgrade

This feature automatically reloads a newly inserted line card (LC) after a successful FPD upgrade. The current auto FPD upgrade process does not reload the line card automatically, the user had to manually reload the LC. To enable this feature on Cisco IOS XR 32 bit operating system, use the **fpd auto-reload** command and use **fpd auto-reload enable** command in Cisco IOS XR 64 bit OS.

Implementation Considerations

The following limitation must be considered while configuring automatic line card reload on FPD upgrade:

- In Cisco IOS XR 32-bit OS, FPDs that are part of MPAs are not auto upgraded neither on inserting them to a line card nor when the entire line card gets inserted into a chassis.
- In Cisco IOS XR 64-bit OS, FPDs that are part of MPAs are auto upgraded. But the MPA will not be auto reloaded.
- If the FPD upgrade fails on a line card then the automatic line card reload feature (if enabled) stops the LC from reloading.
- TimingICs do not support **auto fpd upgrade** on ASR 9000 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**

Configuring Automatic Line Card Reload on FPD Upgrade

The auto-reload feature works only if auto-upgrade feature is also configured on the router. The following sample shows how to configure auto-reload feature for Cisco IOS XR 32-bit OS:

```
RP/0/RSP0/CPU0:ios(config)#admin
RP/0/RSP0/CPU0:ios(admin-config)#fpd auto-upgrade
RP/0/RSP0/CPU0:ios(admin-config)#fpd auto-reload
RP/0/RSP0/CPU0:ios(admin-config)#commit
```

The auto-reload feature is only supported on line cards.

The following sample shows how to configure auto-reload feature for Cisco IOS XR 64-bit OS:

```
RP/0/RSP1/CPU0:ios# config
RP/0/RSP1/CPU0:ios(config)#fpd auto-upgrade enable
RP/0/RSP1/CPU0:ios(config)#fpd auto-reload enable
RP/0/RSP1/CPU0:ios(config)#commit
```

Note During the FPD upgrade process, the linecard may display IOS XR RUN state before triggering auto-reload.

Note **To manually reload the line card on FPD upgrade**

During FPD upgrade process, ensure to use **hw-module location** *node-id* **reload** command in EXEC or administration EXEC mode at the end of the upgrade procedure. This cause the selected card(s) to perform a complete hardware reload, which is required for some FPDs.

Hitless FPD Firmware Upgrade for Optical Transceiver Modules

Table 17: Feature History Table

Feature Name	Release Information	Feature Description
Hitless FPD Firmware Upgrade for Optical Transceiver Modules	Release 7.11.1	We now support a hitless FPD firmware upgrade for some optical transceiver modules where the FPD upgrades without shutting down the laser on the optical line card modules. This functionality avoids operational downtime. This process is more efficient and avoids traffic loss. Prior to this release, the FPD upgrades used to shut down the laser. This resulted in traffic loss. This feature is supported on the following variants: • A9K-8HG-FLEX-SE/TR • A9K-20HG-FLEX-SE/TR • A99-10X400GE-X-SE/TR • A9903-20HG-PEC

We are introducing hitless FPD firmware upgrade process to support firmware upgrade without any traffic loss.

In earlier releases, when the FPD upgrade was triggered on the linecards with the Cisco 400G QSFP-DD High-Power (Bright), QSFP-DD ZR+ and ZR optical modules, the laser in these optical modules used to shut down. Once the upgrade was successful, the laser was turned on. This process resulted in traffic loss. From Cisco IOS XR Software Release 7.11.1, as hitless FPD firmware upgrade is introduced, the process of shutting down the laser is eliminated. This makes the FPD upgrade process more robust and avoids traffic loss. This feature enabled by default. The upgrade process is applicable to manual FPD upgrades.

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.
- 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.

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.

Automatic FPD upgrade

Use the **fpd auto-upgrade enable** command to enable the auto upgrade feature.

The FPD images are upgraded as part of the install activation of the new image. The FPDs are upgraded before the router is reloaded.

During an FPD auto-upgrade, the installed FPD rpm package includes an FPD image with a new version of software that is different than the version of the image running on the hardware. Once the FPDs have been upgraded, even if the base image is rolled backed to the older version, the FPD will not be downgraded to its previous version.

When a reload package is installed with new FPD images, the FPD images are upgraded before the router gets reloaded. This feature is controlled through an fpd auto-upgrade configuration option. The auto-upgrade feature does not address the following:

- FPD Upgrade during initial boot
- FPD Upgrade during new card insertion

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. Line-cards, fabric cards and RP cards 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.

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)
```

Upgrade TimingIC-A and TimingIC-B FPDs

Perform the following steps to upgrade timing IC-A and Timing IC-B FPDs:

- Upgrade Timing IC-A FPD.

```
Router#upgrade hw-module location 0/[slot-number] fpd TimingIC-A
```

- Upgrade TimingIC-B FPD.

```
Router#upgrade hw-module location 0/[slot-number] fpd TimingIC-B
```

- Run the new XR using the **install commit** command, if you're performing this manual FPD upgrade.

```
Router(admin)#install commit
```

If you don't perform the install commit of the new XR, the LC reinstalls itself with this new XR again which could take 30 minutes.

- Reload the 5th Generation ASR9000 Line Card.

```
Router#admin
sysadmin-vm:0_RP0#hw-module location 0/[slot-number] reload
```

YANG Data Model for Field Programmable Device

Table 18: Feature History Table

Feature Name	Release Information	Description
Unified Model for FPD: Cisco-IOS-XR-um-fpd-cfg	Release 7.7.1	We have introduced the Cisco-IOS-XR-um-fpd-cfg unified model to enable or disable the automatic reload and automatic upgrade of Field Programmable Devices. You can access this unified model from the Github repository.

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

Operational Data	Native Data Model	CLI Commands
Auto Upgrade: Enabling or disabling of automatic upgrade of FPD.	Cisco-IOS-XR-fpd-infra-cfg.yang	• **fpd auto-upgrade enable** • **fpd auto-upgrade disable**
Auto Reload: Enabling or disabling of automatic reload of FPD.	Cisco-IOS-XR-fpd-infra-cfg.yang	• **fpd auto-reload enable** • **fpd auto-reload disable**

You can access the data models from the Github repository. To learn more about the data models and put them to use, see the *Programmability Configuration Guide for Cisco ASR 9000 Series Routers*.

How to Upgrade FPD Images

You must determine if an FPD image upgrade is needed using the **show hw-module fpd** command and perform the upgrade, if needed, under the following circumstances:

- Migrate the software to a later Cisco IOS XR software release.
- Swap line cards from a system running a different Cisco IOS XR software release.
- Insert a new line card.

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.

Procedure

	Command or Action	Purpose

Step 1	**show hw-module location** {**all** | *node-id*} **fpd**
	Example:
	`RP/0/RSP0/CPU0:router# show hw-module location all fpd`
	or
	`RP/0/RSP0/CPU0:router# show hw-module location 0/4/cpu0 fpd`
		Displays the current FPD image versions for the specified card or all cards installed in the router. Use this command to determine if you must upgrade the FPD image on your card.

Step 2	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 3	(Optional) **show fpd package** **Example:** `RP/0/RSP0/CPU0:router(admin)# show fpd package`	Displays which cards are supported with your current Cisco IOS XR software release, which FPD image you need for each card, and what the minimum hardware requirements are for the various modules. (A minimum hardware requirement version of 0.0 indicates that all hardware can support this FPD image version.) If there are multiple FPD images for your card, use this command to determine which FPD image to use if you want to upgrade only a specific FPD type.

Step 4	**upgrade hw-module fpd** {**all** | *fpga-type*} [**force**] **location** [**all** | *node-id*]
	Example:
	`RP/0/RSP0/CPU0:router(admin)# upgrade hw-module fpd all location 0/3/1`
	`.`
	`.`
	`.`
	`Successfully upgraded 1 FPD for SPA-2XOC48POS/RPR on location 0/3/1`
		Upgrades all the current FPD images that must be upgraded on the specified card with new images.
		Before continuing to the next step, wait for confirmation that the FPD upgrade has successfully completed. Status messages, similar to these, are displayed to the screen until the FPD upgrade is completed:
		`FPD upgrade started.`
		`FPD upgrade in progress..`
		`FPD upgrade in progress..`
		`FPD upgrade sent to location xxxx`
		`FPD upgrade sent to location yyyy`
		`FPD upgrade in progress..`
		`FPD upgrade finished for location xxx`
		`FPD upgrade in progress..`
		`FPD upgrade finished for location yyyy`
		`FPD upgrade completed.`
		The “FPD upgrade in progress.” message is printed every minute. These logs are information logs, and as such, are displayed if the **logging console informational** command is configured.

	Command or Action	Purpose

		If Ctrl-C is pressed while the FPD upgrade is in progress, the following warning message is displayed:
		
		`FPD upgrade in progress on some hardware, aborting now is not recommended as it might cause HW programming failure and result in RMA of the hardware.`
		`Do you want to continue? [Confirm(y/n)]`
		
		If you confirm that you want to abort the FPD upgrade procedure, this message is displayed:
		
		`FPD upgrade process has been aborted, please check the status of the hardware and reissue the upgrade command if required.`
		
		Note
		• If your card supports multiple FPD images, you can use the **show fpd package** admin command to determine what specific image to upgrade in the **upgrade hw-module fpd** command.
		• A message is displayed when router modules cannot get upgraded during upgrade with **location all** option indicating that the FPGA is intentionally skipped during upgrade. To upgrade such FPGAs, you can use the CLI command with a particular location explicitly specified. For example, **upgrade hw-module fpd all location 0/3/1**.
		• It is recommended to upgrade all FPGAs on a given node using the **upgrade hw-module fpd all location** {**all** | *node-id*} command. Do not upgrade the FPGA on a node using the **upgrade hw-module fpd** <*individual-fpd*> **location** {**all** | *node-id*} as it may cause errors in booting the card.

Step 5	**exit** **Example:** `sysadmin-vm:0_RP0# exit`	
Step 6	**hw-module location**{ *node-id* | **all** } **reload**	Use the **hw-module location reload** command to reload a line card. `sysadmin-vm:0_RP0# hw-module location 0/3 reload`
Step 7	**exit**	
Step 8	**show hw-module fpd**	Verifies that the FPD image on the card has been successfully upgraded by displaying the status of all FPDs in the system.

Configuration Examples for FPD Image Upgrade

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

show hw-module fpd Command Output: Example

Use the **show hw-module fpd** to display the current version of FPD images on the SPAs, SIPs and other cards installed on your router.

This command can be used to identify information about FPDs on any card. If you enter the location of a line card that is not a SPA, the output displays information about any programmable devices on that line card.

The following example shows how to display FPD compatibility for all modules in the router:

```
RP/0/RSP0/CPU0:router# ios#show hw-module fpd
Tue Jan 22 13:56:55.082 UTC
                                                                      FPD Versions
                                                                      ================
Location   Card type        HWver FPD device       ATR Status   Running Programd
-----------------------------------------------------------------------------
0/RP0      NCS-55A2-MOD-S    0.3   MB-MIFPGA            CURRENT    0.19    0.19
0/RP0      NCS-55A2-MOD-S    0.3   Bootloader           CURRENT    1.10    1.10
0/RP0      NCS-55A2-MOD-S    0.3   CPU-IOFPGA           CURRENT    1.18    1.18
0/RP0      NCS-55A2-MOD-S    0.3   MB-IOFPGA            CURRENT    0.18    0.18
0/PM0      NC55-1200W-ACFW   1.0   LIT-PriMCU-ACFW      NEED UPGD  2.08    2.08
0/PM1      NC55-1200W-ACFW   1.0   LIT-PriMCU-ACFW      NEED UPGD  2.08    2.08
RP/0/RP0/CPU0:ios#.
```

Note After Release 5.3.x, Upg/Dng? will display Yes only for upgrade.

The following example shows the FPD for which upgrage will be skipped.

```
RP/0/RSP1/CPU0:router# show hw-module fpd location all

Mon Jun 29 05:38:50.332 PST

==================================== ========================================
                                     Existing Field Programmable Devices
                                     ========================================
                                       HW                          Current SW Upg/
Location     Card Type               Version Type Subtype Inst   Version   Dng?
============ ======================= ======= ==== ======= ==== =========== ====
0/RSP0/CPU0  A9K-RSP-4G                4.8   lc   fpga3   0       1.13      No
                                             lc   fpga1   0       1.5       No
                                             lc   fpga2   0       1.14      No
                                             lc   cbc     0       1.2       No
                                             lc   fpga4   0       1.6       No
                                             lc   rommon  0       1.0       No
-----------------------------------------------------------------------------
0/RSP0/CPU0  ASR-9010-FAN              1.0   lc   cbc     1       4.0       No
-----------------------------------------------------------------------------
0/RSP0/CPU0  ASR-9010-FAN              1.0   lc   cbc     2       4.0       No
-----------------------------------------------------------------------------
```

```
0/1/CPU0      A9K-40GE-B                 1.0   lc    fpga1    0        0.38      No
                                               lc    fpga2    0        0.8       No
                                               lc    cbc      0        2.2       No
                                               lc    cpld1    0        0.15      No
                                               lc    rommon   0        1.0       No
-------------------------------------------------------------------------------------
0/1/CPU0      A9K-40GE-B                 1.0   lc    fpga1    1        0.38      No
-------------------------------------------------------------------------------------
0/4/CPU0      A9K-8T/4-B                 1.0   lc    fpga1    0        0.38      No
                                               lc    fpga2    0        0.10      No
                                               lc    cbc      0        2.2       No
                                               lc    cpld2    0        0.7       No
                                               lc    cpld1    0        0.15      No
                                               lc    cpld3    0        0.3       No
                                               lc    rommon   0        1.0       No
                                               lc    fpga3    0       14.42      No
-------------------------------------------------------------------------------------
0/4/CPU0      A9K-8T/4-B                 1.0   lc    fpga1    1        0.38      No
-------------------------------------------------------------------------------------
0/6/CPU0      A9K-4T-B                   1.0   lc    fpga1    0        0.38      No
                                               lc    fpga2    0        0.10      No
                                               lc    cbc      0        2.2       No
                                               lc    cpld2    0        0.7       No
                                               lc    cpld1    0        0.15      No
                                               lc    cpld3    0        0.3       No
                                               lc    rommon   0        1.0       No
                                               lc    fpga3    0       14.42      No
-------------------------------------------------------------------------------------
0/6/CPU0      A9K-4T-B                   1.0   lc    fpga1    1        0.38      No
-------------------------------------------------------------------------------------
```

The following example shows how to display FPD compatibility for a specific module in the router:

Table 19: show hw-module fpd Field Descriptions

Field	Description
Location	Location of the module in the *rack*/*slot*/*module* notation.
Card Type	Module part number.
HW Version	Hardware model version for the module.
Type	Hardware type. Can be one of the following types: • spa—Shared port adapter • lc—Line card
Subtype	FPD type. Can be one of the following types: • fabldr—Fabric downloader • fpga1—Field-programmable gate array • fpga2—Field-programmable gate array 2 • fpga3—Field-programmable gate array 3 • fpga4—Field-programmable gate array 4 • fpga5—Field-programmable gate array 5 • rommonA—Read-only memory monitor A • rommon—Read-only memory monitor B

Field	Description
Inst	FPD instance. The FPD instance uniquely identifies an FPD and is used by the FPD process to register an FPD.
Current SW Version	Currently running FPD image version.
Upg/Dng?	Specifies whether an FPD upgrade or downgrade is required. A downgrade is required in rare cases when the version of the FPD image has a higher major revision than the version of the FPD image in the current Cisco IOS XR software package.

show fpd package Command Output: Example

Use the **show fpd package** command in administration EXECAdmin EXEC mode mode to find out which line cards are supported with your current Cisco IOS XR software release, which FPD image package you need for each line card, 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 sample output from the **show fpd package** command:

```
show fpd package
Tue Jan 22 13:56:00.212 UTC

================================ ================================================
                                         Field Programmable Device Package
                                 ================================================
                                                Req     SW      Min Req   Min Req
Card Type            FPD Description            Reload  Ver     SW Ver  Board Ver
=================== ========================== ====== ======= ======== =========
--------------------------------------------------------------------------------
NC55-1200W-ACFW      LIT-PriMCU-ACFW(A)          NO      2.09    2.09    0.0
--------------------------------------------------------------------------------
NC55-900W-ACFW-I     LIT-PriMCU-ACFW-I(A)        NO      1.04    1.04    0.0
--------------------------------------------------------------------------------
NC55-900W-DCFW-I     LIT-PriMCU-DCFW-I(A)        NO      2.260   2.260   0.0
--------------------------------------------------------------------------------
NC55-930W-DCFW-C     LIT-PriMCU-DCFW-C(A)        NO      2.259   2.259   0.0
--------------------------------------------------------------------------------
NC55-MPA-12T-S       MPAFPGA                     YES     0.27    0.27    0.0
--------------------------------------------------------------------------------
NC55-MPA-1TH2H-S     -WDM-D-1HL_DCO_2            NO     38.518  38.518   0.1
                     MPAFPGA                     YES     0.53    0.53    0.0
                     WDM-DE-1HL_DCO_2            NO     38.518  38.518   0.1
                     WDM-DS-1HL_DCO_2            NO     38.268  38.268   0.1
--------------------------------------------------------------------------------
NC55-MPA-2TH-HX-S    -WDM-D-1HL_DCO_0            NO     38.518  38.518   0.1
                     -WDM-D-1HL_DCO_1            NO     38.518  38.518   0.1
                     MPAFPGA                     YES     0.53    0.53    0.0
```

```
                    WDM-DE-1HL_DCO_0          NO     38.518  38.518   0.1
                    WDM-DE-1HL_DCO_1          NO     38.518  38.518   0.1
                    WDM-DS-1HL_DCO_0          NO     38.268  38.268   0.1
                    WDM-DS-1HL_DCO_1          NO     38.268  38.268   0.1
-------------------------------------------------------------------------------
NC55-MPA-2TH-S      -WDM-D-1HL_DCO_0          NO     38.518  38.518   0.1
                    -WDM-D-1HL_DCO_1          NO     38.518  38.518   0.1
                    MPAFPGA                   YES     0.53    0.53    0.0
                    WDM-DE-1HL_DCO_0          NO     38.518  38.518   0.1
                    WDM-DE-1HL_DCO_1          NO     38.518  38.518   0.1
                    WDM-DS-1HL_DCO_0          NO     38.268  38.268   0.1
                    WDM-DS-1HL_DCO_1          NO     38.268  38.268   0.1
-------------------------------------------------------------------------------
NC55-MPA-4H-HD-S    MPAFPGA                   YES     0.53    0.53    0.0
-------------------------------------------------------------------------------
NC55-MPA-4H-HX-S    MPAFPGA                   YES     0.53    0.53    0.0
-------------------------------------------------------------------------------
NC55-MPA-4H-S       MPAFPGA                   YES     0.53    0.53    0.0
-------------------------------------------------------------------------------
NC55A2-MOD-SE-H-S   Bootloader(A)             YES     1.11    1.11    0.0
                    CPU-IOFPGA(A)             YES     1.18    1.18    0.1
                    MB-IOFPGA(A)              YES     0.18    0.18    0.1
                    MB-MIFPGA                 YES     0.19    0.19    0.0
                    SATA(A)                   NO      5.00    5.00    0.0
-------------------------------------------------------------------------------
NCS-55A2-MOD-HD-S   Bootloader(A)             YES     1.11    1.11    0.0
                    CPU-IOFPGA(A)             YES     1.18    1.18    0.1
                    MB-IOFPGA(A)              YES     0.18    0.18    0.1
                    MB-MIFPGA                 YES     0.19    0.19    0.0
                    SATA(A)                   NO      5.00    5.00    0.0
-------------------------------------------------------------------------------
NCS-55A2-MOD-HX-S   Bootloader(A)             YES     1.11    1.11    0.0
                    CPU-IOFPGA(A)             YES     1.18    1.18    0.1
                    MB-IOFPGA(A)              YES     0.18    0.18    0.1
                    MB-MIFPGA                 YES     0.19    0.19    0.0
                    SATA(A)                   NO      5.00    5.00    0.0
-------------------------------------------------------------------------------
NCS-55A2-MOD-S      Bootloader(A)             YES     1.11    1.11    0.0
                    CPU-IOFPGA(A)             YES     1.18    1.18    0.1
                    MB-IOFPGA(A)              YES     0.18    0.18    0.1
                    MB-MIFPGA                 YES     0.19    0.19    0.0
                    SATA(A)                   NO      5.00    5.00    0.0
-------------------------------------------------------------------------------
NCS-55A2-MOD-SE-S   Bootloader(A)             YES     1.11    1.11    0.0
                    CPU-IOFPGA(A)             YES     1.18    1.18    0.1
                    MB-IOFPGA(A)              YES     0.18    0.18    0.1
                    MB-MIFPGA                 YES     0.19    0.19    0.0
                    SATA(A)                   NO      5.00    5.00    0.0
                    STATSFPGA                 YES     0.01    0.01    0.0
```

This table describes the significant fields shown in the display:

Table 20: show fpd package Field Descriptions

Field	Description
Card Type	Module part number.
FPD Description	Description of all FPD images available for the line card.

Field	Description
Type	Hardware type. Possible types can be: • spa—Shared port adapter • lc—Line card
Subtype	FPD subtype. These values are used in the **upgrade hw-module fpd** command to indicate a specific FPD image type to upgrade.
SW Version	FPD software version recommended for the associated module running the current Cisco IOS XR software.
Min Req SW Vers	Minimum required FPD image software version to operate the card. Version 0.0 indicates that a minimum required image was not programmed into the card.
Min Req HW Vers	Minimum required hardware version for the associated FPD image. A minimum hardware requirement of version 0.0 indicates that all hardware can support this FPD image version.

Note In the **show fpd package** command output, the "subtype" column shows the FPDs that correspond with each line card image. To upgrade a specific FPD with the **upgrade hw-module fpd** command, replace the *fpga-type* argument with the appropriate FPD from the "subtype" column, as shown in the following example:

```
RP/0/RSP0/CPU0:router(admin)# upgrade hw-module fpd fpga2 location 0/3/1 reload
```

upgrade hw-module fpd Command Output: Example

Use the **upgrade hw-module fpd** command to upgrade the FPD image on a line card. 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/RSP0/CPU0:router# admin
RP/0/RSP0/CPU0:router(admin)# upgrade hw-module fpd fpga location 0/1/cpu0

 Mon Jan 12 05:44:37.611 PST

% RELOAD REMINDER:  - The upgrade operation of the target module will not interrupt its
normal
    operation. However, for the changes to take effect, the target module
    will need to be manually reloaded after the upgrade operation. This can
    be accomplished with the use of "hw-module <target> reload" command.
  - If automatic reload operation is desired after the upgrade, please use
    the "reload" option at the end of the upgrade command.
  - The output of "show hw-module fpd location" command will not display
```

```
    correct version information after the upgrade if the target module is
    not reloaded.
Continue? [confirm] y

Starting the upgrade/download of following FPD:

========== ==== ======= ======= =========== =========
                                  Current    Upg/Dng
Location   Type Subtype Upg/Dng   Version    Version
========== ==== ======= ======= =========== =========
0/1/CPU0   lc   fpga    upg         0.40         0.40
-----------------------------------------------------
LC/0/1/CPU0:Jan 12 05:44:43.700 : lc_fpd_upgrade[192]: %PLATFORM-UPGRADE_FPD-6-START :
  Starting to upgrade fpga subtype image from 0.4 to 0.4 for for this card  on location
0/1/CPU0
LC/0/1/CPU0:Jan 12 05:44:42.990 : fabricq_mgr[152]: EES:Internal clock detect IDLE
  period(-106461) more than threshold(1200000)
LC/0/1/CPU0:Jan 12 05:44:42.990 : ingressq[179]: EES:Internal clock detect IDLE
  period(-106461) more than threshold(1200000)
LC/0/1/CPU0:Jan 12 05:45:09.240 : fabricq_mgr[152]: EES:Internal clock detect IDLE
  period(-105945) more than threshold(1200000)
LC/0/1/CPU0:Jan 12 05:45:09.241 : ingressq[179]: EES:Internal clock detect IDLE
  period(-105944) more than threshold(1200000)
SP/0/1/SP:Jan 12 05:45:16.020 : upgrade_daemon[280]: ...programming...
SP/0/1/SP:Jan 12 05:45:16.034 : upgrade_daemon[280]: ...it will take a while...
SP/0/1/SP:Jan 12 05:45:16.053 : upgrade_daemon[280]: ...it will take a while...
SP/0/1/SP:Jan 12 05:47:42.967 : upgrade_daemon[280]: ...programming...
SP/0/1/SP:Jan 12 05:47:42.981 : upgrade_daemon[280]: ...it will take a while...

% SLC/0/1/CPU0:Jan 12 05:48:08.737 : lc_fpd_upgrade[192]: %PLATFORM-UPGRADE_FPD-6-PASSED :

  Successfully upgrade fpga subtype image for for this card  on location 0/1/CPU0
```

show platform Command Output: Example

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

Troubleshooting Problems with FPD Image Upgrades

This section contains information to help troubleshoot problems that can occur during the upgrade process.

Power Failure or Removal of a SPA During an FPD Image Upgrade

If the FPD upgrade operation is interrupted by a power failure or the removal of the SPA, it could corrupt the FPD image. This corruption of the FPD image file makes the SPA unusable by the router and the system displays the following messages when it tries to power up the SPA. When it cannot successfully power up the SPA, it places it in the failed state, as shown in the following example:

```
LC/0/3/CPU0:Feb  4 08:23:16.672 : spa_192_jacket[188]: %L2-SPA-5-OIR_INSERTED : SPA discovered
 in bay 0
  LC/0/3/CPU0:Feb  4 08:23:23.349 : spa_192_jacket[188]: %L2-SPA-5-OIR_ERROR : SPA (0): An
 error occurred (0x1002), error recovery action: reset SPA
  LC/0/3/CPU0:Feb  4 08:23:26.431 : spa_192_jacket[188]: %L2-SPA-5-OIR_INSERTED : SPA
discovered in bay 0
  LC/0/3/CPU0:Feb  4 08:23:32.593 : spa_192_jacket[188]: %L2-SPA-5-OIR_ERROR : SPA (0): Too
 many retries, error recovery stopped
```

```
 LC/0/3/CPU0:Feb  4 08:23:32.593 : spa_192_jacket[188]: %L2-SPA-5-OIR_ERROR : SPA (0): An
error occurred (0x1002), error recovery action: hold SPA in reset
```

When a SPA is in the failed state, it may not register itself with the FPD upgrade mechanism. In this case, you do not see the SPA listed when you use the **show hw-module fpd** command. To verify the state of a SPA, use the **show hw-module subslot error** command and the **show hw-module subslot status** command.

Performing a SPA FPD Recovery Upgrade

To recover a SPA from the failed state because of a corrupted FPD image, you must manually shut down the SPA. Use the **hw-module subslot** *subslot-id* **shutdown** command in Global Configuration mode to administratively shutdown the SPA. After the SPA is shut down, you can use the **upgrade hw-module fpd** command in administration EXEC mode:

```
RP/0/RSP0/CPU0:router# admin
RP/0/RSP0/CPU0:router(admin)# upgrade hw-module fpd fpga location 0/3/0
```

Performing a SIP FPD Recovery Upgrade

If a SIP upgrade fails for whatever reason, do not reload the SIP. Try to perform the upgrade procedure again. You can perform the upgrade procedure multiple times, as long as you do not reload the SIP. The FPD upgrade procedure takes several minutes to complete; do not interrupt the procedure. If you reload the SIP when the FPD image is corrupted, the SIP malfunctions and you must contact Cisco technical support for assistance.

To recover a SIP from the failed state because of a corrupted FPD image, you must contact Cisco technical support.

To recover a SIP from the failed state because of a corrupted FPD image, you must turn off the automatic reset of the SIP card. Use the **hw-module reset auto disable** command in administration configuration mode, as shown in the following example:

```
RP/0/RSP0/CPU0:router(admin-config)# hw-module reset auto disable location 0/1/4
```