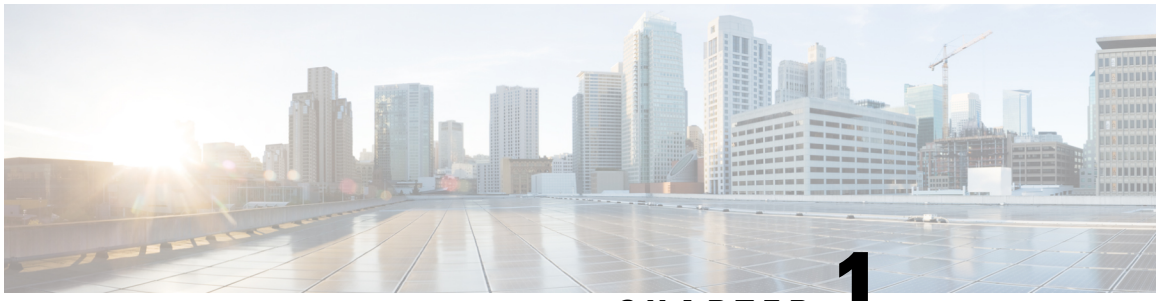
System Setup and Software Installation Guide for Cisco NCS 560 Series Routers, IOS XR Release 7.4.x

First Published: 2021-07-30

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

CONTENTS



CHAPTER 1

Cisco NCS 560-4 Product Overview

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

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

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

  support for extended temperature based on route processor configuration

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

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

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

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

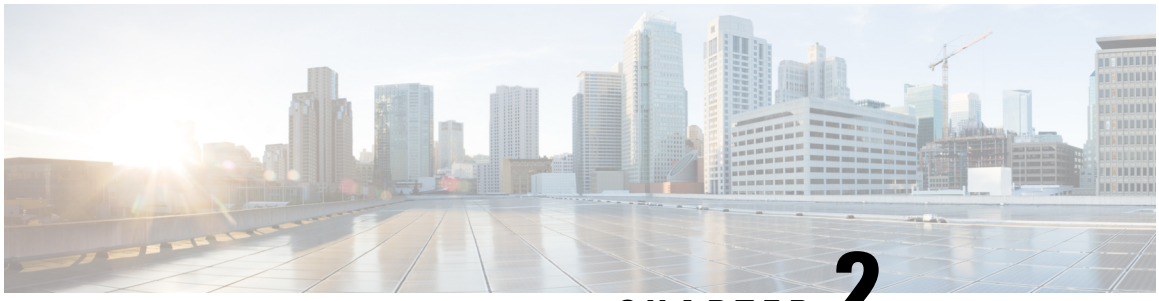
See the *Cisco N560-RSP4 and Cisco N560-RSP4-E Route Processor Hardware Installation Guide* for more information.

- Command Modes, on page 1

Command Modes

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

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



CHAPTER 2

Bring-up the Router

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

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

Boot the Router

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

Procedure

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

Step 2 Start the terminal emulation program on your workstation.

The console settings are:

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

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

Step 3 Power on the router.

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

Procedure

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

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

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

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

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

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

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

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

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

Step 4 **Username:** *username*

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

Step 5 **Password:** *password*

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

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

Displays user details.

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

What to do next

- Configure routing functions from the XR console.
- Configure system administration settings from the System Admin prompt. The System Admin prompt is displayed on accessing the System Admin console. For details on how to get the System Admin prompt, see Access the System Admin Console, on page 5.

Access the System Admin Console

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

Procedure

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

Step 2 **admin**

Example:

The following example shows the command output :

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

Mon May 22 06:57:29.350 UTC

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

Step 3 (Optional) **exit**

Return to the XR mode from the System Admin mode.

Configure the Management Port

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

Before you begin

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

Procedure

Step 1 **configure**

Example:

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

Enters mode.

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

Example:

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

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

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

Example:

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

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

Step 4 **no shutdown**

Example:

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

Places the interface in an "up" state.

Step 5 **exit**

Example:

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

Exits the Management interface configuration mode.

Repeat the above steps for the redundant route processor.

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

Example:

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

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

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

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

end —Prompts user to take one of these actions:

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

What to do next

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

Perform Clock Synchronization with NTP Server

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

Before you begin

Configure and connect to the management port.

Procedure

Step 1 **configure**

Example:

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

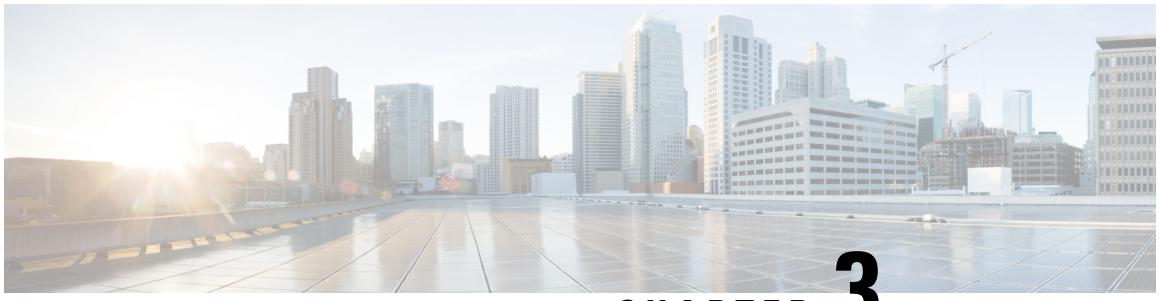
Enters mode.

Step 2 **ntp server** *server_address*

Example:

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

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



CHAPTER 3

Provision Network Devices using Zero Touch Provisioning

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

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

Learn about Zero Touch Provisioning

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

Note Currently, ZTP only supports single name-server. When the DHCP server has more than one server address configured, ZTP fails to apply the server configuration.

ZTP provides multiple options, such as:

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

Benefits of Using ZTP

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

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

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

Use Cases

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

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

You can initiate ZTP in one of the following ways:

- **Fresh Boot**: Use this method for devices that has no pre-loaded configuration. See Getting Started with ZTP on a Fresh Boot of a Router. See Zero Touch Provisioning on a Fresh Boot of a Router, on page 10
- **Manual Invocation:** Use this method when you want to forcefully initiate ZTP on a fully configured device. See Invoke ZTP Manually, on page 24.

Zero Touch Provisioning on a Fresh Boot of a Router

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

Fresh Boot Using Removable Storage Device

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

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

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

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

   Fetcher defines which port ZTP should use to get the provisioning details. By default, each port has a fetcher priority defined in the *ztp.ini* file.

2. ZTP checks for a USB flash drive on the device If the USB drive isn't available, the ZTP process moves to the next fetcher as defined in the fetcher priority of the `ZTP.ini` file.

3. If a USB flash drive is available, the device scans for the `xr-config` file in the root of the USB mount in the following sequence:

   a. The ZTP process first scans for the `router-cfg` file in the folder that is matching the chassis serial number of the device within the `xr-config` folder and applies the device-specific configuration.

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

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

   c. If the `xr-config` folder isn't available, the ZTP process moves to the next fetcher as defined in the fetcher priority of the `ZTP.ini` file.

4. The device applies the configuration.

5. The network device is now up and running.

Configure ZTP using USB

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

1. Create the configuration file. See Build your Configuration File, on page 13.

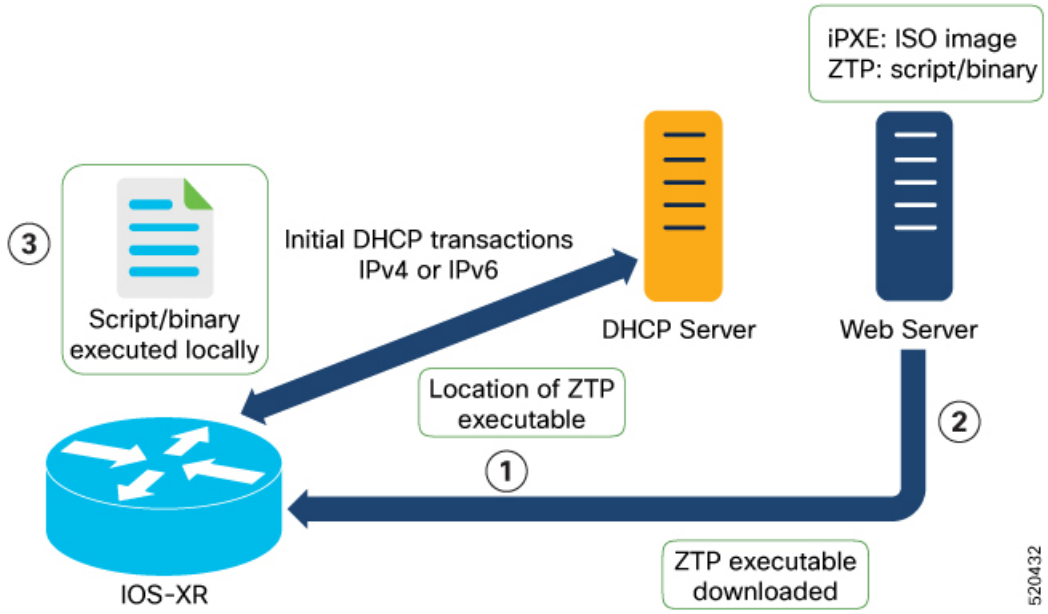
Note When you use a USB flash drive as a source for ZTP, you can't use the script file for provisioning. The script file isn't supported for USB fetcher.

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

Fresh Boot Using DHCP

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

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



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

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

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

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

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

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

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

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

Note

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

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

Build your Configuration File

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

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

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

The following is the sample configuration file:

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

hostname ios
alias exec al show alarms brief system active

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

Create User Script

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

Build your ZTP script with either shell and python. ZTP includes a set of CLI commands and a set of shell utilities that can be used within the user script.

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

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

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

ZTP Shell Utilities

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

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

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

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

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

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

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

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

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

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

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

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

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

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

API Implementation Behavior

The following tables list the API behavior when invoking these utilities:

Table 1: xrcmd

Steps	Description
`create user`	Creates `ztp-user` before executing the `xrapply` command.
`execute cmd`	Runs the command using the `xr_cli` parser utility. For example: `/pkg/bin/xr_cli -n "$*" | tee -a ${ZTP_LOG_XR_CMD} | tee -a ${ZTP_LOG}`
`remove user`	Deletes user using the `xrapply` command.

Table 2: xrapply/xrreplace

Steps	Description
`create user`	Creates `ztp-user` before executing the `xrapply` command, if required.

Steps	Description
`execute cmd`	Initiates configuration commit by using the `cfg-mgr` utility. For example: `/pkg/bin/config -p15 -X -t -f $config_filename $ZTP_CREDS -c "$apply_reason" &>$output`
`remove user`	Deletes user.

Table 3: xrapply_with_extra_auth/xrreplace_with_extra_auth

Steps	Description
`create user`	Creates `ztp-user` before running the `xrapply` command, if required.
`perform authentication`	Performs authentication using `ztp-user`.
`execute cmd`	Initiates configuration commit by using the `cfg-mgr` utility.
`remove user`	Deletes user.

ZTP Helper Python Library

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

ZtpHelpers Class Methods

Following are utility methods of the `ZtpHelpers` class:

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

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

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

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

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

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

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

- `toggle_debug(self, enable):`

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

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

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

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

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

- `setup_syslog(self):`

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

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

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

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

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

           :param reason: Reason for the config commit.
                          Will show up in the output of:
                          "show configuration commit list detail"
```

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

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

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

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

```
  Replace XR Configuration using a file

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

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

API Implementation Behavior

The following tables list the API behavior when invoking these utilities:

Table 4: xrcmd

Steps	Description
`create user`	Creates `ztp-user` before execution of command.
`execute cmd`	executes commands using parser utility `source /pkg/bin/ztp_helper.sh && echo -ne | xrcmd "show running-config"`
`remove user`	Deletes user.

Table 5: xrapply/xrreplace

Steps	Description
`create user`	Creates `ztp-user` before execution of the command.
`execute cmd`	intiate configuration commit using `cfg-mgr` utility
`remove user`	Deletes user.

Example

The following shows the sample script in phyton

```
[apple2:~]$ python sample_ztp_script.py

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


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


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

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


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


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

Building configuration...
{'output': ['!! IOS XR Configuration version = 6.2.1.21I',
            '!! Last configuration change at Sat Dec 17 04:23:25 2016 by UNKNOWN',
            '!',
            'hostname customer2',
            'username root',
            'group root-lr',
            'group cisco-support',
            'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1',
            '!',
            'username noc',
            'group root-lr',
            'group cisco-support',
            'secret 5 $1$7kTu$zjrgqbgW08vEXsYzUycXw1',
```

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

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

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

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

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

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

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

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

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

API Implementation Behavior

The following tables list the API behavior when invoking these utilities:

Table 6: xrcmd

Steps	Description
`create user`	Creates `ztp-user` before running the command.
`execute cmd`	Runs the command using the `xr_cli` parser utility. For example: `source /pkg/bin/ztp_helper.sh && echo -ne | xrcmd show running-config`
`remove user`	Deletes user.

Table 7: xrapply/xrreplace

Steps	Description
`create user`	Creates `ztp-user` before running the command.
`execute cmd`	Initiates configuration commit by using the `cfg-mgr` utility.
`remove user`	Deletes user.

Set Up DHCP Server

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

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

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

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

Example

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

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

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

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

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

The DHCP server responds with the following DHCP options:

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

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

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

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

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

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

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

```
    }

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

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


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

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


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

          }
        }


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

Authentication on Data Ports

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

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

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

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

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

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

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

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

```
log-facility local7;
option dhcp6.name-servers 2001:1451:c632:1::1;
option dhcp6.domain-search "cisco.com";
dhcpv6-lease-file-name "/var/lib/dhcpd/dhcpd6.leases";
option dhcp6.info-refresh-time 21600;
option dhcp6.bootfile-url code 59 = string;
option dhcp6.user-class code 15 = string;
option space CISCO-EXR-CONFIG code width 2 length width 2;
option CISCO-EXR-CONFIG.client-identifier code 1 = string;
option CISCO-EXR-CONFIG.authCode code 2 = integer 8;
option CISCO-EXR-CONFIG.md5sum code 3 = string;
option vsio.CISCO-EXR-CONFIG code 9 = encapsulate CISCO-EXR-CONFIG;
subnet6 2001:1451:c632:1::/64{
 range6 2001:1451:c632:1::2 2001:1451:c632:1::9;
 option CISCO-EXR-CONFIG.client-identifier "exr-config";
 option CISCO-EXR-CONFIG.authCode 1;
 #valid md5
 option CISCO-EXR-CONFIG.md5sum "90fd845ac82c77f834d57a034658d0f0";
 if option dhcp6.user-class = 00:04:69:50:58:45 {
  option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/cisco-2/image.iso";
 }
 else {
   #option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/cisco-2/cisco-mini-x.iso.sh";
   option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/cisco-2/ztp.cfg";
  }
}
```

Invoke ZTP Manually

You can invoke Zero Touch Provisioning (ZTP) manually through the Command Line Interface. This method is Ideal for verifying the ZTP configuration without a reboot. This manual approach helps you to provision

the router in stages. To invoke ZTP on an interface (data ports or management port), you don't have to bring up and configure the interface first.

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

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

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

  Configuration Example:

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

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

  Configuration Example:

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

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

  Configuration Example:

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

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

  Configuration Example:

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

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

  Configuration Example:

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

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

Configuration

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

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

Configuration Example:

```
Router# ztp initiate dataport
```

Configure ZTP BootScript

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

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

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

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

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

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

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

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

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

#
# Force an invoke of ZTP again. If there was a username normally it would not run. This
forces it.
# Kill off ztp if it is running already and suppress errors to the console when ztp runs
below and
# cleans up xrcmd that invokes it. ztp will continue to run however.
#
xrcmd "ztp terminate noprompt" 2>/dev/null
xrcmd "ztp initiate noprompt" 2>/dev/null
```

Customize ZTP Initialization File

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

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

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

In the following example, the Mgmt4 port has the highest priority:

```
[Fetcher Priority]
Mgmt4: 0
Mgmt6: 1
DPort4: 2
DPort6: 3
```

- `progress_bar`: Enable progress bar on the console. By default, the progress bar is disabled. To enable the progress bar, add the following entry in the `ztp.ini` file.

```
[Options]
progress_bar: True
```

By default, the `ztp.ini` file is located in the `/pkg/etc/` location. To modify the ZTP configurable options, make a copy of the file in the `/disk0:/ztp/` directory and then edit the `ztp.ini` file.

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

Note Do not edit or delete the `ztp.ini` file in the `/pkg/etc/` location to avoid issues during installation.

The following example shows the sample of the `ztp.ini` file:

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

[Fetcher Priority]
Mgmt4:  1
Mgmt6:  2
DPort4: 3
DPort6: 4
```

Enable ZTP Using CLI

If you want to enable ZTP using CLI, use the **ztp enable** command.

Configuration example

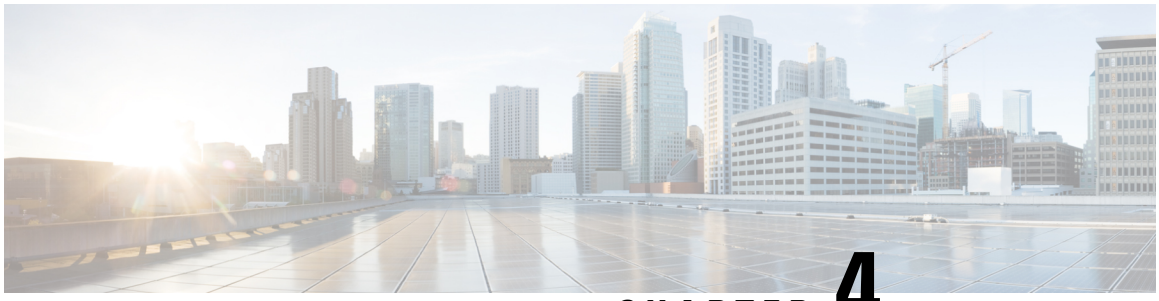
```
Router#ztp enable
Fri Jul 12 16:09:02.154 UTC
Enable ZTP? [confirm] [y/n] :y
ZTP Enabled.
```

Disable ZTP Using CLI

If you want to disable ZTP using CLI, use the **ztp disable** command.

Configuration example

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



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 Status of Hardware Modules

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

Before you begin

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

Procedure

show platform

Example:

Verify Node Status

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

Procedure

Step 1 **show platform**

Example:

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

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

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

Step 2 **admin**

Example:

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

Enters mode.

Step 3 **show platform**

Example:

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

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

```
sysadmin-vm:0_RP0# show platform
Thu Mar  28 08:19:08.640 UTC+00:00
Location  Card Type              HW State      SW State      Config State
------------------------------------------------------------------------------
0/0       NCS4200-1T16G-PS       OPERATIONAL   N/A           NSHUT
0/1       NCS4200-1T16G-PS       OPERATIONAL   N/A           NSHUT
0/2       NCS4200-1T16G-PS       OPERATIONAL   N/A           NSHUT
0/3       NCS4200-1T16G-PS       OPERATIONAL   N/A           NSHUT
0/4       A900-IMA8Z             OPERATIONAL   N/A           NSHUT
0/5       A900-IMA8Z             OPERATIONAL   N/A           NSHUT
0/7       N560-IMA1W             OPERATIONAL   N/A           NSHUT
0/9       N560-IMA2C             OPERATIONAL   N/A           NSHUT
0/10      A900-IMA8Z             OPERATIONAL   N/A           NSHUT
0/11      A900-IMA8Z             OPERATIONAL   N/A           NSHUT
0/12      NCS4200-1T16G-PS       OPERATIONAL   N/A           NSHUT
0/13      NCS4200-1T16G-PS       OPERATIONAL   N/A           NSHUT
0/14      NCS4200-1T16G-PS       OPERATIONAL   N/A           NSHUT
0/15      NCS4200-1T16G-PS       OPERATIONAL   N/A           NSHUT
0/RP0     N560-RSP4-E            OPERATIONAL   OPERATIONAL   NSHUT
0/RP1     N560-RSP4-E            OPERATIONAL   OPERATIONAL   NSHUT
0/FT0     N560-FAN-H             OPERATIONAL   N/A           NSHUT
0/PM0     A900-PWR1200-A         OPERATIONAL   N/A           NSHUT
0/PM2     A900-PWR1200-A         OPERATIONAL   N/A           NSHUT

sysadmin-vm:0_RP0#
```

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

Hardware states:

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

Software states:

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

Verify Environmental Parameters

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

Procedure

Step 1 **show environment temperatures**

Example:

```
sysadmin-vm:0_RP0# show environment temperatures
Mon Jul  29 11:12:24.828 UTC+00:00
================================================================================
Location  TEMPERATURE                 Value   Crit Major Minor Minor Major  Crit
          Sensor                   (deg C)    (Lo) (Lo)   (Lo)   (Hi)   (Hi)   (Hi)
--------------------------------------------------------------------------------
0/RP0
          QMX Die Temp                  55    -40   -30   -20   100   108   112
          Inlet                         34    -40   -30   -20    70    75    85
          FPGA Die                      60    -40   -30   -20    95    98   102
          Outlet                        53    -40   -30   -20    85    90    95
          Humidity                      21    -40   -30   -20    85    95    98
0/FT0
          Fan Inlet                     37    -10    -5     0   100   110   120
0/PM2
          Inlet Temperature             38    -40   -30   -20    95   100   105
          Outlet Temperature            42    -40   -30   -20    75    80    85
sysadmin-vm:0_RP0#
```

Step 2 **show environment fan**

Example:

```
sysadmin-vm:0_RP0# show environment fan
Mon Jul  29 11:13:30.258 UTC+00:00
================================================================================
                        Fan speed (rpm)
```

```
Location    FRU Type           FAN_0    FAN_1    FAN_2    FAN_3    FAN_4    FAN_5
                                                 FAN_6    FAN_7    FAN_8    FAN_9    FAN_10   FAN_11

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

                                                  13215    13321    16189    16304
```

Verify Software Version

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

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

Procedure

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 Perform System Upgrade and Install Feature Packages.

Verify Firmware Version

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

Procedure

show hw-module fpd

Example:

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

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

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

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

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

Displays the list of hardware modules detected on the router.

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

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

- FPD Device—Name of the hardware component, such as IO FPGA, IM FPGA, and BIOS.
- Status—Upgrade status of the firmware. The different states are:
  - CURRENT—The firmware version is the latest version.
  - READY—The firmware of the FPD is ready for an upgrade.
  - NOT READY—The firmware of the FPD is not ready for an upgrade.
  - NEED UPGD—A newer firmware version is available in the installed image. It is recommended that an upgrade be performed.
  - RLOAD REQ—The upgrade has been completed, and the ISO image requires a reload.
  - UPGD DONE—The firmware upgrade is successful.
  - UPGD FAIL—The firmware upgrade has failed.
  - BACK IMG—The firmware is corrupted. Reinstall the firmware.
  - UPGD SKIP—The upgrade has been skipped because the installed firmware version is higher than the one available in the image.
- Running—Current version of the firmware running on the FPD.
- Programmd—Version of the FPD programmed on the module.

What to do next

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

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

Verify Interface Status

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

Procedure

show ipv4 interface summary

Example:

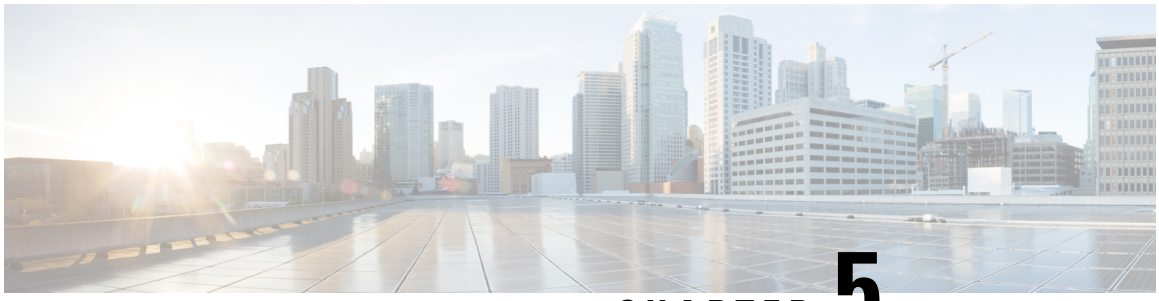
```
RP/0/RP0/CPU0:router#show ipv4 interface summary
```

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

In the above result:

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

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



CHAPTER 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.

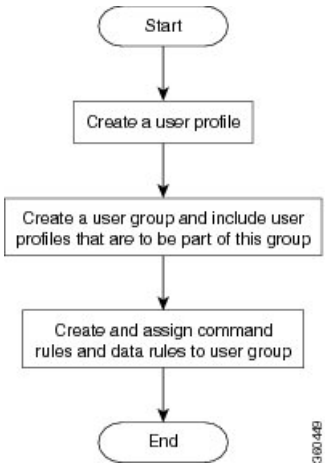
Note If any user on XR is deleted, the local database checks whether there is a first user on System Admin VM.

- 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 a User Profile in System Admin VM

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

The router supports a maximum of 1024 user profiles.

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

Procedure

Step 1 **admin**

Example:

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

Enters mode.

Step 2 **config**

Example:

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

Enters System Admin Config System Admin Configmode.

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

Example:

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

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

Step 4 **password** *password*

Example:

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

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

Step 5 **uid** *user_id_value*

Example:

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

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

Step 6 **gid** *group_id_value*

Example:

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

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

Step 7 **ssh_keydir** *ssh_keydir*

Example:

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

Specify any alphanumeric value.

Step 8 **homedir** *homedir*

Example:

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

Specify any alphanumeric value.

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

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

end —Prompts user to take one of these actions:

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

Create a User Group in System Admin VM

Create a user group for the System Admin VM.

The router supports a maximum of 32 user groups.

Before you begin

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

Procedure

Step 1 **admin**

Example:

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

Enters mode.

Step 2 **config**

Example:

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

Enters System Admin Config mode.

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

Example:

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

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

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

Step 4 **users** *user_name*

Example:

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

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

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

Step 5 **gid** *group_id_value*

Example:

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

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

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

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

end —Prompts user to take one of these actions:

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

What to do next

- Create command rules.
- Create data rules.

Create Command Rules

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

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

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

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

Each command rule is identified by a number associated with it. When multiple command rules are applied to a user group, the command rule with a lower number takes precedence. For example, cmdrule 5 permits

read access, while cmdrule10 rejects read access. When both these command rules are applied to the same user group, the user in this group gets read access because cmdrule 5 takes precedence.

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

Before you begin

Create an user group. See Create a User Group in System Admin VM, on page 40.

Procedure

Step 1 **admin**

Example:

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

Enters mode.

Step 2 **config**

Example:

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

Enters System Admin Config mode.

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

Example:

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

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

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

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

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

Step 4 **command** *command_name*

Example:

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

Specify the command for which permission is to be controlled.

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

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

Example:

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

Specify the operation for which permission has to be specified:

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

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

Example:

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

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

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

Step 7 **group** *user_group_name*

Example:

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

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

Step 8 **context** *connection_type*

Example:

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

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

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

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

end —Prompts user to take one of these actions:

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

What to do next

Create data rules. See Create Data Rules, on page 44.

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 40.

Procedure

Step 1 **admin**

Example:

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

Enters mode.

Step 2 **config**

Example:

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

Enters System Admin Config mode.

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

Example:

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

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

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

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

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

Step 4 **keypath** *keypath*

Example:

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

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

Step 5 **ops** *operation*

Example:

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

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

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

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

Example:

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

Specify whether users are permitted or denied the operation.

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

Step 7 **group** *user_group_name*

Example:

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

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

Step 8 **context** *connection type*

Example:

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

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

Step 9 **namespace** *namespace*

Example:

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

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

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

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

end —Prompts user to take one of these actions:

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

Change Disaster-recovery Username and Password

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

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

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

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

Procedure

Step 1 **admin**

Example:

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

Enters mode.

Step 2 **config**

Example:

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

Enters System Admin Config mode.

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

Example:

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

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

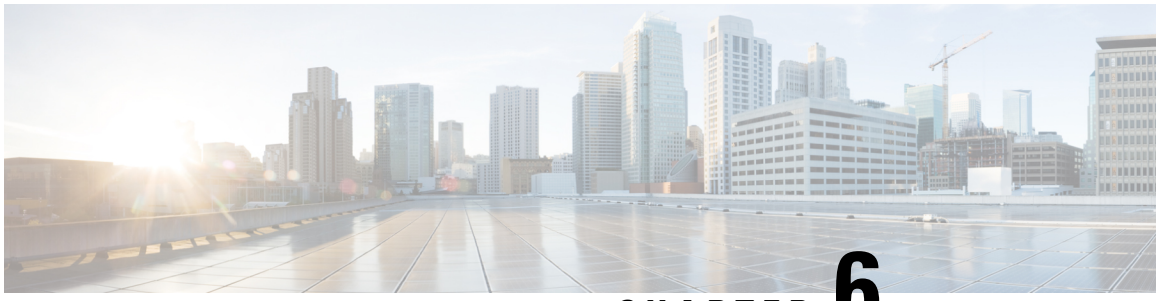
When you need to make use of the disaster recovery username, you need to enter it as *username***@localhost**.

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

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

end —Prompts user to take one of these actions:

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



CHAPTER 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

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

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

Note

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

Perform a system upgrade by installing a base package–Cisco IOS XR Unicast Routing Core Bundle. To install this bundle, run the **install** command. The filename for the Cisco IOS XR Unicast Routing Core Bundle bundle is *ncs560-mini-x.iso*.

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

Do not reload the router during an upgrade operation.

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

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

Upgrading Features

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

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

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

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

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

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

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

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

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

Third-party SMUs

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

- To activate a third-party SMU you should have a corresponding base package.
- When you activate a third-party SMU, the corresponding third-party base package state is inactive, this is an expected behavior.
- To deactivate a third-party SMU, you should activate corresponding third-party base package.

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

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

Workflow for Install Process

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

For installing a package, see Install Packages, on page 52. For uninstalling a package, see Uninstall Packages, on page 57.

Install Packages

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

Note

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

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

Before you begin

- Review the Install the Latest FPD on the Cisco NCS560 Routers TechNote.
- Configure and connect to the management port. The installable file is accessed through the management port. For details about configuring the management port, see Configure the Management Port.
- Copy the package to be installed either on the router's hard disk or on a network server to which the router has access.

Procedure

Step 1 Execute one of these:

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

Example:

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

or

```
RP/0/RP0/CPU0:router# install add source sftp://root@8.33.5.15:/auto/ncs/package/
RP/0/RP0/CPU0:router# install add source
/harddisk:/ncs560-mpls-1.0.0.0-<release-number>.x86_64.rpm
ncs560-mgbl-2.0.0.0-<release-number>.x86_64.rpm
RP/0/RP0/CPU0:router# install add source
/harddisk:/ncs560-mpls-te-rsvp-1.0.0.0-<release-number>.x86_64.rpm
ncs560-mgbl-1.0.0.0-<release-number>.x86_64.rpm
```

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

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

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

Step 2 **show install request**

Example:

```
RP/0/RP0/CPU0:router#show install request
Thu Mar 28 13:29:03.219 IST

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

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

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

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

Step 3 **show install repository**

Example:

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

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

Step 4 **show install inactive**

Example:

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

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

Step 5 Execute one of the following:

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

Example:

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

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

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

Activation does not happen instantaneously, but takes some time. Activation of some SMUs require a manual reloading of the router. When such SMUs are activated, a warning message is displayed to perform reload. The components of the SMU get activated only after the reload is complete. Perform router reload immediately after executing the **install activate** command. If the SMU has dependency on both XR VM and System Admin VM, perform the reload after activating the SMU in both VMs so that they take effect simultaneously. To reload the router, use the **hw-module location all reload** command from the System Admin EXEC mode.

Step 6 **show install active**

Example:

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

Displays packages that are active.

Step 7 **install commit**

Example:

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

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

Installing Packages: Related Commands

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

What to do next

- After performing a system upgrade, upgrade FPD by using the **upgrade hw-module location all fpd all** command from the System Aadmin EXEC mode. The progress of FPD upgrade process can be monitored using the **show hw-module fpd** command in the System Aadmin EXEC mode. Reload the router after the FPD upgrade is completed.
- Verify the installation using the **install verify packages** command.
- Uninstall the packages or SMUs if their installation causes any issues on the router. See Uninstall Packages, on page 57.

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

Install Prepared Packages

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

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

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

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

Procedure

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

For details, see Install Packages, on page 52.

Step 2 **show install repository**

Example:

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

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

Step 3 **show install request**

Example:

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

Thu Mar 28 13:29:03.219 IST

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

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

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

Step 4 Execute one of these:

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

Example:

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

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

Step 5 **show install prepare**

Example:

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

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

Step 6 **install activate**

Example:

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

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

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

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

Step 7 **show install active**

Example:

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

Displays packages that are active.

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

Step 8 **install commit**

Example:

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

Installing Packages: Related Commands

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

What to do next

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

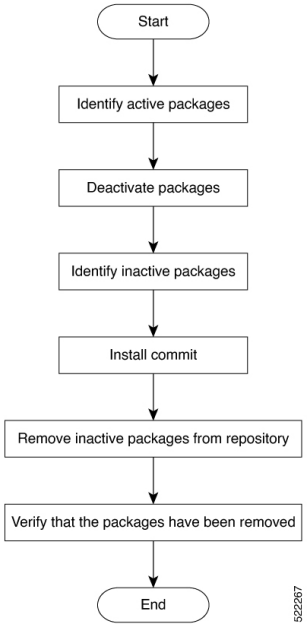
Uninstall Packages

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 2: Uninstalling Packages Workflow



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

Procedure

Step 1 **show install active**

Example:

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

Displays active packages. Only active packages can be deactivated.

Step 2 Execute one of these:

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

Example:

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

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

Step 3 **show install inactive**

Example:

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

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

Step 4 **install commit**

Step 5 **install remove** *package_name*

Example:

The inactive packages are removed from the repository.

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

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

Step 6 **show install repository**

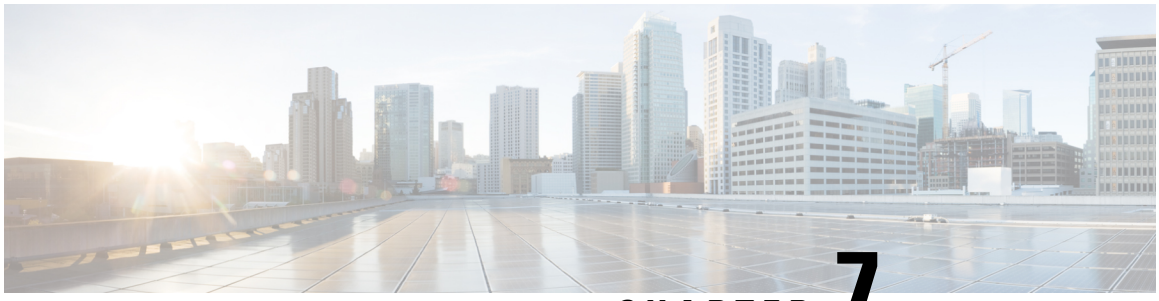
Example:

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

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

What to do next

Install required packages. .



CHAPTER 7

In Service Software Upgrade

This document contains the following topics:

Overview

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

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

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

ISSU execution contains the following phases:

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

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

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

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

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

Restrictions and Usage Guidelines

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

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

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

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

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

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

Pre-installation Tasks

Procedure

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

Example:

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

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

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

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

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

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

Example:

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

RP/0/RP0/CPU0:ios# show platform vm
Node name       Node type       Partner name    SW status       IP address
--------------- --------------- --------------- --------------- ---------------
0/RP1/CPU0      RP (STANDBY)    0/RP0/CPU0      FINAL Band      192.0.4.4
0/RP0/CPU0      RP (ACTIVE)     0/RP1/CPU0      FINAL Band      192.0.0.4

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

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

Example:

```
RP/0/RP0/CPU0:ios# show hw-module fpd
Sun Apr  5 12:47:21.347 CEST
                                                              FPD Versions
                                                              ================
Location   Card type              HWver FPD device       ATR Status   Running Programd
-------------------------------------------------------------------------------------
0/1        A900-IMA8CS1Z-M       0.0   IMFPGA              CURRENT    1.95    1.95
0/5        A900-IMA8Z            0.0   IMFPGA              CURRENT   17.05   17.05
0/9        N560-IMA2C            0.0   IMFPGA              CURRENT    4.80    4.80
```

```
0/10       A900-IMA8Z            0.0   IMFPGA               CURRENT   17.05   17.05
0/RP0      N560-RSP4-E           0.0   ADM                  CURRENT    1.05    1.05
0/RP0      N560-RSP4-E           0.0   IOFPGA               CURRENT    0.56    0.56
0/RP0      N560-RSP4-E           0.0   PRIMARY-BIOS         CURRENT    0.16    0.16
0/RP0      N560-RSP4-E           0.0   SATA                 CURRENT    1.30    1.30
0/RP1      N560-RSP4-E           0.0   ADM                  CURRENT    1.05    1.05
0/RP1      N560-RSP4-E           0.0   IOFPGA               CURRENT    0.56    0.56
0/RP1      N560-RSP4-E           0.0   PRIMARY-BIOS         CURRENT    0.16    0.16
0/RP1      N560-RSP4-E           0.0   SATA                 CURRENT    1.30    1.30
0/FT0      N560-FAN-H            1.0   PSOC                 CURRENT    2.02    2.02

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

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

Example:

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

Media Information for node0_RP1_CPU0.


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

Media Information for node0_RP0_CPU0.


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

************************************************************
Location : 0/RP1
************************************************************
-------------------------------------------------------------------------
Partition                    Size     Used  Percent    Avail
-------------------------------------------------------------------------
rootfs:                      2.4G     616M      28%     1.7G
harddisk:                    5.6G     2.8G      53%     2.5G
log:                         459M     119M      28%     306M
config:                      459M     4.1M       1%     421M
```

```
disk0:                       1011M     1.7M       1%     940M
install:                      5.6G     2.2G      41%     3.1G
install:/tmp                  5.6G     2.2G      41%     3.1G
install:/cache                5.6G     2.2G      41%     3.1G
rootfs:/install/tmp           5.6G     2.2G      41%     3.1G
------------------------------------------------------------------------
rootfs:  = root file system (read-only)
log:     = system log files (read-only)
config:  = configuration storage (read-only)
install: = install repository (read-only)

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

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

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

Example:

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

Fri Apr  3 11:02:01.208 UTC
Apr 03 11:02:03 Install operation 7 started by sanity:
 install add source harddisk:/ 702_ncs560_V2.tar
Apr 03 11:02:10 Install operation will continue in the background
RP/0/RP0/CPU0:tb2-ncs560-pe1#

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

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

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

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

Example:

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

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

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

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

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

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

Example:

For XR VM:

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

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

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

For System Admin VM:

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

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

sysadmin-vm:0_RP0#

sysadmin-vm:0_RP0# show install repository all | in sys|host
ncs560-sysadmin-6.6.3
host-6.6.3
ncs560-sysadmin-7.0.2   extracted V2 image
host-7.0.2              extracted V2 image


sysadmin-vm:0_RP0#
```

ISSU - Single-step Installation

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

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

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

Before you begin

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

Procedure

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

Example:

For System Admin VM:

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

For XR VM:

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

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

Example:

For System Admin VM:

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

For XR VM:

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

Example

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

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

```
host-7.0.2


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

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

sysadmin-vm:0_RP0:#

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

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

sysadmin-vm:0_RP0:#


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

sysadmin-vm:0_RP0:MYISSU#

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

!# Verifying the active package

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

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

!# Commits the latest image

sysadmin-vm:0_RP1# install commit


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

0/RP1/ADMIN0:Oct 31 21:33:02.061 : inst_mgr[6913]:%INFRA-INSTMGR-2-OPERATION_SUCCESS :
```

```
Install operation 3 completed successfully
Wed Oct 31 21:33:02 2018 Install operation 3 completed successfully.
```

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

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

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

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

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

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

Phase           Start-Time      End-Time                  State
-------------------------------------------------------------------------------
Prepare         11:49:40        11:55:26                  Completed
Load            11:56:11        12:06:42                  Completed
Run             12:30:37        12:30:48                  Completed
Cleanup         12:32:46        12:33:46                  Completed
-------------------------------------------------------------------------------
Current Status          : ISSU Orchestration Successfully Completed

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

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

Rack 0   RP0   RP1
```

ISSU - Multi-step Installation

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

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

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

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

Before you begin

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

Procedure

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

Example:

For XR VM:

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

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

Example:

For XR VM:

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

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

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

Example:

For XR VM:

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

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

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

Example:

For XR VM:

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

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

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

Example:

For XR VM:

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

Recovering from a Failed ISSU Operation

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

Abort an ISSU operation manually after the *Prepare* phase

Procedure

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

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

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

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

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

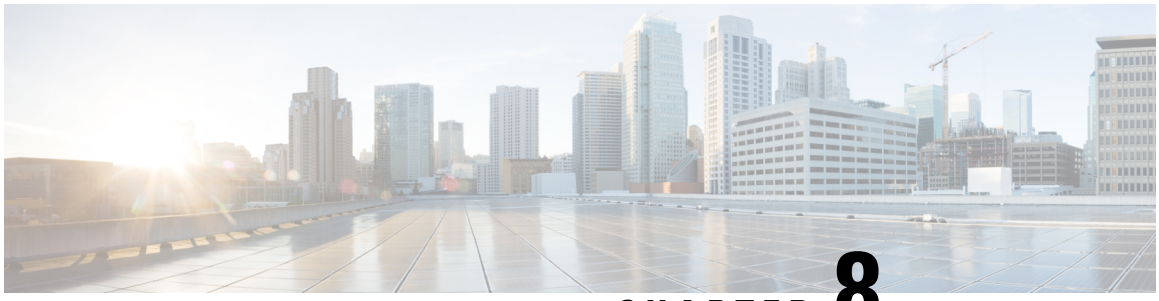
```
RP/0/RP0/CPU0:ios# install activate issu abort cleanup
```

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

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

Installing Packages Using ISSU: Related Commands

Command	Purpose
show issu	Displays the state or status of the ISSU operation. Effective with Cisco IOS XR Release 7.0.2, this command is also supported for System Admin VM ISSU.
admin show issu	Displays the status of the ISSU operation.
show install request	Displays the progress of the ISSU installation.
admin show install request	Displays details about the progress of the install operation.
show install active	Displays the active packages on the system.
install prepare clean	Clears the existing prepared image. If there is a failure in the Prepare phase, run this command to clear the prepared image.
show issu milestone summary	Displays information about the ISSU milestones.

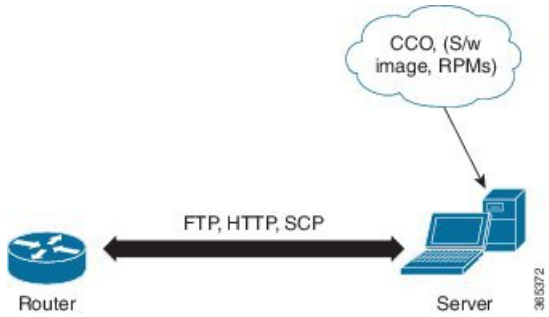


CHAPTER 8

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



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

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

The new commands are **install update** and **install upgrade**. The **install update** command identifies and updates dependent packages, but does not update the base package. The **install upgrade** command upgrades the base package.

Note **1.**
2.

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 add** 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 add** command is:

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

Four scenarios in which you can use the **install add** 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 add source [repository]

- **When a package name is specified**

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

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

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

- **When an SMU is specified**

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

Upgrade Base Software Version

You can upgrade to a newer version of the base software when it becomes available. To upgrade to the latest base software version, use the **install upgrade** 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 upgrade** command is:

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

  For example,

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

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

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

Downgrade an RPM

After an RPM is activated, you may need to downgrade it by activating an RPM of a lower version. Use the **force** option with the **install activate** command to activate an RPM of a lower version.

The syntax of the command is: **install activate[rpm]force**

For example, to add and activate an RPM of a lower version, use the following steps.

Configuration

1. Download the lower version RPM to the .

   RPM currently active: `mpls-2.0.0.0-r60011I`

   RPM to be activated: `mpls-2.0.0.0-r6006I`

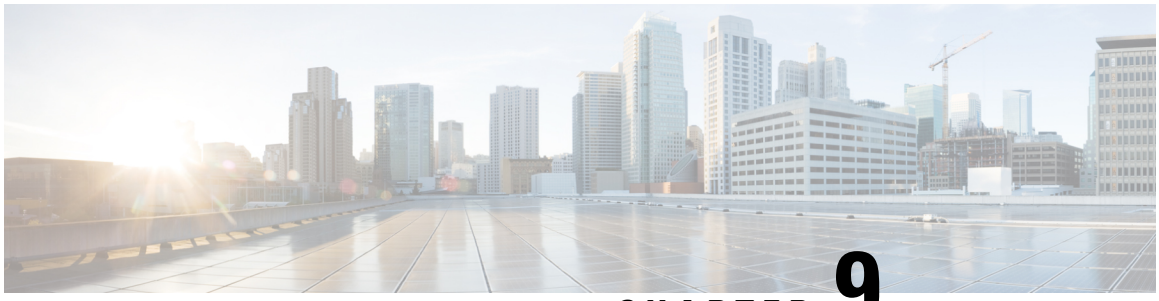
   install add source`[repository] mpls-2.0.0.0-r6006I.rpm`

2. Activate the downloaded RPM.

   install activate`mpls-2.0.0.0-r6006I.rpm force`

   On activation, **mpls-2.0.0.0-r60011I.rpm** is automatically rendered inactive.

You can use the **show install active** command to check the active version of the RPM.



CHAPTER 9

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 80.

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:

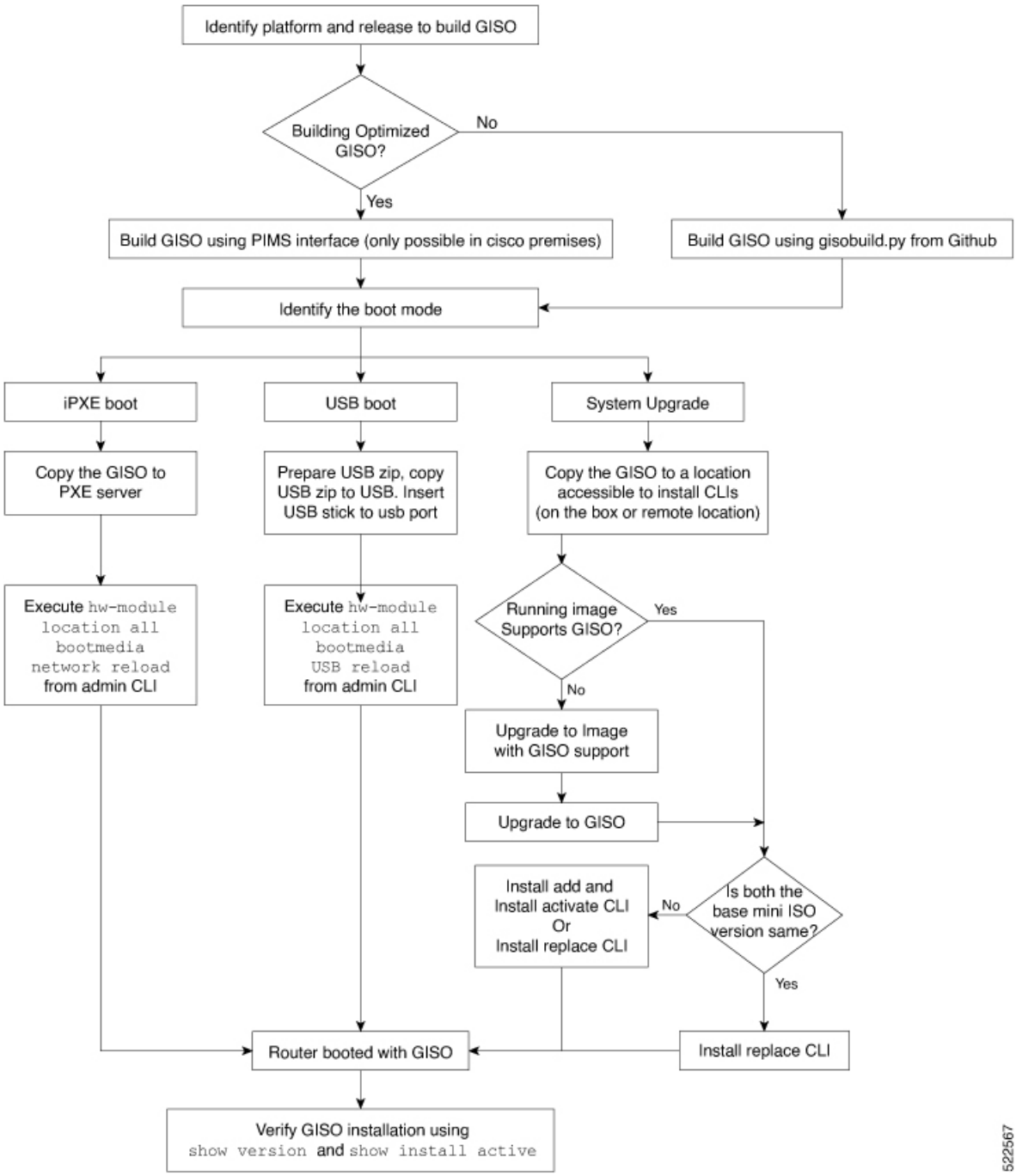
- Building and booting GISO for asynchronous package (a package of different release than the ISO) is not supported.
- Verifying the XR configuration is not supported in the GISO build script `gisobuild.py`.

- 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 4: 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 is dependent on the parent RPM . If only the child RPM is included, the Golden ISO build fails.

Install Golden ISO

Golden ISO (GISO) automatically performs the following actions:

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

Procedure

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

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

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

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

- **System Upgrade:** when the system is upgraded, GISO can be installed using **install add**, **install activate**, or using **install replace** commands.

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

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

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

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

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

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

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

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

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

The options to upgrade the system are as follows:

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

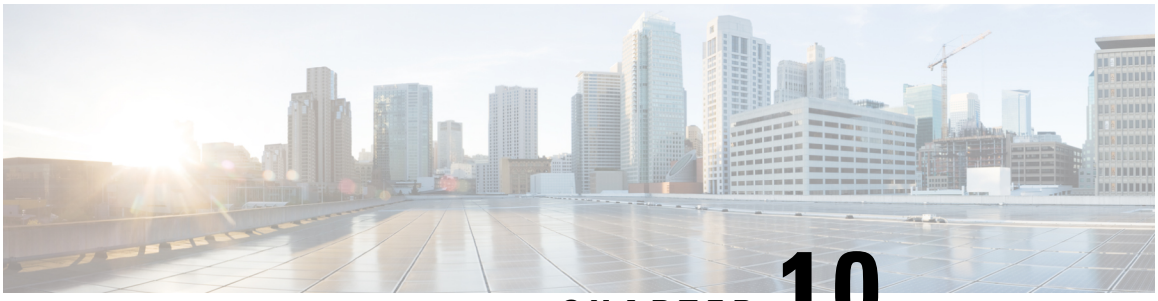
  Using **install replace** command performs a system reload, irrespective of the difference between ISO and the existing version.
- **system upgrade across releases from version1 GISO to version2 GISO:** Both the GISO images have different base versions. Use **install add** and **install activate** commands,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

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

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

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

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



CHAPTER 10

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 (max). USB 2.0 and USB 3.0 are supported.
- Copy the compressed boot file from the software download page at cisco.com to your local machine. The file name for the compressed boot file is in the format *ncs560-usb-boot-<release_number_zip>*.

Procedure

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

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

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

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

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

Step 5 Eject the USB drive from your local machine.

What to do next

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

Boot the Router Using iPXE

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

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

Zero Touch Provisioning

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

ZTP auto provisioning involves:

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

Setup DHCP Server

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

Before you begin

- Consult your network administrator or system planner to procure IP addresses and a subnet mask for the management interface.
- Physical port Ethernet 0 on RP is the management port. Ensure that the port is connected to management network.
- Enable firewall to allow the server to process DHCP packets.
- For DHCPv6, a Routing advertisement (RA) message must be sent to all nodes in the network that indicates which method to use to obtain the IPv6 address. Configure Router-advertise-daemon (radvd, install using yum install radvd) to allow the client to send DHCP request. For example:

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

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

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

Procedure

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

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

- Use MAC address of the router:

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

Ensure that the above configuration is successful.

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

Step 3 Restart DHCP.

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

Example

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

```
allow bootp;
allow booting;
ddns-update-style interim;
option domain-name "cisco.com";
option time-offset -8;
ignore client-updates;
default-lease-time 21600;
max-lease-time 43200;
option domain-name-servers <ip-address-server1>, <ip-address-server2>;
log-facility local0;
 :
subnet <subnet> netmask <netmask> {
  option routers <ip-address>;
  option subnet-mask <subnet-mask>;
  next-server <server-addr>;
}
  :
host <hostname> {
  hardware ethernet e4:c7:22:be:10:ba;
  fixed-address <address>;
  filename "http://<address>/<path>/<image.bin>";
}
```

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

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

What to do next

Invoke ZTP.

Invoke ZTP

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

Before you begin

Ensure that a DHCP server is setup. For more information, see Setup DHCP Server, on page 85.

Procedure

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

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

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

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

With this configuration, the system boots using during installation, and then download and execute when XR VM is up.

Invoke ZTP Manually

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

Before you begin

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

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

```
#"
...
```

Procedure

Step 1 Boot the router.

Step 2 Login manually.

Step 3 Enable interfaces.

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

```
Router#ztp initiate
```

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

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

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

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

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

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

To override the prompt:

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

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

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

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

What to do next

Boot the router using iPXE.

Boot the Router Using iPXE

Before you use the iPXE boot, ensure that:

- DHCP server is set and is running.
- You have logged in to the System Admin console using the **admin** command.

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

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

Example:

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

The following example shows the output of the command:

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


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

Disaster Recovery Using Manual iPXE Boot

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

Procedure

Step 1 Use the arrow keys (up, down) to select **UEFI: Built-in EFI IPXE** to enable iPXE boot. The iPXE boot launches the auto boot.

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

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

Example:

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

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

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