# CISCO

# System Setup and Software Installation Guide for Cisco NCS 5000 Series Routers, IOS XR Release 6.6.x

**First Published:** 2018-12-21

**Last Modified:** 2019-05-30

# CONTENTS

# Preface

This preface contains these sections:

- Changes to This Document, on page vii
- Obtaining Documentation and Submitting a Service Request, on page vii

## Changes to This Document

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

This table lists the technical changes made to this document since it was first released.

| Date | Summary |
|------|---------|
| December 2018 | Initial release of this document |
| April 2019 | Republished for 6.6.2. |

## Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation*, at: http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html.

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation as an RSS feed and delivers content directly to your desktop using a reader application. The RSS feeds are a free service.

# New and Changed Feature Information

This table summarizes the new and changed feature information for the *System Setup and Software Installation Guide for Cisco NCS 5000 Series Routers*.

- New and Changed System Setup Features, on page 1

## New and Changed System Setup Features

| Feature | Description | Changed in Release | Where Documented |
|---------|-------------|--------------------|------------------|
| None | No new features introduced | Not applicable | Not applicable |

**C H A P T E R** **2**

# Cisco NCS 5000 Series Product Overview

### Cisco NCS 5001 Overview

Cisco NCS 5001 is a dense 10/100 Gigabit Ethernet Router in 1 RU form factor. It is designed for service provider access and aggregation network. The Cisco NCS 5001 runs the industry-leading Cisco IOS XR Software operating system, with robust features and functions such as application hosting, machine to machine interface, telemetry, and flexible package delivery.

NCS 5001 contains the following ports:

- 40 x 10G SFP+ Ports:

    - 16 x Regular 10G SFP+ Ports

    - 24 x DWDM & ZR Capable 10G SFP+ Ports

- 4 x 100G QSFP28 Ports

#### Features

The Cisco NCS 5001 router has the following features:

- 10Gbps bandwidth for each of the 40 fixed SFP+ ports

- Four QSPF ports capable of providing 100Gbps bandwidth

- Two 1+1 redundant, hot-swappable power supplies, which provide port side intake or exhaust for cooling

- Two N+1 redundant, hot-swappable fan modules, which provide port side intake or exhaust for cooling

- A management console and USB interface on the fan side of the router

### Cisco NCS 5002 Overview

Cisco NCS 5002 is a dense 10/100 Gigabit Ethernet Router in 2RU form factor. It is designed for service provider access and aggregation network. The Cisco NCS 5002 runs the industry-leading Cisco IOS XR Software operating system, with robust features and functions such as application hosting, machine to machine interface, telemetry, and flexible package delivery.

NCS 5002 contains the following ports:

- 80 x 10G SFP+ Ports:

    - 40 x Regular 10G SFP+ Ports

- 40 x DWDM & ZR Capable 10G SFP+ Ports

- 4 x 100G QSFP28 Ports

### Features

The Cisco NCS 5002 router has the following features:

- 10Gbps bandwidth for each of the 80 fixed SFP+ ports

- Four QSPF ports capable of providing 100Gbps bandwidth

- Two 1+1 redundant, hot-swappable power supplies, which provide port side intake or exhaust for cooling

- Two N+1 redundant, hot-swappable fan modules, which provide port side intake or exhaust for cooling

- A management console and USB interface on the fan side of the router

# Cisco NCS 5000 Series Product Overview

### Cisco NCS 5001 Overview

Cisco NCS 5001 is a dense 10/100 Gigabit Ethernet Router in 1 RU form factor. It is designed for service provider access and aggregation network. The Cisco NCS 5001 runs the industry-leading Cisco IOS XR Software operating system, with robust features and functions such as application hosting, machine to machine interface, telemetry, and flexible package delivery.

NCS 5001 contains the following ports:

- 40 x 10G SFP+ Ports:

    - 16 x Regular 10G SFP+ Ports

    - 24 x DWDM & ZR Capable 10G SFP+ Ports

- 4 x 100G QSFP28 Ports

### Features

The Cisco NCS 5001 router has the following features:

- 10Gbps bandwidth for each of the 40 fixed SFP+ ports

- Four QSPF ports capable of providing 100Gbps bandwidth

- Two 1+1 redundant, hot-swappable power supplies, which provide port side intake or exhaust for cooling

- Two N+1 redundant, hot-swappable fan modules, which provide port side intake or exhaust for cooling

- A management console and USB interface on the fan side of the router

### Cisco NCS 5002 Overview

Cisco NCS 5002 is a dense 10/100 Gigabit Ethernet Router in 2RU form factor. It is designed for service provider access and aggregation network. The Cisco NCS 5002 runs the industry-leading Cisco IOS XR Software operating system, with robust features and functions such as application hosting, machine to machine interface, telemetry, and flexible package delivery.

NCS 5002 contains the following ports:

- 80 x 10G SFP+ Ports:

    - 40 x Regular 10G SFP+ Ports

    - 40 x DWDM & ZR Capable 10G SFP+ Ports

- 4 x 100G QSFP28 Ports

### Features

The Cisco NCS 5002 router has the following features:

- 10Gbps bandwidth for each of the 80 fixed SFP+ ports

- Four QSPF ports capable of providing 100Gbps bandwidth

- Two 1+1 redundant, hot-swappable power supplies, which provide port side intake or exhaust for cooling

- Two N+1 redundant, hot-swappable fan modules, which provide port side intake or exhaust for cooling

- A management console and USB interface on the fan side of the router

# Command Modes

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

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

| Command Mode | Description |
|---|---|
| XR EXEC mode<br>(XR LXC execution mode) | Run commands on the XR LXC to display the operational state of the router.<br>Example:<br>`RP/0/RP0/CPU0:router#` |
| XR Config mode<br>(XR LXC configuration mode) | Perform security, routing, and other XR feature configurations on the XR LXC.<br>Example:<br>`RP/0/RP0/CPU0:router#configure`<br>`RP/0/RP0/CPU0:router(config)#` |

| Command Mode | Description |
|---|---|
| System Admin EXEC mode<br><br>(System Admin LXC execution mode) | Run commands on the System Admin LXC to display and monitor the operational state of the router hardware. The chassis or individual hardware modules can be reloaded from this mode.<br><br>Example:<br><br>`RP/0/RP0/CPU0:router#`**`admin`**<br>`sysadmin-vm:0_RP0#` |
| System Admin Config mode<br><br>(System Admin LXCconfiguration mode) | Run configuration commands on the System Admin LXC to manage and operate the hardware modules of the entire chassis.<br><br>Example:<br><br>`RP/0/RP0/CPU0:router#`**`admin`**<br>`sysadmin-vm:0_RP0#`**`config`**<br>`sysadmin-vm:0_RP0(config)#` |

**C H A P T E R 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.

# 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.

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

**Step 2**   Start the terminal emulation program on your workstation.

- For modular chassis RP, the console settings are baud rate 9600 bps, no parity, 2 stop bits and 8 data bits

- For fixed chassis, the console settings are baud rate 115200 bps, no parity, 2 stop bits and 8 data bits.

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

For NCS5001 and 5002 systems, the baud rate is 115200 bps, no parity, 2 stop bits and 8 data bits. For NCS5011 system, the console settings are baud rate 9600 bps, no parity, 2 stop bits and 8 data bits.

**Step 3**   Power on the router.

Connect the power cord to Power Entry Module (PEM) and 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 from source location to the server from where router boots up with new version. This ensures that if `md5sum` mismatch is observed, you can remove the corrupted file and ensure that a working copy of the image file is available for setup to begin.

**What to do next**

Specify the root username and password.

# Setup Root User Credentials

When the router boots for the first time, the system prompts the user to configure root credentials (username and password). These credentials are configured as the root user on the XR (root-lr) console, the System Admin VM (root-system), and as disaster-recovery credentials.

**Before you begin**

The boot process must be complete. For details on how to initiate the boot process, see .

**SUMMARY STEPS**

1. **Enter root-system username:** *username*
2. **Enter secret:** *password*
3. **Enter secret again:** *password*
4. **Username:** *username*
5. **Password:** *password*
6. (Optional) **show run username**

**DETAILED STEPS**

**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 a reimage, 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 username and password must be safeguarded as it has the superuser privileges. It is used to access the complete router configuration.

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

Reenter the password for the root user. The password is not accepted if it does not match the password that is entered in the previous step. 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 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/
!
```

**Note**    The NCS 5000 series routers running IOS XR 64-bit OS can operate as a standalone device, ZTP-controlled device or as an nV satellite.

When the router ships from the factory, the mode in which the router must operate is not predefined. Therefore, the software scans for a few events based on the usage, post-rack mounting, and power up, before deciding the mode of operation. Now, there is a time window when the software is making this decision. During this duration, the router that is intended to operate in standalone or ZTP modes, could be compromised to fall into the nV satellite mode. Thereby, opening up privileged control of the router to a hostile external entity.

Ensure that the external entity has access to the same network as the autoplay ports (highest 10G and lowest 100G ports) in order to gain control. Once compromised, the router could become inaccessible to legitimate users but can be recovered by physical disconnection to the network and reset to factory defaults.

For deployments within insecure or public networks, it is recommended to explicitly change the operating mode of NCS 5000 series router to the standalone mode using the **set sdac system-mode standalone** command in EXEC mode. This will be a one-time staging step for the first boot after unboxing, or after factory reset of the router before it is connected to an insecure network. Especially, if the links connecting to the router on the lowest 100G and the highest 10G ports are not known to be secure.

If you want to change the standalone mode to the satellite mode, use the **set sdac system-mode satellite** command in EXEC mode, and reload the router.

**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 10.

# Access the System Admin Console

You must login to the System Admin console through the XR console to perform all system administration and hardware management setups.

## SUMMARY STEPS

1. Login to the XR console as the root user.
2. (Optional) Disable the login banner on console port when accessing the System Admin mode from XR mode.
3. **admin**
4. (Optional) **exit**

## DETAILED STEPS

**Step 1**  Login to the XR console as the root user.

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

a) **configure**

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

**Example:**

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

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

c) **commit**

d) **end**

**Step 3**  **admin**

**Example:**

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

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

Mon May 22 06:57:29.350 UTC

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

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

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

**Step 4**  (Optional) **exit**

Return to the XR mode from the System Admin mode.

# Configure the Management Port

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

### Before you begin

- Consult your network administrator or system planner to procure IP addresses and a subnet mask for the management interface.

- Physical port Ethernet 0 and Ethernet 1 on RP are the management ports. Ensure that the port is connected to management network.

**SUMMARY STEPS**

1. **configure**
2. **interface MgmtEth** *rack/slot/port*
3. **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**

**Step 1**    **configure**

**Example:**

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

Enters mode.

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

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

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

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

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

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

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

**Example:**

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

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

**Step 5**     **no shutdown**

**Example:**

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

Places the interface in an "up" state.

**Step 6**     **exit**

**Example:**

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

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

**Step 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.

## SUMMARY STEPS

1. **configure**
2. **ntp  server**  *server_address*

## DETAILED STEPS

**Step 1**  **configure**

**Example:**

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

Enters mode.

**Step 2**  **ntp  server**  *server_address*

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

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

**CHAPTER 4**

# Perform Preliminary Checks

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

## Verify Software Version

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

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

**SUMMARY STEPS**

1. **show version**

**DETAILED STEPS**

**show version**

**Example:**

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

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

**Example**

**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 .

# 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.

**SUMMARY STEPS**

1. **show hw-module fpd**

**DETAILED STEPS**

**show hw-module fpd**

**Example:**

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

Displays the list of hardware modules detected on the router.

```
FPD Versions
=================
Location Card type HWver FPD device ATR Status Running Programd
-------------------------------------------------------------
0/RP0    NCS5002   3.0   DB-MIFPGA      CURRENT  0.13   0.13
0/RP0    NCS5002   3.0   MB-MIFPGA      CURRENT  0.13   0.13
0/RP0    NCS5002   3.0   BIOS           CURRENT  1.07   1.07
0/RP0    NCS5002   3.0   IOFPGA         CURRENT  0.16   0.16
```

# 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**

---

**show hw-module fpd**

**Example:**

```
RP/0/RP0/CPU0:router# show hw-module fpd
FPD Versions
=================
Location Card type HWver FPD device    ATR Status Running Programd
------------------------------------------------------------------------------
0/RP0    NCS5002   3.0   DB-MIFPGA      CURRENT    0.13    0.13
0/RP0    NCS5002   3.0   MB-MIFPGA      CURRENT    0.13    0.13
0/RP0    NCS5002   3.0   BIOS           CURRENT    1.07    1.07
0/RP0    NCS5002   3.0   IOFPGA         CURRENT    0.16    0.16
```

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 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**

---

**show ipv4 interface summary**

**Example:**

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

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

```
IP address      State    State        State         State
config          up,up    up,down      down,down     shutdown,down
-------------------------------------------------------------------
Assigned        0        0            0             0
Unnumbered      0        0            0             0
Unassigned      0        0            0             84
```

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.

&bull; Unassigned—No IP address is assigned to the interface.

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

**CHAPTER 5**

# Create User Profiles and Assign Privileges

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

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

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

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

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

- If there is a first user, no syncing occurs.

- If there is no first user, then the first user on XR (based on the order of creation) is synced to System Admin VM.

- When a user is added in XR, if there is no user on System Admin mode, then the user is synced to sysadmin-vm. After the synchronization, any changes to the user on XR VM does not synchronize on the System Admin VM.

- A user added on the System Admin VM does not synchronize with XR VM.

- Only the first user or disaster-recovery user created on System Admin VM synchronizes with the host VM.

- Changes to credentials of first user or disaster-recovery user on System Admin VM synchronizes with the host VM.

- The first user or disaster-recovery user deleted on System Admin VM does not synchronize with the host VM. The host VM retains the user.

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

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

**Figure 1: Workflow for Creating User Profiles**



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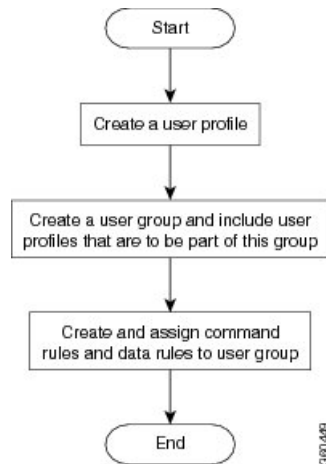
**Note**    The root-lr user, created for the XR VM during initial router start-up, is mapped to the root-system user for the System Admin VM. The root-system user has superuser permissions for the System Admin VM and therefore has no access restrictions.

---

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

The topics covered in this chapter are:

# Create User Groups

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

For extensive information about creating user groups, task groups, RADIUS and TACACS configurations, see the *Configuring AAA Services* chapter in the *System Security Configuration Guide for Cisco NCS 5000 Series Routers*. For detailed information about commands, syntax and their description, see the *Authentication, Authorization, and Accounting Commands* chapter in the *System Security Command Reference for Cisco NCS 5000 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.

## SUMMARY STEPS

1.  **configure**
2.  **usergroup** *usergroup-name*
3.  **description** *string*
4.  **inherit usergroup** *usergroup-name*
5.  **taskgroup** *taskgroup-name*
6.  Repeat Step for each task group to be associated with the user group named in Step 2.
7.  Use the **commit** or **end** command.

## DETAILED STEPS

**Step 1**  **configure**

**Example:**

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

Enters mode.

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

**Example:**

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

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

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

**Step 3**  **description** *string*

**Example:**

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

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

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

**Example:**

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

• Explicitly defines permissions for the user group.

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

**Example:**

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

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

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

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

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

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

**end** —Prompts user to take one of these actions:

• **Yes** — Saves configuration changes and exits the configuration session.

• **No** —Exits the configuration session without committing the configuration changes.

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

# Create a User Group in System Admin VM

Create a user group for the System Admin VM.

The router supports a maximum of 32 user groups.

**Before you begin**

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

**SUMMARY STEPS**

1. **admin**
2. **config**
3. **aaa authentication groups group** *group_name*
4. **users** *user_name*
5. **gid** *group_id_value*
6. Use the **commit** or **end** command.

**DETAILED STEPS**

**Step 1**      **admin**

**Example:**

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

Enters mode.

**Step 2**     **config**

**Example:**

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

Enters System Admin Config mode.

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

**Example:**

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

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

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

**Step 4**     **users** *user_name*

**Example:**

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

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

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

**Step 5**     **gid** *group_id_value*

**Example:**

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

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

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

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

**end** —Prompts user to take one of these actions:

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

**What to do next**

- Create command rules. See Create Command Rules, on page 30.
- Create data rules. See Create Data Rules, on page 32.

# Create Users

Create new users for the XR VM and System Admin VM.

**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**

When the user profile is created for the first time in XR VM, the user name and password are synced to the System Admin VM if no user already exists in System Admin VM.

However, the subsequent password change or user deletion in XR VM for the synced user is not synchronized with the System Admin VM.

Therefore, the passwords in XR VM and System Admin VM may not be the same. Also, the user synced with the System Admin VM will not be deleted if the user is deleted in XR VM.

For extensive information about creating user groups, task groups, RADIUS and TACACS configurations, see the *Configuring AAA Services* chapter in the *System Security Configuration Guide for Cisco NCS 5000 Series Routers*. For detailed information about commands, syntax and their description, see the *Authentication, Authorization, and Accounting Commands* chapter in the *System Security Command Reference for Cisco NCS 5000 Series Routers*.

# Create a User Profile in XR VM

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

For more information about AAA, and creating users, see the *Configuring AAA Services* chapter in the *System Security Configuration Guide for Cisco NCS 5000 Series Routers*. For detailed information about related commands, syntax and their description, see the *Authentication, Authorization, and Accounting Commands* chapter in the *System Security Command Reference for Cisco NCS 5000 Series Routers*.

**Step 1**   **configure**

**Example:**

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

Enters mode.

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

**Example:**

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

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

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

**Step 3**     Do one of the following:

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

**Example:**

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

or

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

Specifies a password for the user named in Step 2.

- Use the **secret** command to create a secure login password for the user names specified in Step 2.

- Entering **0** following the **password** command specifies that an unencrypted (clear-text) password follows. Entering **7** following the **password** command specifies that an encrypted password follows.

- For the **secret** command, the following values can be entered:

  - **0** : specifies that a secure unencrypted (clear-text) password follows

  - **5** : specifies that a secure encrypted password follows that uses MD5 hashing algorithm

  - **8** : specifies that Type 8 secret that uses SHA256 hashing algorithm follows

  - **9** : specifies that Type 9 secret that uses SCrypt hashing algorithm follows

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

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

  **Note**          • Type 10 secret is supported only for Cisco IOS XR 64 bit platform.

  - Backward compatibility issues such as configuration loss, authentication failure, and so on, are expected when you downgrade to lower versions that still use **MD5** or **SHA256** encryption algorithms. If there are any type 10 secrets, convert the **secrets** to type 5 if you are downgrading the system from versions 7.0.1 and above to versions 6.5.3 and above. If you are downgrading the system from versions 7.0.1 and above to versions below 6.5.3, then un-configure all users from the XR-vm and sysadmin-vm before executing install activate.

  - In a first user configuration scenario or when you reconfigure a user, the system synchronises only the Type 5 and Type 10 secrets from XR VM to System Admin VM and Host VM. It does not synchronize the Type 8 and Type 9 secrets in such scenarios.

- Type **0** is the default for the **password** and **secret** commands.

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

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

**Example:**

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

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

- The user takes on all attributes of the user group, as defined by that user group's association to various task groups.

- Each user must be assigned to at least one user group. A user may belong to multiple user groups.

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

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

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

**end** —Prompts user to take one of these actions:

- **Yes** — Saves configuration changes and exits the configuration session.

- **No** —Exits the configuration session without committing the configuration changes.

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

# Create a User Profile in System Admin VM

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

The router supports a maximum of 1024 user profiles.

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

## SUMMARY STEPS

1. **admin**
2. **config**
3. **aaa authentication users user** *user_name*
4. **password** *password*
5. **uid** *user_id_value*
6. **gid** *group_id_value*
7. **ssh_keydir** *ssh_keydir*
8. **homedir** *homedir*
9. Use the **commit** or **end** command.

## DETAILED STEPS

**Step 1**     **admin**

**Example:**

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

Enters mode.

**Step 2**     **config**

**Example:**

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

Enters System Admin Config mode.

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

**Example:**

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

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

**Step 4**     **password** *password*

**Example:**

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

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

**Step 5**     **uid** *user_id_value*

**Example:**

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

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

**Step 6**     **gid** *group_id_value*

**Example:**

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

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

**Step 7**     **ssh_keydir** *ssh_keydir*

**Example:**

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

Specify any alphanumeric value.

**Step 8**     **homedir** *homedir*

**Example:**

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

Specify any alphanumeric value.

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

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

**end** —Prompts user to take one of these actions:

- **Yes** — Saves configuration changes and exits the configuration session.

- **No** —Exits the configuration session without committing the configuration changes.

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

---

**What to do next**

- Create user group that includes the user created in this task. See .

- Create command rules that apply to the user group. See .

- Create data rules that apply to the user group. See .

# 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 .

**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

**Step 1** **admin**

**Example:**

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

Enters mode.

**Step 2** **config**

**Example:**

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

Enters System Admin Config mode.

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

**Example:**

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

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

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

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

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

**Step 4** **command** *command_name*

**Example:**

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

Specify the command for which permission is to be controlled.

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

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

**Example:**

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

Specify the operation for which permission has to be specified:

- **r** — Read

- **x** — Execute

• **rx** — Read and execute

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

**Example:**

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

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

• **accept** — users are permitted to perform the operation

• **accept_log**— users are permitted to perform the operation and every access attempt is logged.

• **reject**— users are restricted from performing the operation.

**Step 7** **group** *user_group_name*

**Example:**

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

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

**Step 8** **context** *connection_type*

**Example:**

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

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

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

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

**end** —Prompts user to take one of these actions:

• **Yes** — Saves configuration changes and exits the configuration session.

• **No** —Exits the configuration session without committing the configuration changes.

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

**What to do next**

Create data rules. See .

# Create Data Rules

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 24.

## 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

**Step 1** **admin**

**Example:**

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

Enters mode.

**Step 2** **config**

**Example:**

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

Enters System Admin Config mode.

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

**Example:**

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

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

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

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

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

**Step 4** **keypath** *keypath*

**Example:**

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

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

**Step 5**       **ops** *operation*

**Example:**

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

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

- c—Create

- d—Delete

- u—Update

- w— Write (a combination of create, update, and delete)

- r—Read

- x—Execute

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

**Example:**

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

Specify whether users are permitted or denied the operation.

- **accept** — users are permitted to perform the operation

- **accept_log**— users are permitted to perform the operation and every access attempt is logged

- **reject**— users are restricted from performing the operation

**Step 7**       **group** *user_group_name*

**Example:**

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

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

**Step 8**       **context** *connection type*

**Example:**

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

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

**Step 9**       **namespace** *namespace*

**Example:**

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

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

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

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

end —Prompts user to take one of these actions:

- **Yes** — Saves configuration changes and exits the configuration session.

- **No** —Exits the configuration session without committing the configuration changes.

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

# Change Disaster-recovery Username and Password

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

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

- Access the system when the AAA database, which is the default source for authentication in System Admin console is corrupted.

- Access the system through the management port, when, for some reason, the System Admin console is not working.

- Create new users by accessing the System Admin console using the disaster-recovery username and password, when the regular username and password is forgotten.

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

**SUMMARY STEPS**

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

**DETAILED STEPS**

**Step 1**    **admin**

**Example:**

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

Enters mode.

**Step 2** **config**

**Example:**

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

Enters System Admin Config mode.

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

**Example:**

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

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

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

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

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

**end** —Prompts user to take one of these actions:

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

# Recover Password using PXE Boot

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

**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 78.

**Step 2** Reset the password.

**C H A P T E R  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

**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.

System upgrade is done by installing a base package–Cisco IOS XR Unicast Routing Core Bundle. The file name for this bundle is *ncs5k-mini-x.iso*. Install this ISO image using **install** commands. For more information about the install process, see Workflow for Install Process, on page 39.

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

Do not reload the router during an upgrade operation.

Cisco IOS XR supports RPM signing and signature verification for Cisco IOS XR RPM packages in the ISO and upgrade images. All RPM packages in the Cisco IOS XR ISO and upgrade images are signed to ensure cryptographic integrity and authenticity. This guarantees that the RPM packages have not been tampered with

and the RPM packages are from Cisco IOS XR. The private key, used for signing the RPM packages, is created and securely maintained by Cisco.

For more information on upgrading the system and the RPMs, see *Manage Automatic Dependency* chapter.

# Upgrading Features

Upgrading features is the process of deploying new features and software patches on the router. Feature upgrade is done by installing package files, termed simply, packages. Software patch installation is done by installing Software Maintenance Upgrade (SMU) files.

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

For example, the components of the routing package are split into individual RPMs, such as BGP and OSPF. BGP is a mandatory RPM which is a part of the base software version and hence cannot be removed. Optional RPMs such as OSPF can be added and removed as required.

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

| Package | Requirement | Example |
|---|---|---|
| BGP | Mandatory | ncs5k-bgp-1.0.0.0-*<release-number>*.x86_64.rpm |
| NCS5K RM | Mandatory | ncs5k-rm-1.0.0.0-*<release-number>*.x86_64.rpm |
| NCS 5K Forwarding | Mandatory | ncs5k-fwding-1.0.0.0-*<release-number>*.x86_64.rpm |
| ios-xr CE | Mandatory | ncs5k-iosxr-ce-1.0.0.0-*<release-number>*.x86_64.rpm |
| iosxr-fwding | Mandatory | ncs5k-iosxr-fwding-1.0.0.0-*<release-number>*.x86_64.rpm |
| iosxr-infra | Mandatory | ncs5k-iosxr-infra-1.0.0.0-*<release-number>*.x86_64.rpm |
| iosxr-infra-test | Optional | ncs5k-infra-test-1.0.0.0-*<release-number>*.x86_64.rpm |
| iosxr-mgbl | Optional | ncs5k-iosxr-mgbl-1.0.0.0-*<release-number>*.x86_64.rpm |
| iosxr-mpls | Optional | ncs5k-iosxr-mpls-1.0.0.0-*<release-number>*.x86_64.rpm |
| iosxr-os | Mandatory | ncs5k-iosxr-os-1.0.0.0-*<release-number>*.x86_64.rpm |
| iosxr-routing | Mandatory | ncs5k-iosxr-routing-1.0.0.0-*<release-number>*.x86_64.rpm |
| iosxr-security | Optional | ncs5k-k9sec-1.0.0.0-*<release-number>*.x86_64.rpm |
| os-support | Mandatory | ncs5k-os-support-1.0.0.0-*<release-number>*.x86_64.rpm |
| base | Mandatory | ncs5k-base-1.0.0.0-*<release-number>*.x86_64.rpm |
| mcast | Optional | ncs5k-mcast-1.0.0.0-*<release-number>*.x86_64.rpm |

Package and SMU installation is performed using **install** commands. For more information about the install process, see Install Packages, on page 39.

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

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

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

# Workflow for Install Process

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

For installing a package, see Install Packages, on page 39. For uninstalling a package, see Uninstall Packages, on page 46.

# 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**

- The System Admin package and XR package can be executed using **install** commands in the System Admin EXEC mode and XR EXEC mode. All **install** commands are applicable in both these modes.

- Install operation over IPv6 is not supported.

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

*Figure 2: Installing Packages Workflow*
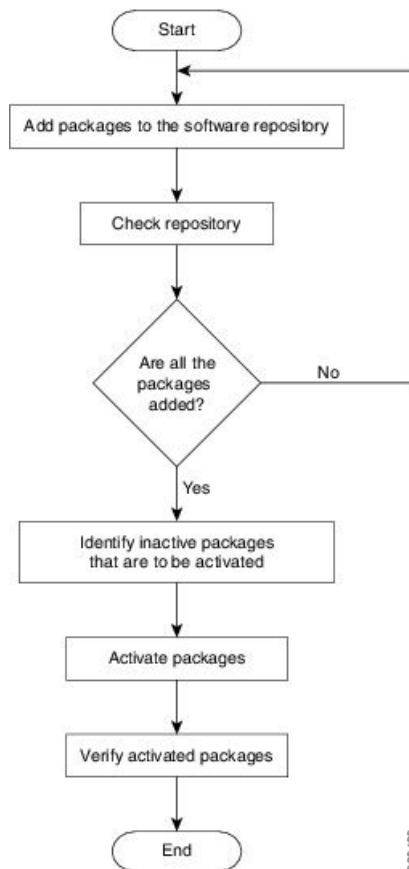


**Before you begin**

- Configure and connect to the management port. The installable file is accessed through the management port. For details about configuring the management port, see Configure the Management Port, on page 11.

- Copy the package to be installed either on the router's hard disk or on a network server to which the router has access.

**SUMMARY STEPS**

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 ...*

2. **show install request**
3. **show install repository**
4. **show install inactive**
5. Execute one of these:

> > - **install activate** *package_name*
> > - **install activate id** *operation_id*
>
> **6.** **show install active**
>
> **7.** **install commit**

**DETAILED STEPS**

---

**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 ...*

**Example:**

or

```
RP/0/RP0/CPU0:router#install add source sftp://root@8.33.5.15:/auto/ncs/package/
ncs5k-mcast-1.0.0.0-<release-number>.x86_64.rpm ncs5k-iosxr-mpls-1.0.0.0-<release-number>.x86_64.rpm
```

or

```
RP/0/RP0/CPU0:router#install add source sftp://root@8.33.5.15:/auto/ncs/package/
ncs5k-mcast-1.0.0.0-<release-number>.x86_64.rpm ncs5k-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 as soon as possible.

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

**Step 2**    **show install request**

**Example:**

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

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

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

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

**Step 3**    **show install repository**

**Example:**

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

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

**Step 4**    **show install inactive**

**Example:**

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

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

**Step 5**    Execute one of these:

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

**Example:**

```
RP/0/RP0/CPU0:router#install activate ncs5k-mcast-1.0.0.0-<release-number>.x86_64.rpm
ncs5k-iosxr-mpls-1.0.0.0-<release-number>.x86_64.rpm
```

or

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

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

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

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

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

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

**Step 6**    **show install active**

**Example:**

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

Displays packages that are active.

```
RP/0/RP0/CPU0:skywarp-tb#show install active
Tue Dec 22 16:02:46.873 UTC
Node 0/RP0/CPU0 [RP]
  Boot Partition: xr_lv55
  Active Packages: 2
        ncs5k-xr-<release-number> version=<release-number> [Boot image]
        ncs5k-k9sec-1.0.0.0-<release-number>
```

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

**Step 7**    **install commit**

**Example:**

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

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

Note On Multi-SDR mode, you can use the **install commit sdr** to commit just the sdr from where the CLI is being triggered.

---

Installing Packages: Related Commands

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

**What to do next**

- Ensure that you commit the upgrade using **install commit**.

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

- Verify the installation using the **install verify packages** command.

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

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

# Install Prepared Packages

- 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 reduced.

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.

- Performs 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.

- Performs package compatibility check. This ensures that all the required installation packages are available. For any package compatibility check error, details of the package and version are logged.

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

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

## SUMMARY STEPS

1. Add the required ISO image and packages to the repository.
2. **show install repository**
3. Execute one of these:
   - **install prepare** *package_name*
   - **install prepare id** *operation_id*
4. **show install prepare**
5. **install activate**
6. **show install active**
7. **install commit**

## DETAILED STEPS

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

For details, see Install Packages, on page 39.

**Step 2**  **show install repository**

**Example:**

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

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

**Step 3**     Execute one of these:

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

**Example:**

```
RP/0/RP0/CPU0:router#install prepare ncs5k-mcast-1.0.0.0-r60014I.x86_64.rpm
```

or

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

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

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

**Step 4**     **show install prepare**

**Example:**

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

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

**Step 5**     **install activate**

**Example:**

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

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

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

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

**Step 6**     **show install active**

**Example:**

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

Displays packages that are active.

```
Node 0/RP0/CPU0 [RP]
  Boot Partition: xr_lv55
  Active Packages: 2
        ncs5k-xr-6.0.0 version=6.0.0 [Boot image]
        ncs5k-k9sec-1.0.0.0-r600
```

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

**Step 7**     **install commit**

**Example:**

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

**Installing Packages: Related Commands**

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

**What to do next**

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

- Verify the installation using the **install verify packages** command.

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

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

# Uninstall Packages

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

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

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

*Figure 3: Uninstalling Packages Workflow*
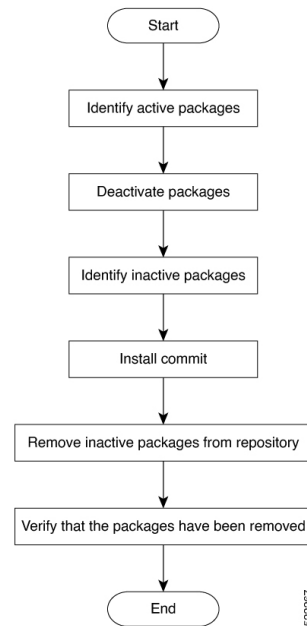


This task uninstalls XR VM packages. If you need to uninstall System Admin packages, run the same commands from the System Admin EXEC mode.

## SUMMARY STEPS

1. **show install active**
2. Execute one of these:

    • **install deactivate** *package_name*
    • **install deactivate id** *operation_id*

3. **show install inactive**
4. **install commit**
5. **install remove** *package_name*
6. **show install repository**

## DETAILED STEPS

**Step 1**     **show install active**

**Example:**

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

Displays active packages. Only active packages can be deactivated.

```
Node 0/RP0/CPU0 [RP]
  Boot Partition: xr_lv55
  Active Packages: 2
        ncs5k-xr-6.0.0 version=6.0.0 [Boot image]
```

```
ncs5k-k9sec-1.0.0.0-r600
```

**Step 2** Execute one of these:

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

**Example:**

```
RP/0/RP0/CPU0:router#install deactivate ncs5k-mcast-1.0.0.0-r60014I.x86_64.rpm
ncs5k-iosxr-mpls-1.0.0.0-r60014I.x86_64.rpm
```

or

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

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

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

**Step 3** **show install inactive**

**Example:**

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

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

**Step 4** **install commit**

**Step 5** **install remove** *package_name*

**Example:**

```
RP/0/RP0/CPU0:router#install remove ncs5k-mcast-1.0.0.0-r60014I.x86_64.rpm
ncs5k-iosxr-mpls-1.0.0.0-r60014I.x86_64.rpm
```

The inactive packages are removed from the repository.

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

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

**Step 6** **show install repository**

**Example:**

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

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

**What to do next**

Install required packages. .

**C H A P T E R 7**

# Manage Automatic Dependency

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

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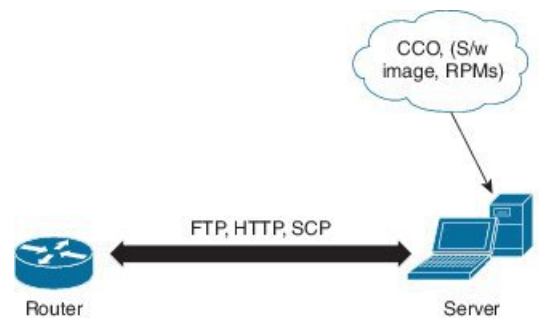


Until this release, you downloaded the software image and required RPMs from CCO on a network server (the repository), and used the **install add** and the **install activate** commands to add and activate the downloaded files on the router. Then, you manually identify relevant dependent RPMs, to add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

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

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

**Note**

1. Cisco IOS XR Version 6.0.2 and later does not provide third party and host package SMUs as part of automatic dependency management (**install source** command). The third party and host package SMUs must be installed separately, and in isolation from other installation procedures (installation of SMUs and RPMs in IOS XR or admin containers).

The rest of this chapter contains these sections:

# 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] ncs5k-mcast.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] ncs5k-mcast-1.0.0.1-r611.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] ncs5k-mcast-1.0.0.1-r611.CSCva85697.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 6.2.2**
  **ncs5k-mcast-1.0.0.0-r623.x86_64.rpm**

# Downgrade an RPM

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

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

  For example:

  - &lt;platform&gt;-sysadmin-hostos-6.5.1-r651.CSChu77777.host.arm

  - &lt;platform&gt;-sysadmin-hostos-6.5.1-r651.CSChu77777.admin.arm

  - &lt;platform&gt;-sysadmin-hostos-6.5.1-r651.CSChu77777.host.x86_64

  - &lt;platform&gt;-sysadmin-hostos-6.5.1-r651.CSChu77777.admin.x86_64

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

  For example:

  - &lt;platform&gt;-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 `ncs5000-sysadmin-hostos-6.5.1-r651.CSChu44444.host.arm` is the RPM installed, then `ncs5000-sysadmin-hostos-6.5.1-r651.host.arm` is its base RPM.

        2. Commit the configuration.

           ```
           install commit
           ```

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

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

- **Downgrade Non-Hostos RPM**

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

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

    2. Check the active version of the RPM.

       ```
       show install active
       ```

    3. Commit the configuration.

```
install commit
```

# CHAPTER **8**

# Customize Installation using Golden ISO

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

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

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

When a system boots with GISO, additional SMUs and RPMs in GISO are installed automatically, and the router is pre-configured with the XR configuration in GISO. For more information about downloading and installing GISO, see Install Golden ISO, on page 59.

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

- Initial deployment of the router

- Software disaster recovery

- System upgrade from one base version to another

- System upgrade from same base version but with additional SMUs

- Install update to identify and update dependant packages

## Limitations

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

- Building and booting GISO for asynchronous package (a package of different release than the ISO) is not supported.

• Verifying the XR configuration is not supported in the GISO build script `gisobuild.py`.

• Renaming a GISO build and then installing from the renamed GISO build is not supported.

• Install operation over IPv6 is not supported.

# Golden ISO Workflow

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

*Figure 5: Golden ISO Workflow*
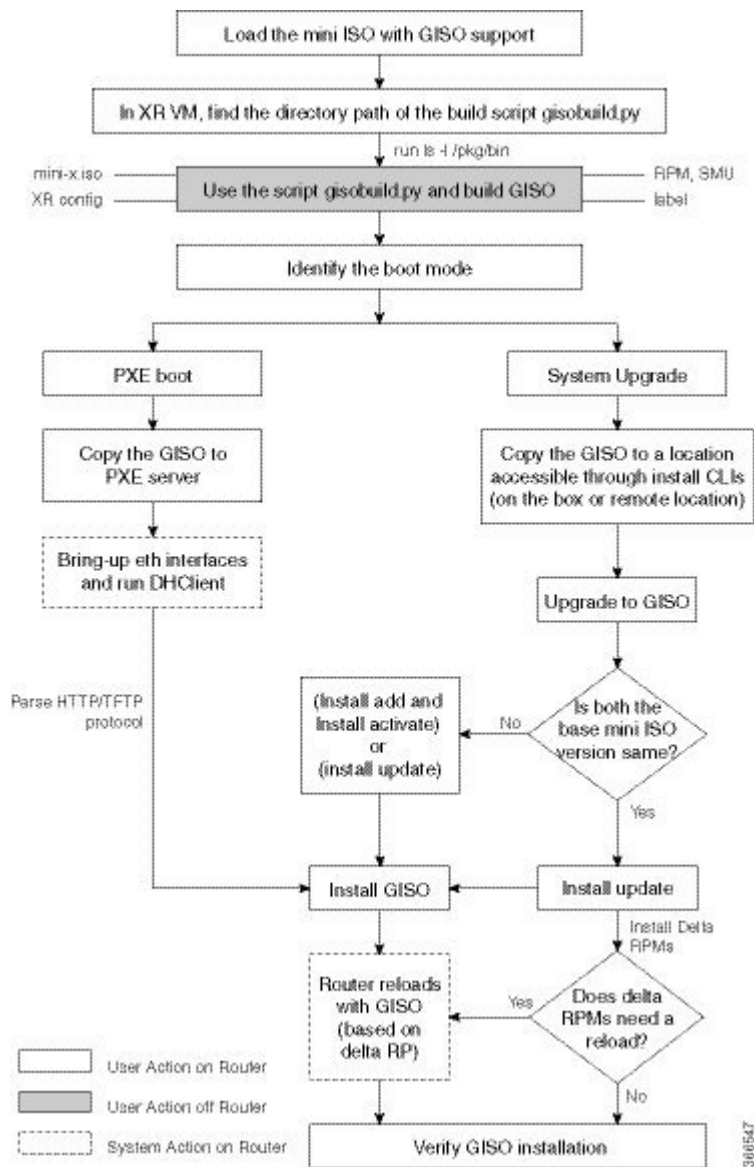
# Build Golden ISO

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

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

- Builds RPM database of all the packages present in package repository.

- Scans the repositories and selects the relevant Cisco RPMs that matches the input iso.

- Skips and removes third-party RPMs that are not SMUs of already existing third-party base package in mini-x.iso.

- Displays an error and exits build process if there are multiple base RPMs of same release but different versions.

- Performs compatibility check and dependency check for all the RPMs. For example, the child RPM ncs5000-mpls-te-rsvp is dependent on the parent RPM ncs5000-mpls . If only the child RPM is included, the Golden ISO build fails.

# Build Golden ISO Using Script

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

- Base mini-x.iso (mandatory)

- XR configuration file (optional)

- one or more Cisco-specific SMUs for host, XR and System admin (mandatory)

- one or more third-party SMUs for host, XR and System admin (mandatory)

- Label for golden ISO (optional)

- Optional RPMs

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

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

To build GISO, perform the following steps:

**Before you begin**

- To upgrade from a release that did not support GISO to a release supporting GISO version, it is mandatory to first upgrade to mini ISO with GISO support. For NCS 5001 series routers, upgrade to release 6.2.2 or later.

- The system where GISO is built must meet the following requirements:

    - System must have Python version 2.7 and later.

- System must have free disk space of minimum 3 to 4 GB.

- Verify that the Linux utilities `mount`, `rm`, `cp`, `umount`, `zcat`, `chroot`, `mkisofs` are present in the system. These utilities will be used by the script. Ensure privileges are available to execute all of these Linux commands.

- Kernel version of the system must be later than 3.16 or later than the version of kernel of Cisco ISO.

- Verify that a `libyaml` rpm supported by the Linux kernel is available to successfully `import yaml` in the tool.

- User should have proper permission for security rpm(k9sec-rpm) in rpm repository, else security rpm would be ignored for Golden ISO creation.

- The system from where the `gisobuild.py` script is executed must have root credentials.

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

**Step 2**   Run the script `gisobuild.py` and provide parameters to build the golden ISO off the router.

**Example:**

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

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

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

```
[directory-path]$ gisobuild.py [-h] [-i ncs5000-mini-x.iso] [-r <rpm repository>]
[-c <config-file>] [-l <giso label>] [-m] [-v]

System requirements check [PASS]

Platform: ncs5000 Version: <version>

Scanning repository [repository-path]...

Building RPM Database...
Total 3 RPM(s) present in the repository path provided in CLI

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

        (+) ncs5000-mgbl-3.0.0.0-r<version>.x86_64.rpm

        ...RPM compatibility check [PASS]

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

        (+) ncs5000-sysadmin-system-<version>-r<version>.CSCcv44444.x86_64.rpm

Following SYSADMIN arm rpm(s) will be used for building Golden ISO:

        (+) ncs5000-sysadmin-system-<version>-r<version>.CSCcv44444.arm.rpm

        ...RPM compatibility check [PASS]

Building Golden ISO...
```

```
Summary .....

XR rpms:
        ncs5000-mgbl-3.0.0.0-r<version>.x86_64.rpm

SYSADMIN rpms:
        ncs5000-sysadmin-system-<version>-r<version>.CSCcv44444.x86_64.rpm
        ncs5000-sysadmin-system-<version>-r<version>.CSCcv44444.arm.rpm

        ...Golden ISO creation SUCCESS.

Golden ISO Image Location: <directory-path>/ncs5000-golden-x.iso-<version>

Detail logs: <directory-path>/Giso_build.log-2017-07-10:18:17:43.395586
```

where:

- -i is the path to mini-x.iso

- -r is the path to RPM repository

- -c is the path to XR config file

- -l is the golden ISO label

- -h shows the help message

- -v is the version of the build tool `gisobuild.py`

- -m is to build the migration tar to migrate from IOS XR to IOS XR 64 bit

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

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

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

**What to do next**

Install the GISO image on the router.

# Install Golden ISO

Golden ISO (GISO) automatically performs the following actions:

- Installs host and system admin RPMs.

- Partitions repository and TFTP boot on RP.

- Creates software profile in system admin and XR modes.

- Installs XR RPMs. Use **show install active** command to see the list of RPMs.

• Applies XR configuration. Use **show running-config** command in XR mode to verify.

---

**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.
- **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.isso /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 update** command installs only the delta RPMs. System reload is based on restart type of the delta RPMs.

```
Router#install replace <path-to-image> <platform-name-golden-x-<version>-<label>.iso
Install operation 1 started by root:
exec-timeout is suspended.
No install operation in progress at this moment
Label = <label-name>
ISO <platform-name-golden-x-<version>.iso in input package list. Going to upgrade the system
 to version 6.5.2.

Current label: <None>

Updating contents of golden ISO
Scheme : localdisk
Hostname : localhost
Username : None
SourceDir : /ws

Collecting software state..

Getting platform
Getting supported architecture
Getting active packages from XR
Getting inactive packages from XR
Getting list of RPMs in local repo
Getting list of provides of all active packages
Getting provides of each rpm in repo
Getting requires of each rpm in repo
Fetching .... <platform-name-golden-x-<version>.iso

Label within GISO: <label-name>

Adding packages
        <platform-name-golden-x-<version>.iso

UTC: sdr_instmgr[1150]: %INSTALL-INSTMGR-2-OPERATION_SUCCESS : Install operation 2 finished
 successfully

Install add operation successful

Activating <platform-name-golden-x-<version>

Install operation 3 started by root:
  install activate pkg <platform-name-golden-x-<version>-<label> replace

Package list:
    <platform-name-golden-x-<version>-<label>

This install operation will reload the system, continue?

 [yes/no]:[yes] Install operation will continue in the background

exec-timeout is resumed.
```

```
Router# Install operation 3 finished successfully
%INSTALL-INSTMGR-2-OPERATION_SUCCESS : Install operation 3 finished successfully
sdr_instmgr[1150]: %INSTALL-INSTMGR-2-SYSTEM_RELOAD_INFO : The whole system will be reloaded
 to complete install operation 3
```

- **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
--------------------
ncs5000-sysadmin-6.2.2
ncs5000-sysadmin-hostos-6.2.2-r622.CSCcv10001.admin.x86_64
ncs5000-sysadmin-hostos-6.2.2-r622.CSCcv10001.admin.arm
ncs5000-sysadmin-system-6.2.2-r622.CSCcv10005.x86_64
ncs5000-sysadmin-system-6.2.2-r622.CSCcv10005.arm
....
XR repository
-----------------
ncs5000-iosxr-mgbl-3.0.0.0-r622.x86_64
ncs5000-xr-6.2.2
....
Host repository
--------------------
host-6.2.2
```

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

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

```
Router#show install package ncs5000-goldenk9-x64-6.2.2

This may take a while ...
    ISO Name: ncs5000-goldenk9-x64-6.2.2
    ISO Type: bundle
    ISO Bundled: ncs5000-mini-x64-6.2.2
    Golden ISO Label: temp
    ISO Contents:
        ISO Name: ncs5000-xr-6.2.2
        ISO Type: xr
        rpms in xr ISO:
            iosxr-os-ncs5000-64-5.0.0.0-r622
            iosxr-ce-ncs5000-64-3.0.0.0-r622
            iosxr-infra-ncs5000-64-4.0.0.0-r622
            iosxr-fwding-ncs5000-64-4.0.0.0-r622
            iosxr-routing-ncs5000-64-3.1.0.0-r6122

        ISO Name: ncs5000-sysadmin-6.2.2
        ISO Type: sysadmin
        rpms in sysadmin ISO:
            ncs5000-sysadmin-topo-6.2.2-r622
            ncs5000-sysadmin-shared-6.2.2-r622
            ncs5000-sysadmin-system-6.2.2-r622
            ncs5000-sysadmin-hostos-6.2.2-r622.admin
        ...

        ISO Name: host-6.2.2
        ISO Type: host
```

```
        rpms in host ISO:
            ncs5000-sysadmin-hostos-6.2.2-r622.host

        Golden ISO Rpms:
            xr rpms in golden ISO:
                ncs5000-k9sec-x64-2.2.0.1-r622.CSCxr33333.x86_64.rpm
                openssh-scp-6.6p1.p1-r0.0.CSCtp12345.xr.x86_64.rpm
                openssh-scp-6.6p1-r0.0.xr.x86_64.rpm
                ncs5000-mpls-x64-2.1.0.0-r622.x86_64.rpm
                ncs5000-k9sec-x64-2.2.0.0-r622.x86_64.rpm

            sysadmin rpms in golden ISO:
                ncs5000-sysadmin-system-6.2.2-r622.CSCcv11111.x86_64.rpm
                ncs5000-sysadmin-system-6.2.2-r622.CSCcv11111.arm.rpm
                openssh-scp-6.6p1-r0.0.admin.x86_64.rpm
                openssh-scp-6.6p1-r0.0.admin.arm.rpm
                openssh-scp-6.6p1.p1-r0.0.CSCtp12345.admin.x86_64.rpm
                openssh-scp-6.6p1.p1-r0.0.CSCtp12345.admin.arm.rpm
                ncs5000-sysadmin-hostos-6.2.2-r622.CSCcv10001.admin.x86_64.rpm
                ncs5000-sysadmin-hostos-6.2.2-r622.CSCcv10001.admin.arm.rpm

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

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

# Install Replace with Golden ISO

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

**Example:**

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

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

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

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

Install operation 13 finished successfully


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

Router# Install operation 46 finished successfully
```

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

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

For example,

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

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

Using the `commit` keyword in the command automatically starts the **install commit** operation after the router reloads during activation of the image.

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

**Step 2**  **show  version**

**Example:**

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

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

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

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

# Upgrade the System Using IOS XR Configuration File in GISO

You can upgrade the system when a newer version is available. A system upgrade replaces the existing version of the software with a newer version. By keeping the software up to date, you can ensure that the device works with the latest features and bug fixes.

You can replace the currently active software on your system with the software from a specified ISO image or GISO image. Only a minimal set of changes is performed to upgrade to the new software. Packages are

not removed and reinstalled if they have the same name and version. For example, an upgrade that differs by only one package, removes and install only that one modified package.

**Note** The instructions in this section also apply to system downgrade.

This section shows replacing the current software version with image.

*Table 1: Feature History Table*

| Feature Name | Release Information | Description |
|---|---|---|
| IOS XR Configuration File in Golden ISO (GISO) | Release 7.5.2 | GISO is a customized image with the standard functional components and additional configuration files. This feature extracts the IOS XR configuration file in GISO and automates the updating of configuration files when the router is reloaded with the new GISO.<br><br>This feature introduces **iso-config [ignore \| replace]** keywords to the **install replace** and **install package replace** commands. |

In this scenario, you replace the current software with image, apply the changes, and commit the install operation. Committing the changes indicates the end of the current transaction. The updated software is used after the changes are applied, before the install transaction is committed.

A reboot is not always necessary. Bugfixes can be applied using a GISO that contains the current running software and the additional bugfix. These fixes may not require a reboot. Reboots are required for a system version change, if too many processes need restarting, there is configuration in the GISO to apply or if the bugfix is marked as requiring a reload.

**Note** The ztp.ini is an initialization file for ZTP that is provided when building a golden ISO (GISO). This file defines the GISO ZTP configuration. If the `.ini` file is used, the next time ZTP starts, this custom `.ini` file that was used is picked up. For example, if the install starts a reload operation or when ZTP is initiated manually, the ZTP runs with the custom `ini`. This file is carried over during install replace and install rollback, as long as both the `from-version` and the `to-version` are Cisco IOS XR, Release 7.3.1 or higher.

If you start an **install replace** or **install package replace** operation with a GISO that has a ztp.ini file, the GISO ZTP configuration is updated on both RPs when the operation is applied. To undo this install operation and also the change in GISO ZTP configuration, use **install package abort** command. If you replace the image using an ISO without a ztp.ini file, the GISO ZTP configuration is removed from both RPs, if it exists, when the operation is applied.

Cisco IOS XR Software Release 7.5.2 introduces **iso-config [ignore \| replace]** keywords to the **install replace** and **install package replace** commands. Before this release, the configuration file in GISO is ignored by default. If the GISO has a configuration file and it is not ignored using the **iso-config ignore** option, then the configuration in the file is applied, and the install operation requires a reboot. With the **iso-config** keyword, the default value is set to **replace** to replace the running configuration with the configuration in the file.

You can extract the configuration file from GISO using `image.py` script that is embedded in the ISO. This script interprets the GISO, and serves as an API to the ISO content at the filesystem level. The extracted configuration file replaces the running configuration after the router reloads.

*Table 2: Configuration File in GISO*

| Scenario | iso-config replace | iso-config ignore | Nothing Specified |
| --- | --- | --- | --- |
| Configuration is present in GISO | Extracts the configuration and replaces the running configuration after the changes are applied via router reload. The output of the **show install log** command displays `ISO config replace requested (ISO config present).` | Does not extract or apply the configuration. The output of the **show install log** command does not display any information related to the configuration. | Extracts the configuration and replaces the running configuration after the changes are applied via router reload. The behavior is similar to **iso-config replace** operation. |
| Configuration is not present in GISO | Attempts to extract the configuration, and the operation continues as if configuration is not requested. The output of the **show install log** command displays `ISO config replace requested (ISO config not present).` | Does not extract or apply the configuration. The output of the **show install log** command does not display any information related to the configuration. | Attempts to extract the configuration, and the operation continues as if configuration is not requested. The behavior is similar to **iso-config replace** operation. |
| Configuration extraction is not supported in target GISO (downgrade) | Attempts to extract the configuration, and the operation continues as if configuration is not requested. The output of the **show install log** command does not display any information related to the configuration. | Does not extract or apply the configuration. The output of the **show install log** command does not display any information related to the configuration. | Attempts to extract the configuration, and the operation continues as if configuration is not requested. The behavior is similar to **iso-config replace** operation. |
| Configuration is supported but the extraction fails | Attempts to extract the configuration. But the operation fails with error `Failed to extract the gISO config file from the ISO: <path to ISO>.` | Does not extract or apply the configuration. The output of the **show install log** command does not display any information related to the configuration. | The behavior is similar to **iso-config replace** operation. |

**Step 1**    Copy the ISO (or GISO) image to the `/harddisk:` location on the router.

**Step 2**    Upgrade the system to replace the current software with the image.

**Example:**

```
Router#install package replace /harddisk:/
```

**Step 3**    Activate the new .iso image on the router by applying the changes.

**Example:**

```
Router#install apply {reload | restart} [noprompt]
```

**Note**    You can use a single command to perform both the packaging operation and activating the applying the changes using **install replace /harddisk:/5500-x64.iso noprompt** command.

To identify whether a reload is required or only process restart is needed, use either **show install history last transaction verbose** command or **show install request** command.

Include the keyword `noprompt` in the command to enable the system to bypass your permission to reload the router.

Applying the change gives you the flexibility to test the operation of the new software before committing the changes. If you reload the router, the router reverts the software to its previous software state.

All operations that automatically apply the new software are prohibited when an atomic change is already in progress. You must address the current atomic-change before performing this operation. To address the change, apply the current atomic-change, or cancel it with the **install package abort all-since-apply** command.

**Step 4**    View the install log.

**Example:**

```
Router#show install log detail
Fri Nov 12 09:43:49.702 UTC
2021-11-12 09:33:47 UTC    Transaction 1 started
2021-11-12 09:33:47 UTC    Atomic change 1.1 started
2021-11-12 09:33:47 UTC    Packaging operation 1.1.1 started
2021-11-12 09:33:47 UTC    Replace


2021-11-12 09:35:58 UTC    Packaging operation 1.1.1 success
2021-11-12 09:36:04 UTC    Apply by reload started
2021-11-12 09:38:48 UTC    Atomic change 1.1 successfully applied by reload
```

**Step 5**    Verify that the image is activated successfully.

**Example:**

```
Router#show install request
```

**Step 6**    Commit the transaction.

**Example:**

```
Router#install commit
```

**Note**    Any action requests may be run synchronously from the CLI. During this request, you are updated with the status of the request whenever it changes. The following example shows the output from a synchronous action request:

```
Router#install commit synchronous
Starting:
  install commit
Transaction 1
The install operation will continue in the background
Press Ctrl-C to return to the exec prompt. This will not cancel the install operation

Current activity: Initializing ....
Current activity: Commit transaction ......

Transaction 4: 'install commit' completed without error
```

# Disaster Recovery

The topics covered in this chapter are:

# Boot using USB Drive

The bootable USB drive is used to re-image the router for the purpose of system upgrade or boot the router in case of boot failure. The bootable USB drive can be created using a compressed boot file.

## Create a Bootable USB Drive Using Compressed Boot File

A bootable USB drive is created by copying a compressed boot file into a USB drive. The USB drive becomes bootable after the contents of the compressed file are extracted.

**Note**    In case of failure to read or boot from USB drive, ensure that the drive is inserted correctly. If the drive is inserted correctly and still fails to read from USB drive, check the contents of the USB on another system.

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

**Before you begin**

- You have access to a USB drive with a storage capacity that is between 8GB (min) and 32 GB (max). USB 2.0 and USB 3.0 are supported.

- Copy the compressed boot file from the software download page at cisco.com to your local machine. The file name for the compressed boot file is in the format *ncs5k-usb-boot-<release_number>.zip* .

**Step 1**    Connect the USB drive to your local machine and format it with FAT32 or MS-DOS file system using the Windows Operating System or Apple MAC Disk Utility.

**Step 2**    Copy the compressed boot file to the USB drive.

**Step 3** Verify that the copy operation is successful. To verify, compare the file size at source and destination. Additionally, verify the MD5 checksum value.

**Step 4** Extract the content of the compressed boot file by unzipping it inside the USB drive. This converts the USB drive to a bootable drive.

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

**Step 5** Eject the USB drive from your local machine.

### What to do next

Use the bootable USB drive to boot the router or upgrade its image.

# Boot the Router Using USB

The router can be booted using an external bootable USB drive. This might be required when the router is unable to boot from the installed image. A boot failure may happen when the image gets corrupted. During the USB boot, process the router gets re-imaged with the version available on the USB drive.

> **Note** During the USB boot process, the router is completely re-imaged with the ISO image version present in the bootable USB drive. All existing configurations are deleted because the disk 0 content is erased. No optional packages are installed during the upgrade process; they need to be installed after the upgrade is complete.

### Before you begin

- Create a bootable USB drive. See Create a Bootable USB Drive Using Compressed Boot File, on page 71.

- Ensure that an external connection unit (ECU) with two solid-state drives (SSDs) is present.

**Step 1** Connect the USB drive to the active RP.

**Step 2** Connect to the console

**Step 3** Power the router.

**Step 4** Press **Esc** to pause the boot process and get the RPs to BIOS menu.

**Step 5** Select the USB from the boot menu on the RP to which the USB is connected to.

The image is copied in internal disk, and the router is restarted automatically.

### What to do next

- After the booting process is complete, specify the root username and password.

- Install the required optional packages.

# Boot the Router Using iPXE

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

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

## Zero Touch Provisioning

Zero Touch Provisioning (ZTP) helps in auto provisioning after the software installation of the router using iPXE.

ZTP auto provisioning involves:

- **Configuration:** Downloads and executes the configuration file. The first line of the file must contain `!! IOS XR` for ZTP to process the file as a configuration.

- **Script:** Downloads and executes the script files. The script files include a programmatic approach to complete a task. For example, scripts created using IOS XR commands to perform patch upgrades. The first line of the file must contain `#! /bin/bash` or `#! /bin/sh` for ZTP to process the file as a script.

## Setup DHCP Server

A DHCP server must be configured for IPv4, IPv6 or both communication protocols. The following example shows ISC-DHCP server running on Linux system.

**Before you begin**

- Consult your network administrator or system planner to procure IP addresses and a subnet mask for the management interface.

- Physical port Ethernet 0 or Ethernet 1 on RP is the management port. Ensure that the port is connected to management network.

- Enable firewall to allow the server to process DHCP packets.

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

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

```
                                    AdvAutonomous on;
                                    AdvRouterAddr off;
                    };
             };
```

• The HTTP server can be in the same server as that of the DHCP server, or can be on a different server. After the IP address is assigned from DHCP server, the router must connect to the HTTP server to download the image.

**Step 1** Create the `dhcpd.conf` file (for IPv4, IPv6 or both communication protocols), `dhcpv6.conf` file (for IPv6) or both in the `/etc/` or `/etc/dhcp` directory. This configuration file stores the network information such as the path to the script, location of the ISO install file, location of the provisioning configuration file, serial number, MAC address of the router.

**Step 2** Test the server once the DHCP server is running. For example, for IPv4:

• Use MAC address of the router:

> **Note** Using the `host` statement provides a fixed address that is used for DNS, however, verify that option `77` is set to iPXE in the request. This option is used to provide the bootfile to the system when required.

```
host ncs5k {
    hardware ethernet <router-mac-address>;
    if exists user-class and option user-class = "iPXE" {
        filename = "http://<httpserver-address>/<path-to-image>/ncs5k-mini-x.iso";
    }
    fixed-address <ip address>;
}
```

Ensure that the above configuration is successful.

• Use serial number of the router:

```
host ncs5k
{
option dhcp-client-identifier "<router-serial-number>";
  filename "http://<IP-address>/<path-to-image>/ncs5k-mini-x.iso";
  fixed-address <IP-address>;
}
```

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

**Step 3** Restart DHCP.

```
killall dhcpd
/usr/sbin/dhcpd -f -q -4 -pf /run/dhcp-server/dhcpd.pid
-cf /etc/dhcp/dhcpd.conf ztp-mgmt &
```

**Example**

The example shows a sample `dhcpd.conf` file:

```
allow bootp;
allow booting;
ddns-update-style interim;
option domain-name "cisco.com";
option time-offset -8;
ignore client-updates;
default-lease-time 21600;
```

```
        max-lease-time 43200;
        option domain-name-servers <ip-address-server1>, <ip-address-server2>;
        log-facility local0;
         :
        subnet <subnet> netmask <netmask> {
          option routers <ip-address>;
          option subnet-mask <subnet-mask>;
          next-server <server-addr>;
        }
         :
        host <hostname> {
          hardware ethernet e4:c7:22:be:10:ba;
          fixed-address <address>;
          filename "http://<address>/<path>/<image.bin>";
        }
```

The example shows a sample `dhcpd6.conf` file:

```
option dhcp6.name-servers <ip-address-server>;
option dhcp6.domain-search "cisco.com";
dhcpv6-lease-file-name "/var/db/dhcpd6.leases";
option dhcp6.info-refresh-time 21600;
option dhcp6.bootfile-url code 59 = string;
subnet6 <subnet> netmask <netmask> {
        range6 2001:1851:c622:1::2 2001:1851:c622:1::9;
         option dhcp6.bootfile-url "http://<address>/<path>/<image.bin>";
```

**What to do next**

Invoke ZTP.

# Invoke ZTP

ZTP runs within the XR namespace, and within the global VPN routing/forwarding (VRF) namespace for management interfaces and line card interfaces.

**Before you begin**

Ensure that a DHCP server is setup. For more information, see .

Edit the dhcpd.conf file to utilize the capabilities of ZTP.

The following example shows a sample DHCP server configuration including iPXE and ZTP:

```
host <host-name>
{
hardware ethernet <router-serial-number or mac-id>;
fixed-address <ip-address>;
  if exists user-class and option user-class = "iPXE" {
  # Image request, so provide ISO image
  filename "http://<ip-address>/<directory>/ncs5k-mini-x.iso";
  } else
{
  # Auto-provision request, so provide ZTP script or configuration
  filename "http://<ip-address>/<script-directory-path>/ncs5k-ztp.script";
  #filename "http://<ip-address>/<script-directory-path>/ncs5k-ztp.cfg
```

```
      }
   }
```

**Note**    Either the ZTP .script file or the .cfg file can be provided at a time for auto-provisioning.

With this configuration, the system boots using ncs5k-mini-x.iso during installation, and then download and execute ncs5k-ztp.script when XR VM is up.

# Invoke ZTP Manually

ZTP can also be invoked manually with the modified one touch provisioning approach. The process involves:

### Before you begin

A configuration file can be used to specify a list of interfaces that will be brought up in XR and DHCP will be invoked on. /pkg/etc/ztp.config is a platform specific file that allows the platform to specify which if any additional interfaces will be used.

```
#
# List all the interfaces that ZTP will consider running on. ZTP will attempt
# to bring these interfaces. At which point dhclient will be able to use them.
#
# Platforms may add dynamically to this list.
#
#ZTP_DHCLIENT_INTERFACES=" \
#    Gi0_0_0_0 \
#"
...
```

**Step 1**    Boot the router.

**Step 2**    Login manually.

**Step 3**    Enable interfaces.

**Step 4**    Invoke a new ZTP DHCP session manually using the **ztp initiate** command.

```
Router#ztp initiate
```

For example, to send DHCP requests on the GigabitEthernet interface 0/0/0/0, run the command:

```
Router#ztp initiate debug verbose interface GigabitEthernet0/0/0/0
```

ZTP will run on the management port by default unless the platform has configured otherwise. The logs will be logged in /disk0:/ztp/ztp/log location.

**Note**    To configure a 40G interface into 4 separate 10G interfaces, use the **ztp breakout nosignal-stay-in-breakout-mode** command.

Note    To enable dataport breakouts and invoke DHCP sessions on all dataport and line card interfaces that are detected, use the **ztp breakout** command.

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

To override the prompt:

```
Router#ztp initiate noprompt
Invoke ZTP?(this may change your configuration) [confirm] [y/n]:

ZTP will now run in the background.
Please use "show logging" or look at /disk0:/ztp/ztp/log to check progress.
```

ZTP runs on the management interfaces that are UP by default.

**Step 5**    To terminate the ZTP session, use the **ztp terminate** command.

### What to do next

Boot the router using iPXE.

# Additional Commands for Manually Invoking ZTP

The following table lists some of the additional commands that are useful while manually invoking ZTP.

*Table 3: Additional Commands for Manually Invoking ZTP*

| Command | Description |
|---------|-------------|
| **ztp initiate management** | Use this command to send DHCP request on the management interface |
| **ztp initiate dhcp4** | Use this command to send DHCP IPv4 requests. |
| **ztp initiate dhcp4-client-identifier** *unique-identifier* | Use this command to override the default DHCP IPv4 client identifier. |
| **ztp initiate dhcp6** | Use this command to send DHCP IPv6 requests. |
| **ztp initiate dscp** *dscp-value* | Use this command to set the DSCP value in the IPv4 packet header. |
| **ztp initiate dscp6** *dscp-value* | Use this command to set the DSCP value in the IPv6 header |
| **ztp breakout nosignal-stay-in-breakout-mode** | Use this command to keep the interfaces in breakout mode when there is no signal. |
| **ztp breakout nosignal-stay-in-state-noshut** | Use this command to keep the interfaces up when there is no signal. |
| **ztp breakout hostname** *hostname* | Use this command to set the XR hostname. |

| ztp clean | Use this command to remove all ZTP log and settings. |
|---|---|
| ztp terminate | Use this command to cancel the ongoing ZTP request. |

# Boot the Router Using iPXE

Before you use the iPXE boot, ensure that:

- DHCP server is set and is running.

- You have logged in to the System Admin console using the **admin** command.

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

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

**Example:**

```
sysadmin-vm:0_RP0# hw-module location all bootmedia network reload
Wed Dec 23 15:29:57.376 UTC
Reload hardware module ? [no,yes]
```

The following example shows the output of the command:

```
iPXE 1.0.0+ (3e573) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP TFTP VLAN EFI ISO9660 NBI Menu
Trying net0...
net0: c4:72:95:a6:14:e1 using dh8900cc on PCI01:00.1 (open)
[Link:up, TX:0 TXE:0 RX:0 RXE:0]
Configuring (net0 c4:72:95:a6:14:e1).................. Ok << Talking to DHCP/PXE server to
 obtain network information
net0: 10.37.1.101/255.255.0.0 gw 10.37.1.0
net0: fe80::c672:95ff:fea6:14e1/64
net0: 2001:1800:5000:1:c672:95ff:fea6:14e1/64 gw fe80::20c:29ff:fefb:b9fe
net1: fe80::c672:95ff:fea6:14e3/64 (inaccessible)
Next server: 10.37.1.235
Filename: http://10.37.1.235/ncs5k/ncs5k-mini-x.iso


http://10.37.1.235/ncs5k/ncs5k-mini-x.iso ... 58% << Downloading file as indicated by
DHCP/PXE server to boot install image
```

# Disaster Recovery Using Manual iPXE Boot

Manually booting the system using iPXE can be used to reinstall a clean system in case of a corrupt install or recover lost password. However, all the disks will be wiped out and the configuration will be removed.

**Step 1** Press the right arrow key to enter the **Cisco Boot Options** menu.

**Step 2** Use the arrow keys (up, down) to select **UEFI: Built-in EFI IPXE** to enable iPXE boot. The iPXE boot launches the auto boot.

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

**Step 3** Identify the management interface. If the management interface is connected properly and is UP, it displays Link:up in the following output:

**Example:**

```
iPXE initialising devices...Sysconf checksum failed. Using default values
ok

iPXE 1.0.0+ (aa070) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP TFTP VLAN EFI ISO9660 NBI Menu
iPXE> ifstat
net0: c4:72:95:a7:c9:30 using dh8900cc on PCI01:00.1 (closed)
  [Link:up, TX:0 TXE:0 RX:0 RXE:0]
net1: c4:72:95:a7:c9:31 using dh8900cc on PCI01:00.2 (closed)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Down (http://ipxe.org/38086193)]

iPXE> set net0/ip  10.x.x.y
iPXE> set net0/netmask 255.x.x.x
iPXE> set net0/gateway 10.x.x.x
iPXE> ifopen net0
iPXE> ping 10.x.x.z
64 bytes from 10.x.x.z: seq=1
64 bytes from 10.x.x.z: seq=2
Finished: Operation canceled (http://ipxe.org/0b072095)

iPXE> boot http://10.x.x.z/<dir-to-iso>/ncs5k-mini-x.iso-<version>_IMAGE
http://10.x.x.z/<dir-to-iso>/ncs5k-mini-x.iso-<version>_IMAGE... ok
Booting iso-image@0x430173000(803784704), bzImage@0x4301a0000(4473806)
...
```

Choose the net interface that shows `Link:up`. If there are multiple interfaces that show the status as UP, identify the management interface with MAC address.

iPXE also supports HTTP, TFTP and FTP. For more information, see https://ipxe.org/cmd.

After installing the mini ISO image, the system reboots. After successful reboot, specify the root username and password. Once you get back to the XR prompt, you can load the configuration and install remaining packages.