Cisco Nexus 3000 Series NX-OS Software Upgrade and Downgrade Guide, Release 9.2(x)

First Published: 2018-07-16

Last Modified: 2021-02-22

CONTENTS

Preface

This preface includes the following sections:

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which you supply the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x	y]	Square brackets enclosing keywords or arguments that are separated by a vertical bar indicate an optional choice.
{x	y}	Braces enclosing keywords or arguments that are separated by a vertical bar indicate a required choice.
[x {y	z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
variable	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string includes the quotation marks.

Examples use the following conventions:

Convention	Description
screen font	Terminal sessions and information the switch displays are in screen font.
boldface screen font	Information that you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Related Documentation for Cisco Nexus 9000 Series Switches

The entire Cisco Nexus 9000 Series switch documentation set is available at the following URL:

http://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus9k-docfeedback@cisco.com. We appreciate your feedback.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.

- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.

- To submit a service request, visit Cisco Support.

- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.

- To obtain general networking, training, and certification titles, visit Cisco Press.

- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

CHAPTER **1**

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 3000 Series NX-OS Software Upgrade and Downgrade Guide, Release 9.x*.

- Topic 1, on page 1
- New and Changed Information, on page 1

Topic 1

New and Changed Information

Table 1: New and Changed Features for Cisco NX-OS Release 9.x

Feature	Description	Changed in Release	Where Documented
NX-OS Optionality	Added support for NX-OS optionality.	9.2(1)	Optionality in Cisco NX-OS Software, on page 31

Cisco Nexus 3000 Series NX-OS Software Upgrade and Downgrade Guide, Release 9.2(x)

1

Upgrading or Downgrading the Cisco Nexus 3000 Series NX-OS Software

This document describes how to upgrade or downgrade the Cisco NX-OS software on Cisco Nexus 3000 Series switches.

Information About Software Images

Cisco Nexus 3000 Series switches are shipped with the Cisco NX-OS software preinstalled on the switches. Before upgrading or downgrading from an existing image, you should read through the information in this document to understand the guidelines, prerequisites, and procedures for upgrading the software. For updated information about the Cisco NX-OS software for Cisco Nexus 3000 Series switches, see the Cisco Nexus 3000 Series Release Notes.

The Cisco NX-OS software consists of one Cisco NX-OS software image. The image filename begins with "nxos." Only this image is required to load the Cisco NX-OS operating system. This image runs on all the

Cisco Nexus 3000 and 3100 Series switches. Using the **install all** command is the only supported method of upgrading to new releases.

Note There is a single image for booting the Cisco Nexus 3000 Series switches. The kickstart and system images are not required. Due to a single image binary, the following commands are updated:

- **boot nxos** *single_image_binary*

- **install all nxos** *single_image_binary*

Supported Hardware

Cisco Nexus 3000 and 3100 Series switches are shipped with the Cisco NX-OS software preinstalled. Cisco NX-OS upgrades and downgrades are supported on the hardware listed in the following sections:

Cisco Nexus 3000 Series Switches

- Cisco Nexus 3016 switches

- Cisco Nexus 3048 switches

- Cisco Nexus 3064 switches

Cisco Nexus 3100 Series Switches

- Cisco Nexus 3132Q and 3132Q-XL switches

- Cisco Nexus 3172 switches

Note For software upgrade and downgrade information for Cisco Nexus 3000 Series switches that operate in N9K mode, see the Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide.

About ISSU

An in-service software upgrade (ISSU) allows you to upgrade the device software while the switch continues to forward traffic. An ISSU reduces or eliminates the downtime typically caused by software upgrades. You can perform an ISSU, also known as a nondisruptive upgrade, for some switches. (See the "Guidelines and Limitations for ISSU" section for a list of supported platforms.)

ISSU Prerequisites

Follow the guidelines in the "Guidelines and Limitations for ISSU" section to ensure that the ISSU works smoothly.

Make sure that the network is stable and no changes are made while the ISSU is in progress.

Ensure feature compatibility between the current running release and the target release.

Make sure that interfaces are not in a spanning-tree designated forwarding state. Also, make sure that bridge assurance is not configured on any interface. The vPC peer-link is an exception to these requirements.

Verify that the current STP topology is consistent with the ISSU requirements. Use the **show spanning-tree issu-impact** command to display the STP configuration and whether or not there are potential STP issues.

Use the **show lacp issu-impact** command to display if a port or a peer switch is configured in the rate fast mode. The ISSU process is aborted if the system has any LACP fast timers configured.

Guidelines and Limitations for ISSU

The following is a list of important guidelines and limitations for ISSU:

- Non-disruptive standard ISSU on Cisco Nexus 3172PQ, 3172TQ, 3132Q, 3132Q-X, 3064, 3064-X, 3064-T, 3048, 3016 (4 GB low-memory platforms) is supported from Cisco Nexus 7.0(3)I7(7) or later releases to the Cisco NX-OS 9.2(X) release.

- You can perform an ISSU for segment routing or VXLAN.

- Standard ISSUs are supported on the following platforms:

    - Cisco Nexus 3172 switches

    - Cisco Nexus 3132Q and 3132Q-XL switches

    - Cisco Nexus 3064 switches

    - Cisco Nexus 3048 switches

    - Cisco Nexus 3016 switches

- Beginning with Cisco NX-OS Release 7.0(3)I5(1), Cisco Nexus 3000 series platforms with 16 GB of memory or higher support enhanced ISSU. The upgrade will be disruptive.

- The minimum free bootflash space required to perform ISSU is as follows:

    - For compact image: 200 MB

    - For non-compact image: 300 MB

- The minimum free bootflash space required to perform ISSU is 550 MB.

- CLI and the SNMP configuration change requests are denied during ISSU operations.

- STP topology changes are not expected during an ISSU.

- The Guest Shell is disabled during an ISSU, and it is reactivated after the upgrade. During an ISSU, all First-Hop Redundancy Protocols (FHRPs) cause the other peer to become active if the node undergoing the ISSU is active.

- Beginning with Cisco NX-OS Release 9.2(1), a simplified NX-OS numbering format is used for the platforms that are supported in the release. In order to support a software upgrade from releases prior to Release 7.0(3)I7(4) that have the old release format, an installer feature supplies an I9(1) label as a suffix to the actual release during the **install all** operation. This label is printed as part of the image during the install operation from any release prior to 7.0(3)I7(4) to Release 9.2(1), and it can be ignored. See the following example.

- ```
  switch# install all nxos bootflash:nxos.9.2.1.bin
  Installer will perform compatibility check first. Please wait.
  Installer is forced disruptive

  Verifying image bootflash:/nxos.9.2.1.bin for boot variable "nxos".
  [####################] 100% -- SUCCESS

  Verifying image type.
  [####################] 100% -- SUCCESS

  Preparing "nxos" version info using image bootflash:/nxos.9.2.1.bin.
  [####################] 100% -- SUCCESS

  Preparing "bios" version info using image bootflash:/nxos.9.2.1.bin.
  [####################] 100% -- SUCCESS

  Performing module support checks.
  [####################] 100% -- SUCCESS

  Notifying services about system upgrade.
  [####################] 100% -- SUCCESS

  Compatibility check is done:
  Module  bootable    Impact        Install-type  Reason
  ------  --------  ------------  ------------  ------
    1       yes      disruptive      reset       Incompatible image for ISSU

  Images will be upgraded according to following table:
  Module   Image               Running-Version(pri:alt)  New-Version          Upg-Required
  ------  -------  --------------------------------------  --------------------  ------------

    1     nxos                              7.0(3)I7(3)          9.2(1)I9(1)
   yes
    1     bios     v05.31(05/17/2018):v05.26(11/06/2017)   v05.31(05/17/2018)
    no


  Switch will be reloaded for disruptive upgrade.
  Do you want to continue with the installation (y/n)?  [n] y
  ```

- Cisco Nexus 3048, 3064, 3132, and 3172 switches with a model number that does not end in **-XL** must run a compact NX-OS software image due to limited bootflash space. This compact image can be created using the NX-OS Compact Image procedure; alternatively, a compact NX-OS software image can be downloaded directly from Cisco's Software Download website. This requirement does not apply to any other model of Nexus 3000 or 3100 series switch.

  - The MD5/SHA512 checksum published on Cisco's Software Download website for a compact NX-OS software image may not match the MD5/SHA512 checksum of a compact image created through the NX-OS Compact Image procedure.

Compact Image for Cisco Nexus 3000, 3100, and 3500

Early models of Cisco Nexus 3000, 3100, and 3500 Series switches with a model number that does not end in **-XL** have 1.4 to 1.6 gigabytes of storage space allocated to the bootflash. Over time, the file size of NX-OS software images has steadily increased to be over 1 gigabyte. As a result, it is difficult for Nexus 3000, 3100, and 3500 Series switches with a model number that does not end in **-XL** to simultaneously store more than one full NX-OS binary image at a time. Therefore, administrators cannot follow the standard NX-OS software

upgrade procedure on Nexus 3000, 3100, and 3500 Series switches with a model number that does not end in **-XL** that is used for other Nexus platforms, such as Nexus 5000, 6000, 7000, and 9000 Series switches.

Starting with NX-OS software release 7.0(3)I3(1), the file size of NX-OS software images can be reduced through the NX-OS Compact Image procedure. This is a non-disruptive procedure that does not affect the switch's control plane or ability to forward data plane traffic.

Compact NX-OS Software Images on Cisco's Software Download Website

Compact NX-OS software images are available for download on Cisco's Software Download website for a few NX-OS software releases. These compact images have a published MD5/SHA512 checksum that can be used to verify the integrity of the NX-OS binary image file. The compact NX-OS software images can be downloaded from Cisco's Software Download website for the following NX-OS software releases:

- 9.3(4) and later

- 9.2(4)

- 7.0(3)I7(8) and later

Note   The MD5/SHA512 checksum published on Cisco's Software Download website for a compact NX-OS software image may not match the MD5/SHA512 checksum of a compact image created through the NX-OS Compact Image procedure.

Compact Image to be Run for Different Switch Models

The following table describes the appropriate compact image to be run for each applicable model of switch, using NX-OS software release 9.2(4) as an example.

Table 2: Compact Image Table for Each Cisco Nexus 3000, 3100, and 3500 Series Platform

Switch Model Number	Compact NX-OS Software Image Filename
N3K-C3016Q-40GE	n3000-compact.9.2.4.bin
N3K-C3048TP-1GE	n3000-compact.9.2.4.bin
N3K-C3064PQ-10GX	n3000-compact.9.2.4.bin
N3K-C3064TQ-10GT	n3000-compact.9.2.4.bin
N3K-C3064TQ-32T	n3000-compact.9.2.4.bin
N3K-C3132Q-40GE	n3100-compact.9.2.4.bin
N3K-C3132Q-40GX	n3100-compact.9.2.4.bin
N3K-C3172PQ-10GE	n3100-compact.9.2.4.bin
N3K-C3172TQ-10GT	n3100-compact.9.2.4.bin
N3K-C3172TQ-32T	n3100-compact.9.2.4.bin

Switch Model Number	Compact NX-OS Software Image Filename
N3K-C3524P-10G	n3500-compact.9.2.4.bin
N3K-C3524P-10GX	n3500-compact.9.2.4.bin
N3K-C3548P-10G	n3500-compact.9.2.4.bin
N3K-C3548P-10GX	n3500-compact.9.2.4.bin

NX-OS Compact Image Procedure

There are two ways to initiate the NX-OS Compact Image procedure. The following subsections describe each option in further detail.

NX-OS Compact Image Procedure on Bootflash/USB

Starting with NX-OS software release 7.0(3)I3(1), you can use the install all command with the compact option to initiate the NX-OS Compact Image procedure on an image stored in the switch's bootflash or an attached USB drive. This can also be performed on an NX-OS software image that the switch is currently booted from - however, a minimum of 750MB of free space on the switch's bootflash is required to compact the currently booted image.

An example of how to initiate the NX-OS Compact Image procedure on an NX-OS software image stored on the switch's bootflash is as follows:

switch# **install all nxos booftflash:nxos.7.0.3.I7.8.bin compact**

NX-OS Compact Image Procedure Through SCP

Starting with NX-OS software release 7.0(3)I5(2), the NX-OS Compact Image procedure can be initiated while copying the image to the switch's bootflash or an attached USB drive with SCP (Secure Copy Protocol) using an additional option in the copy command. The compact option in the copy command overrides the bootflash space limitation as the image is compacted at the time of transferring the image to the switch's bootflash or an attached USB drive. This option is only supported with SCP - other protocols (such as SFTP [Secure File Transfer Protocol] and FTP [File Transfer Protocol]) are not supported.

An example of how to initiate the NX-OS Compact Image procedure while copying an NX-OS software image to the switch's bootflash through SCP is as follows:

switch# **copy scp://username@192.0.2.100/nxos.7.0.3.I7.8.bin bootflash:nxos.7.0.3.I7.8.bin compact vrf management**

NX-OS Compact Image Procedure Sequence

The sequence of compacting NX-OS software images is important. You cannot compact the currently loaded NX-OS software image if you have another NX-OS software image present on the bootflash due to the limited bootflash size of the switch.

First, you should compact the currently booted image on the bootflash using the NX-OS Compact Image Procedure on Bootflash/USB method previously described. Next, you should copy the desired NX-OS software image to the switch's bootflash using the NX-OS Compact Image Procedure through SCP method previously described, if possible. If this option is not possible, because you are upgrading to an NX-OS software release prior to 7.0(3)I5(2), you should copy the desired NX-OS software image to the switch's bootflash normally,

then use the NX-OS Compact Image Procedure on Bootflash/USB method previously described to compact the image.

The minimum required free space to compact a non-booted image is 450MB of free space on the switch's bootflash. If the free space available on the switch's bootflash is less than 450MB after copying the target image, then the target image needs to be compacted using an attached USB drive or through the NX-OS Compact Image Procedure through SCP method previously described.

NX-OS Compact Image Platform Groups

The NX-OS Compact Image procedure described in this document is applicable to three separate platform groups:

- Nexus 3000 devices (N3K-C3048, N3K-C3064, and so on)

- Nexus 3100 devices (N3K-C3132, N3K-C3172, and so on)

- Nexus 3500 devices (N3K-C3524, N3K-C3548, and so on)

A compacted NX-OS binary image file can be re-used among other devices within the same platform group. However, a compact NX-OS binary image file cannot be used on devices that belong to a different platform group.

For example, consider a scenario where you have four Nexus 3000 devices of the following models:

- N3K-C3048TP-1GE

- N3K-C3064PQ-10GE

- N3K-C3172PQ-40GX

- N3K-C3548P-10G

An NX-OS binary image file compacted on the N3K-C3048TP-1GE can be transferred directly to the bootflash of the N3K-C3064PQ-10GE through a file transfer protocol of your choice (provided there is enough room on the N3K-C3064PQ-10GE). Furthermore, the N3K-C3064PQ-10GE can be upgraded with the use of this compact NX-OS binary image file through a supported method. However, the same compact NX-OS binary image file cannot be used to upgrade the N3K-C3172PQ-40GX and N3K-C3548P-10G devices. The NX-OS Compact Image procedure must be executed on both N3K-C3172PQ-40GX and N3K-C3548P-10G devices separately.

This compatibility between Nexus devices within the same platform group can be used to optimize the NX-OS software upgrade of a large number of devices. For example, if you have 100 N3K-C3048TP-1GE devices, you can use the NX-OS Compact Image procedure on a single device, then transfer the compact NX-OS binary image file to the resulting 99 devices. There is no need to perform the NX-OS Compact Image procedure on all 100 devices.

Booting the Switch from the USB

An option is provided to boot the switch from the USB using a loader prompt. For example:

```
loader> boot usb1:nxos.9.2.1.bin
```

Upgrading the BIOS and Power Sequencer Images

Changes to BIOS and power sequencers are rare; however, when they occur, they are included in the Cisco NX-OS image, and the BIOS and power sequencer are upgraded. The summary displayed by the installer during the installation process indicates the current version of the BIOS and power sequencer and the target version.

> **Note** After a successful power sequence upgrade, you must switch off the power to the system and then power it up.

BIOS Upgrade

See the following guidelines for the BIOS upgrade:

Cisco Nexus 3000 Series platforms require a BIOS upgrade to load Release 7.0(3)I2(1) or later images. Only BIOS versions 3.x.x are compatible with Release 7.0(3)I2(1) and later releases on all Cisco Nexus 3000 Series switches. This is taken care in the regular **install all** method of upgrade but it needs to be explicitly upgraded prior to the fast-reload upgrade.

Only after the BIOS upgrade is complete, it allows the system to perform fast-reload to the newer releases.

BIOS Versions for Each Cisco Nexus 3000 Series Platform

See the following BIOS versions for each Cisco Nexus 3000 Series platform.

Table 3: BIOS Versions Table for Each Cisco Nexus 3000 Series Platform

Sr.No.	Switch Name	Switch Model	Latest BIOS version
1.	Cisco Nexus 3132Q switch	C3132Q-40GE	5.2.0
2.	Cisco Nexus 3132Q-X switch	C3132Q-40GX	5.2.0
3.	Cisco Nexus 3172PQ switch	C3172PQ-10GE	5.2.0
4.	Cisco Nexus 3172CR switch	C3172PQ-10GE	5.2.0
5.	Cisco Nexus 3064-X switch	C3064PQ-10GX	5.2.0
6.	Cisco Nexus 3064-TQ switch	C3064TQ-10GT	5.0.0
7.	Cisco Nexus 3016Q switch	C3016Q-40GE	5.0.0
8.	Cisco Nexus 3064-E switch	C3064PQ-10GE	5.0.0
9.	Cisco Nexus 3064PQ switch	C3064PQ-FA	5.0.0
10.	Cisco Nexus 3048TP switch	C3048TP-1GE	5.0.0

Guidelines for Upgrading in Fast-Reload Scenarios

Procedure

Upgrade to a new release using fast-reload. Using fast-reload after the BIOS upgrade or using **install all** are the only supported methods of upgrading to a Cisco NX-OS 9.x release. For example:

`switch#`**`fast-reload nxos bootflash:nxos.9.2.1.bin`**

Example:

Note   The configuration must be backed up prior to upgrading to a new release as the configuration is required for the downgrade later.

Guidelines for Upgrading in Non-Fast Reload Scenarios

Complete the following steps in the given sequence to upgrade to Cisco NX-OS Release 9.x when you are not using fast-reload:

Procedure

Perform **install all** to upgrade BIOS and also upgrade the NX-OS image to Cisco NX-OS Release 9.x. Upgrade the Cisco NX-OS software using the **install all nxos bootflash:**_filename_ [**no-reload** | **non-disruptive** | **non-interruptive** | **serial**] command.

Example:

`switch#` **`install all nxos bootflash:nxos.9.2.1.bin`**

The following options are available:

- no-reload—Exits the software upgrade process before the device is reloaded.

- non-disruptive—Performs an in-service software upgrade (ISSU) to prevent the disruption of data traffic. (By default, the software upgrade process is disruptive.)

- non-interruptive—Upgrades the software without any prompts. This option skips all error and sanity checks.

- serial—Upgrades the line cards in the system one at a time. (By default, the line cards are upgraded in batches to save time.)

The configuration must be backed up prior to upgrading to a new release as the configuration is required for the downgrade later.

Upgrade Prerequisites

- Ensure that the network is stable and no changes are made while an upgrade is in progress.

- Ensure that you check for feature compatibility between the current running release and the target release.

Management Services After an Upgrade

Before the switch is reset for an upgrade, inband and management ports are brought down and are brought back up after the upgrade completes. Services that depend on the inband and management ports are impacted during this time.

Table 4: Inband and Management Ports Services Impacted During Upgrade Reset

Service	Description
Telnet/SSH	When an upgrade resets the system to load the target Cisco NX-OS version, all Telnet/SSH sessions are disconnected and need to be reestablished after the upgrade completes.
AAA/RADIUS	Applications that leverage the AAA Service (such as login) are disabled during an upgrade, because all Network Management services are disabled during this time, this behavior is consistent.
HTTP	HTTP sessions to the switch are disconnected during an upgrade reboot. After the reboot, the HTTP is restarted and the switch will accept an HTTP sessions.
NTP	NTP sessions to and from the switch are disrupted during an upgrade reboot. After the reboot, NTP session are reestablished based on the saved startup configuration.

Layer-2 Protocols Impact

The following table lists the upgrade impacts to Layer 2 protocols.

Table 5: Upgrade Impact to Layer 2 Protocols

Protocol	Description
LACP	IEEE 802.3ad provides for the default slow aging timers to be transmitted once every 30 seconds in steady state and to expire after 90 seconds. Upgrade should not impact peers that rely on LACP because the recovery time is less than 90 seconds.
IGMP	IGMP does not disrupt existing flows of multicast traffic that are already present, but new flows are not learned (and are dropped) until an upgrade completes. New router ports or changes to router ports are not detected during this time.

Protocol	Description
DCBX and LLDP	DCBX uses LLDP to exchange parameters between peer devices. Because DCBX is a link-local protocol, when the switch undergoes an upgrade, the age time is increased on all ports on the switches that are being upgraded. Manual configurations are ignored during this time.
CDP	During an upgrade, the time-to-live value is increased (180 seconds) if it is less than the recommended timeout value. The configuration is ignored if manually specified.
L2MP IS-IS	Before a switch reboots for an upgrade, the switch transmits L2 IS-IS hellos on all interfaces to prevent neighbor switches from marking routes to the upgrade switch as down. Any topology changes during this time are also not acted upon until the upgrade completes.

Ethernet Interfaces on the Switch

To avoid link down to link up transitions during the control plane outage time, the laser is turned off for administratively up ports that are operationally down. This situation occurs during the upgrade reboot starting state. After the upgrade reboot and a stateful restart, the laser is turned back on. This action prevents the link state from transitioning from down to up during an upgrade.

Pre-Installation Checks

You should do certain sanity checks to ensure that the system is ready for an upgrade and to understand the impact of the upgrade:

- Enter the **show incompatibility** command to verify that the target image is feature-wise compatible with the current image.

- Enter the **show logging level** command to ensure that the severity level for all processes is set to 5 or below.

- Enter the **show install all impact** command to identify the upgrade impact.

- Enter the **install all** command to update to the latest Cisco NX-OS software.

- Review the installer impact analysis and choose to continue.

Note   The switch might reload at this time and cause a traffic disruption.

- Monitor the installation progress.

- Verify the upgrade.

- Enter the **show install all status** command to verify the status of the installation

The following table lists the show commands that identify the impact or potential problems that may occur when performing an upgrade.

Table 6: Upgrade show Commands

Command	Definition
show incompatibility system	Displays incompatible configurations on the current system that will impact the upgrade version.
show logging level	Displays the facility logging severity level configuration. Logging levels for all processes must be set at 5 or below when performing an upgrade. Processes with a logging level greater than 5 are not displayed when you enter the show install all impact command.
show install all impact	Displays information that describes the impact of the upgrade. This command also displays if the upgrade is disruptive or not and if the switch needs to be rebooted and the reason why.

You can also perform the following tasks to identify potential problems before they occur:

- Ensure that you have enough space to store the images on bootflash:
- Display incompatible configurations on the current system that will impact the upgrade version.

```
switch# show incompatibility system pcco.s
No incompatible configurations
```

- Verify the impact of the upgrade.

```
switch# show install all impact nxos bootflash:nxos.9.2.1.bin
```

Information About Fast Reboot

During fast reboot, the image that runs on the CPU reloads the new image and runs it without a CPU or firmware reset. Although there is a brief disruption in traffic during fast reboot, it enables a switch to reload faster than during cold reboot.

Cisco NX-OS software allows you to use fast reboot in a non-interruptive mode. In this mode, fast reboot begins the installation process without any prompts. In this release, fast reboot also supports BGP graceful restart (GR) for compatible peers. You can trigger a fast reboot with graceful restart by using the **trigger-gr** option.

Fast Reboot Timing Requirements

Fast reboot has the following timing requirements for the configurations that it supports:

- Time taken to reset the ASIC and disrupt the data plane after control plane disruption—Less than 90 seconds, when the control plane is disrupted.
- Time taken to resume forwarding traffic—Less than 30 seconds from ASIC reset.

Fast Reboot Guidelines

Fast reboot is supported only with limited configurations and topologies. Some of supported configurations and guidelines are listed in this section. When reloading system software by using the **fast-reload** command, use the supported follow these guidelines:

- Configuration changes—You cannot enter configuration mode during a reload or an upgrade. You should save, commit, or discard any active configuration sessions before upgrading or downgrading the Cisco NX-OS software image. The active configuration session is deleted without a warning during a reload.

  Use the **show configuration session summary** command to verify that there are no active configuration sessions.

  ```
  switch# show configuration session summary
  There are no active configuration sessions
  ```

  Ensure that you check the compatibility of configurations before using the **fast-reload** command.

  > **Note** Do not use the **fast-reload** command for upgrades that may lead to kernel or BIOS changes.

  For more information on configuration sessions, see the Cisco Nexus 3000 Series NX-OS System Management Configuration Guide.

  > **Note** The CLI and SNMP configuration change requests are denied.

- Topology—You should make topology changes such as Spanning Tree Protocol (STP) before you perform an upgrade. You should perform module installations or removals only before or after an upgrade. However, you should not make changes to the Layer 2 and routing topologies, and the default root bridge should not be configured.

- Scheduling—You should upgrade when your network is stable and steady. Ensure that everyone who has access to the switch or the network is not configuring the switch or the network during this time. You cannot configure a switch during an upgrade.

- Space—Verify that sufficient space is available in the location where you are copying the images. The internal bootflash requires approximately 650 MB of free space.

- Hardware—Avoid power interruptions during an installation procedure. Power interruptions can corrupt the software image.

- Connectivity to remote servers—Configure the IPv4 address or IPv6 address for the 10/100/1000 BASE-T Ethernet port connection (interface mgmt0). Ensure that the switch has a route to the remote server. The switch and the remote server must be in the same subnetwork if you do not have a router to route traffic between subnets.

- Link Aggregation Control Protocol (LACP) fast timers—To allow fast-reload, ensure that LACP fast timers are not configured.

- Retrieve compatible images in one of two ways:

    - Locally—Images are locally available on the switch.

- Remotely—Images are in a remote location and you specify the destination using the remote server parameters and the filename to be used locally.

- Command—Use the following commands to prepare for and install the new software:

  - Use the **ping** command to verify connectivity to the remote server.

  - Use the **dir** command to verify the required space is available for the image files to be copied.

  - Use the **show install all impact** command to identify the upgrade impact. This command also displays whether the upgrade is disruptive or the reason why the upgrade is disruptive, whether the switch needs to be rebooted, and the reason why it needs to be rebooted.

> **Note**   We recommended that you log in to the console port to begin the upgrade process.

- Between control plane disruption and data plane disruption, the CPU stops responding.

- Configuration—Fast reboot currently supports the following configuration:

  - BGP v4 and v6

  - 16-way ECMP

  - 48 downlink L2 ports

  - 4 SVIs

  - Less than 10 VLANs

  - 2000 v4 routes and 2000 v6 routes

  - RACLs

  - ARPs

  - STP edge port configuration

- Repaved fast-reload also supports the change in configurations that earlier required a complete reload, for example, portmode profile, URPF enable/disable, and TCAM re-carving.

- Do not use the **fast-reload** option for upgrade to/from a version that has reached EOL.

- The fast reload feature also supports the change in the configurations that earlier required a complete reload, for example, portmode profile, URPF enable/disable, and TCAM re-carving.

- Fast reload does not upgrade the BIOS.

- If you upgrade from a Cisco NX-OS release that does not support the CoPP feature to a release that does support the CoPP feature, you must run the setup utility after the upgrade to enable CoPP on the device.

- Fast reload upgrades are supported from Cisco NX-OS Release 7.0(3)I7(4) or 7.0(3)I7(5) to a Cisco NX-OS 9.x release.

- Beginning with Cisco NX-OS Release 9.3(10), Fast Reload support is "hidden" in the NX-OS CLI, but still available if the command is executed. Fast Reload feature is not in the Release 9.3(x) train going forward.

- Cisco removes any public documentation claiming fast reload performance of under 30 seconds for the Cisco Nexus 3164Q switches.

- If you downgrade to a release lower than Cisco NXOS release 6.0(2)U2(1) using the **install-all** command, fast reload does not work.

Using the Fast-Reload Command

The **fast-reload** command reloads Cisco Nexus 3000 or Nexus 3100 Series switches faster than the **reload** command.

ASCII configuration based fast-reload is also supported in addition to the PSS/binary configuration based fast-reload. The Cisco Nexus 3000 Series switches do not go through a complete reset with fast reload. The boards of Cisco Nexus 3000 and 3100 Series switches have two reset domains. The reset domain 1 contains the CPU, the Platform Controller Hub (PCH), the Management Ethernet controller, the PCI bridge, the OBFL, the USB, and the Fan Controller. The reset domain 2 contains the ASIC, the PHY retimers, the SFP+ modules, and the QSFP modules. Fast reload resets only the reset domain 2 and there is no reset of CPU, reload of BIOS, and firmware.

After you run the **fast-reload** command, the following sequence of events take place:

1. The switch loads the NXOS software image and upgrades the kernel. All applications undergo a stateless cold reboot and they are restarted through the startup configuration.

2. The control plane is disrupted. During control plane disruption, all control protocol communication stops. Control plane disruption is always less than 90 seconds.

3. After the control plane disruption, all control plane applications undergo a stateless cold reboot and do not retain their state. The new configuration is applied when the switch reloads.

4. The data plane is disrupted. Data plane disruption is always less than 30 seconds.

5. On the forwarding plane, all links become unavailable and the data plane does not retain its state after reload. Traffic forwarding is resumed within less than 30 seconds.

Note Ensure that you have a working image and that you analyze the impact of the fast reboot operation before using this command.

Fast Reload in PSS/Binary Configuration

Procedure

Step 1 Log in to the switch.

Step 2 To perform fast reboot on Cisco Nexus 3000 and 3100 Series switches, use the following command: **fast-reload** [**save-config**] [**trigger-gr**] [**nxos bootflash:**_nxos-image-name_] [**non-interruptive**] command to perform a fast reload.

Fast Reload In ASCII Configuration

ASCII configuration based fast-reload is also supported in addition to the PSS/binary configuration based fast-reload.

Note The copy file startup and fast reload are supported only for specific configurations, namely Layer 3 ports with port channels, eBGP, and a few physical l2 ports having SVI only towards the hosts.

To use ASCII-file based fast reload, use the following command:

copy *configuration-file* **startup-config**

fast-reload nxos bootflash:*nxos-image-name*

The *configuration-file* is an ASCII file that contains the system configurations that fast reload uses on upgrade or fast reload. It can be copied from the remote location also. If the NXOS software image is not specified, the image existing on the switch is reloaded. If the NXOS software image provided is a higher version than the existing version, an upgrade is triggered.

Note To ensure that subsequent fast reboot operations, use the NXOS software image as the boot variables, specify the save-config option while running the fast-reload command. If the save-config option is not specified, the fast-reload command does not save the boot variables.

Fast Reload Upgrade

You can perform a faster reload and upgrade to a newer release with minimal data downtime compared to install all command.

The feature is similar to existing fast-reload support on Cisco Nexus 3000 Series switches from Release 6.0(2)U2(1) onwards with a few additional steps to upgrade BIOS before upgrading to Release 7.0(3)I2(1) or later releases.

Caution The fast-reload to Release 7.0(3)I2(1) is supported only from Release 6.0(2)U6(3a) or later releases, after the BIOS has been upgraded. Using fast-reload from Release 6.0(2)U6(3a) to Release 7.0(3)I2(1) without upgrading the BIOS will result in the switch not booting up. See BIOS upgrade for more information.

Note If guest shell or any virtual-services are enabled, the **install all** command should be used. Use the **show virtual-service list** command to verify the presence or the state of the guest shell or any virtual services..

Enabling BGP Graceful Restart with Fast Reboot

Cisco NX-OS software allows you to enable BGP graceful restarts (GR) with fast reboot. You can now use the **fast-reload trigger-gr** command to enable BGP GR. Use this command only when all BGP peers are GR-capable.

To enable BGP GR with fast reboot on Cisco Nexus 3000 and 3100 Series switches, use the following command:

fast-reload [**save-config**] [**trigger-gr**] [**nxos bootflash**:*nxos-image-name*] [**non-interruptive**] command to perform a fast reload.

```
switch# fast-reload trigger-gr nxos bootflash:nxos.9.2.1.bin non-interruptive
```

Upgrading and Downgrading Using Fast Reboot

Upgrading Using Fast Reboot

You can upgrade the software on a switch by using fast reboot. To upgrade, you must specify the NXOS software image.

Before You Begin

Ensure that the version of the NXOS software image specified in the fast-reload command is higher than the version of the image currently existing on the switch.

```
switch# fast-reload nxos bootflash:nxos.9.2.1.bin
```

Downgrading Using Fast Reboot

Downgrading the system software by using fast reboot is not supported. To downgrade the image software, use the **install all** command instead of fast reboot.

Using the Install All Command

The **install all** command triggers a disruptive software install on Cisco Nexus 3000 and Nexus 3100 Series switches. The following images are upgraded during the installation:

- The NXOS software image
- System BIOS
- Power sequencers on the system

The **install all** command provides the following benefits:

- You can upgrade the Cisco Nexus 3000 Series switches by using just one command.
- You can receive descriptive information about the intended changes to your system before you continue with the installation. For example, it identifies potential disruptive upgrades.
- You can continue or cancel the upgrade when you see this question (the default is no):

  ```
  Do you want to continue (y/n) [n]: y
  ```

- You can also use the **install all non-interruptive** command to install a new image without any prompts.

- The command automatically checks the image integrity, which includes the NXOS software image.

- The command performs a platform validity check to verify that a wrong image is not used.

- Pressing Ctrl + C gracefully ends the **install all** command. The command sequence completes the update step in progress and returns to the EXEC prompt.

- After entering the **install all** command, if any step in the sequence fails, the upgrade ends.

- The following message appears to warn you about the impact of upgrading the power sequencer:

  ```
  Warning: please do not remove or power off the module at this time.

  Note: Power-seq upgrade needs a power-cycle to take into effect.
  ```

Note After a successful power sequence upgrade, you must switch off the power to the system and then power it up.

Using the Install All Non-Interruptive Command

Cisco NX-OS software supports the use of non-interruptive **install all** command. You can now use the **install all non-interruptive** command to install a new image without any prompts.

To perform a non-interruptive install all on Cisco Nexus 3000 and 3100 Series switches, use the following command:

install all nxos bootflash: [*nxos-image-name*] [**non-interruptive**]

```
switch# install all nxos bootflash:nxos.9.2.1.bin
```

Upgrading Procedures

The upgrade process is triggered when you enter the **install all** command. This section describes the sequence of events that occur when you upgrade a single Cisco Nexus 3000 Series switch.

Note If you have a release prior to Release 7.0(3)I2(1), upgrade to Cisco Nexus 3000 Release 6.0.2.U6(3a) first and then upgrade to Release 7.0(3)I2(1) or later releases.

Note During the compatibility check, the following ISSU-related messages might appear in the Reason field:

Table 7: ISSU- related messages

Reason Field Message — in Cisco NX-OS Release 7.0(3)I3(1)	Reason Field Message — in Cisco NX-OS Release 7.0(3)I4(1) or a Later Release	Description
Incompatible image	Incompatible image for ISSU	The Cisco NX-OS image to which you are attempting to upgrade does not support ISSU.
Hitless upgrade is not supported	Default upgrade is not hitless	By default, the software upgrade process is disruptive. You must configure the non-disruptive option to perform an ISSU.

Installation At-A-Glance

The following table shows an overview of the upgrade process.

Table 8: Upgrade Process At-A-Glance

Upgrade Preparation	1. Log into the first Cisco Nexus 3000 Series switch. We recommend that you log into the console port. In vPC topologies, the first upgrade can be performed on either the primary or secondary switch in the topology.
	2. Log into Cisco.com to access the Software Download Center. To log into Cisco.com, go to https://www.cisco.com and click Log In at the top of the page. Enter your Cisco username and password.
	3. Choose and download the software image to the server.
	4. Verify that the required space is available in the bootflash: directory for the image file(s) to be copied.
	5. If you need more space in the bootflash: directory, delete unnecessary files to make space available.
	6. Copy the Cisco NX-OS software image to the bootflash using a transfer protocol such as ftp:, http:, https:, tftp:, scp:, or sftp. Example: ```switch# copy scp://user@scpserver.cisco.com//download/nxos.9.3.1.bin bootflash:nxos.9.3.1.bin```
	7. Compare the file sizes of the images that were transferred using the **dir bootflash** command. The file sizes of the images obtained from https://www.cisco.com and the image sizes of the transferred files should be the same.
	8. Complete these steps for each switch in the topology.
Pre-upgrade Checks	1. Enter the **show incompatibility** command to verify that the target image is feature-wise compatible with the current image.
	2. Enter the **show install all impact** command to identify the upgrade impact.

Upgrade Begins	1.	Enter the **install all** command to update to the latest Cisco NX-OS software.
	2.	Peruse the installer impact analysis and accept to proceed.
	3.	The installer upgrades the software.
Upgrade Verification	1.	Enter the **show install all status** command to verify the status of the installation.

Copying the Running Configuration from an External Flash Memory Device

You can copy configuration files from an external flash memory device.

Before you begin

Insert the external flash memory device into the active supervisor module.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **dir** {**usb1**:|**usb2**:} [*directory/*] **Example:** `switch# dir usb1:`	Displays the files on the external flash memory device.
Step 2	**copy** {**usb1**:|**usb2**:} [*directory/*] *filename* {**bootflash:**} [*directory/*] *filename* **Example:** `switch# copy usb1:pcco.k bootflash:pcco.k`	Copies the image from an external flash memory device into the bootflash. The filename argument is case sensitive.

Step 3	**copy** {**usb1**:|**usb2**:} [*directory/*] *filename* **running-config**
	
	Example:
	`switch# copy usb1:dsn-config.cfg`
	`running-config`
		Copies the running configuration from an external flash memory device. The filename argument is case sensitive.

Step 4	(Optional) **show running-config** **Example:** `switch# show running-config`	Displays the running configuration.

Step 5	(Optional) **copy running-config startup-config**
	
	Example:
	`switch# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Copying the Startup Configuration from an External Flash Memory Device

You can recover the startup configuration on your Cisco NX-OS device by downloading a new startup configuration file saved on an external flash memory device.

Before you begin

Insert the external flash memory device into the active supervisor module.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **dir**{**usb1**:|**usb2**:}[*directory/*] **Example:** switch# **dir usb1:**	Displays the files on the external flash memory device.
Step 2	**copy**{**usb1**:|**usb2**:}[*directory/*] *filename*{**bootflash:**}[*directory/*]*filename* **Example:** switch# **copy usb1:pcco.k bootflash:pcco.k.**	Copies the image from an external flash memory device into the bootflash. The filename argument is case sensitive.
Step 3	**copy**{**usb1**:|**usb2**:}[*directory/*] *filename* **startup-config** **Example:** switch# **copy usb1:dsn-config.cfg startup-config**	Copies the startup configuration from an external flash memory device. The filename argument is case sensitive.
Step 4	(Optional) **show startup-config** **Example:** switch# **show startup-config**	Displays the startup configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Upgrade Process in a Non-vPC Topology

The following list summarizes the upgrade process in a non-vPC topology:

1. The **install all** command triggers the installation upgrade.

2. The compatibility checks display the impact of the upgrade.

3. The installation proceeds or not based on the upgrade impact.

4. The current state is saved.

5. The system unloads and runs the new image.

6. The stateful restart of the system software and application occurs.

7. The installer resumes with the new image.

8. The installation completes.

The following example displays the upgrade process:

```
switch# install all nxos bootflash:<nxos-image-name>
```

Upgrade Process for vPCs

Upgrade Process for a vPC Topology on the Primary Switch

The following list summarizes the upgrade process on a switch in a vPC topology that holds either the Primary or Operational Primary vPC roles. Steps that differ from a switch upgrade in a non-vPC topology are in bold.

> **Note** In vPC topologies, the two peer switches must be upgraded individually. An upgrade on one peer switch does not automatically update the vPC peer switch.

1. **The install all command issued on the vPC primary switch triggers the installation upgrade.**

2. The compatibility checks display the impact of the upgrade.

3. The installation proceeds or not based on the upgrade impact.

4. **The configuration is locked on both vPC peer switches.**

5. The current state is saved.

6. The system unloads and runs the new image.

7. The stateful restart of the system software and application occurs.

8. The installer resumes with the new image.

9. The installation is complete.

When the installation is complete, the vPC primary switch is upgraded.

> **Note** The vPC primary switch is running the upgraded version, and the vPC secondary switch is running the original software version.

Upgrade Process for a vPC Topology on the Secondary Switch

The following list summarizes the upgrade process on a switch in a vPC topology that holds either the Secondary or Operational Secondary vPC roles. Steps that differ from a switch upgrade in a non-vPC topology are in bold.

1. **The install all command issued on the vPC secondary switch triggers the installation upgrade.**

2. The compatibility checks display the impact of the upgrade.

3. The installation proceeds or not based on the upgrade impact.

4. The current state is saved.

5. The system unloads and runs the new image.

6. The stateful restart of the system software and application occurs.

7. The installer resumes with the new image.

8. **The configuration is unlocked on the primary and secondary switches.**

9. The installation is complete.

vPC Upgrade and Downgrade Procedure for Nexus 9000 -R series switches

In vPC topologies, the two peer switches usually must be upgraded individually. An upgrade on one peer switch does not automatically update the vPC peer switch.

However, Cisco NX-OS Releases 7.0(3)F3(3c) and 7.0(3)F3(4) are not compatible with Cisco NX-OS Release 9.2(x) for vPC peer switches. Both vPC peers must be upgraded simultaneously to Cisco NX-OS Release 9.2(x) to avoid one switch running a 7.0(3)F3(x) release and the other switch running 9.2(x). Optionally, if the switches are being upgraded from Cisco NX-OS Release 7.0(3)F3(4), you can use the following procedure to minimize the traffic impact during upgrade.

> **Note**   This procedure not to be used on Broadcom or Cloudscale-based switches.

1. Switch A and B are running a Cisco NX-OS release. Switch A is the primary switch, and switch B is the secondary switch. Use the **copy r s** command on both switches.

```
primary_switch# show vpc role
vPC Role status
----------------------------------------------------
vPC role : primary
vPC system-mac : 00:23:04:ee:be:64
vPC system-priority : 32667
vPC local system-mac : 70:df:2f:eb:86:1f
vPC local role-priority : 90
vPC peer system-mac : 70:df:2f:eb:1c:ab
vPC peer role-priority : 100
primary_switch#

secondary_switch# show vpc role
vPC Role status
----------------------------------------------------
vPC role : secondary
vPC system-mac : 00:23:04:ee:be:64
vPC system-priority : 32667
vPC local system-mac : 70:df:2f:eb:1c:ab
vPC local role-priority : 100
vPC peer system-mac : 70:df:2f:eb:86:1f
vPC peer role-priority : 90
```

```
secondary_switch#


primary_switch# copy r s v
[#####################################] 100%
Copy complete.

secondary_switch# copy r s v
[#####################################] 100%
Copy complete.
```

2. Bring down the peer link (PL) on the primary switch. The secondary switch brings down its vPC legs.

```
primary_switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
primary_switch(config)# int port-channel 100
primary_switch(config-if)# shutdown

Reload the secondary switch with Release 9.2.1 image (change bootvar /reload)

secondary_switch(config)# boot nxos nxos.9.2.1.bin
Performing image verification and compatibility check, please wait....
secondary_switch(config)#
secondary_switch(config)# copy r s v
[#####################################] 100%
Copy complete.


secondary_switch# reload
This command will reboot the system. (y/n)? [n] y


After reload
-------------------
secondary_switch# show vpc
Legend:
(*) - local vPC is down, forwarding via vPC peer-link
vPC domain id : 100
Peer status : peer link is down
vPC keep-alive status : peer is alive
Configuration consistency status : failed
Per-vlan consistency status : success
Configuration inconsistency reason: Consistency Check Not Performed
Type-2 inconsistency reason : Consistency Check Not Performed
vPC role : none established
Number of vPCs configured : 20
Peer Gateway : Enabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Disabled (due to peer configuration)
Auto-recovery status : Disabled
Delay-restore status : Timer is off.(timeout = 90s)
Delay-restore SVI status : Timer is off.(timeout = 10s)
Operational Layer3 Peer-router : Disabled
vPC Peer-link status
---------------------------------------------------------------------
id Port Status Active vlans
-- ---- ------ ------------------------------------------------
1 Po100 down -

secondary_switch#

primary_switch(config-if)# show vpc
Legend:
```

```
(*) - local vPC is down, forwarding via vPC peer-link
vPC domain id : 100
Peer status : peer link is down
vPC keep-alive status : peer is alive
Configuration consistency status : success
Per-vlan consistency status : success
Type-2 consistency status : success
vPC role : primary
Number of vPCs configured : 20
Peer Gateway : Enabled
Peer gateway excluded VLANs : -
Dual-active excluded VLANs and BDs : -
Graceful Consistency Check : Enabled
Auto-recovery status : Enabled, timer is off.(timeout = 240s)
Operational Layer3 Peer-router : Disabled
vPC Peer-link status
---------------------------------------------------------------------
id Port Status Active vlans
-- ---- ------ -------------------------------------------------
1 Po100 down -
```

3. Configure vPC auto-recovery under the vPC domain on the secondary switch. Enable **vpc upgrade** (exec command).

```
secondary_switch(config)# vpc domain 100
secondary_switch(config-vpc-domain)# auto-recovery
secondary_switch(config-vpc-domain)# end

secondary_switch# show running-config vpc
!Command: show running-config vpc
!Running configuration last done at: Wed May 16 06:34:10 2018
!Time: Wed May 16 06:34:14 2018
version 9.2(1) Bios:version 01.11
feature vpc
vpc domain 100
peer-switch
role priority 100
peer-keepalive destination 10.1.31.30 source 10.1.31.29
delay restore 90
peer-gateway
auto-recovery
ipv6 nd synchronize
ip arp synchronize
interface port-channel100
vpc peer-link
interface port-channel2001
vpc 101


secondary_switch# show vpc upgrade
vPC upgrade : TRUE
SVI Timer : 0
Delay Restore Timer : 0
Delay Orphan Port Timer : 0
secondary_switch#

secondary_switch# show vpc upgrade   >> Hidden command
vPC upgrade : FALSE
SVI Timer : 10
Delay Restore Timer : 90
Delay Orphan Port Timer : 0
```

```
secondary_switch# vpc upgrade   >> Hidden command
```

4. After Layer 3 routes are learned on the secondary switch, reload the primary switch with the new release image. The secondary switch takes over the primary role and brings up its vPC legs in approximately 5 seconds.

```
primary_switch(config)# show boot
Current Boot Variables:
sup-1
NXOS variable = bootflash:/nxos.9.2.1.bin
No module boot variable set
Boot Variables on next reload:
sup-1
NXOS variable = bootflash:/nxos.9.2.1.bin

No module boot variable set
primary_switch(config)# end

primary_switch# show boot
Current Boot Variables:
sup-1
NXOS variable = bootflash:/nxos.9.2.1.bin
No module boot variable set
Boot Variables on next reload:
sup-1
NXOS variable = bootflash:/nxos.9.2.1.bin

No module boot variable set
primary_switch# reload
This command will reboot the system. (y/n)? [n] y

secondary_switch# show vpc
Legend:
(*) - local vPC is down, forwarding via vPC peer-link
vPC domain id : 100
Peer status : peer link is down
vPC keep-alive status : peer is not reachable through peer-keepalive
Configuration consistency status : failed
Per-vlan consistency status : success
Configuration inconsistency reason: Consistency Check Not Performed
Type-2 inconsistency reason : Consistency Check Not Performed
vPC role : primary
Number of vPCs configured : 20
Peer Gateway : Enabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Disabled (due to peer configuration)
Auto-recovery status : Enabled, timer is off.(timeout = 240s)
Delay-restore status : Timer is off.(timeout = 0s)
Delay-restore SVI status : Timer is off.(timeout = 0s)
Operational Layer3 Peer-router : Disabled
vPC Peer-link status
---------------------------------------------------------------------
id Port Status Active vlans
-- ---- ------ -------------------------------------------------
1 Po100 down -
vPC status
```

5. When the primary switch comes back up, the peer link on it is operationally up.

```
primary_switch# show vpc
Legend:
(*) - local vPC is down, forwarding via vPC peer-link
vPC domain id : 100
Peer status : peer adjacency formed ok
vPC keep-alive status : peer is alive
Configuration consistency status : success
Per-vlan consistency status : success
Type-2 consistency status : success
vPC role : primary, operational secondary
Number of vPCs configured : 20
Peer Gateway : Enabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled
Auto-recovery status : Disabled
Delay-restore status : Timer is off.(timeout = 90s)
Delay-restore SVI status : Timer is off.(timeout = 10s)
Operational Layer3 Peer-router : Disabled
vPC Peer-link status
---------------------------------------------------------------------
id Port Status Active vlans
-- ---- ------ -------------------------------------------------
1 Po100 up 1,101-400
```

For downgrade, reload both switches at the same time.

Monitoring the Upgrade Status

The following table lists the **show** commands that are used to monitor installation upgrades.

Command	Definition
show install all failure-reason	Displays the applications that failed during an installation and why the installation failed.
show install all status	Displays a high-level log of the installation.
show tech-support	Displays the system and configuration information that you can provide to the Cisco Technical Assistance Center when reporting a problem.

Downgrading from a Higher Release

The procedure for entering the **install all** command to downgrade the switch is identical to using the **install all** command for a switch upgrade, except that the image files to be loaded are for an earlier release than the image that is currently running on the switch. You can use the **show incompatibility system command** to ensure that there are no feature incompatibilities between the current release and the target release. Note that downgrades are disruptive.

Note	Before you downgrade to a specific release, check the release notes for the current release installed on the switch, to ensure that your hardware is compatible with the specific release. See the Cisco *Nexus 3000 Series Switch Release Notes* for details.

Downgrading from a Higher Release to a Lower Release

This section contains an example for downgrading from a higher release to a lower release.

Caution	Make sure that you store the configuration file for later use.

Complete the following steps to downgrade:

Procedure

Step 1   Enter the **write erase** command.

Step 2   Enter the **write erase boot** command.

Step 3   Enter the **copy** *Release 6.0(2)U6(3a)-config* **startup-config** command.

```
switch# copy downgrade startup-config
```

Step 4   Enter the **install all kickstart** *img.kick system img.sys* **no-save bios-force** command

Step 5   To verify whether the boot variables exist, enter the **show boot** command. If the boot variables do not exist, enter the following set of commands to update the boot variables manually:

```
switch# configure t
switch (config)# boot kickstart <img.kick>
switch (config)# boot system <img.sys>
switch (config)# write run start
```

Note	If lockup occurs on the switch, do a tftp boot.

Troubleshooting Installations

Some common causes for upgrade failure are as follows:

- The bootflash: does not have enough space to accept the updated image.

- The hardware is installed or removed while the upgrade is in process.

- A power disruption occurs while an upgrade is in progress.

- The entire path for the remote server location is not specified accurately.

CHAPTER 3

Optionality in Cisco NX-OS Software

This chapter describes optionality in Cisco NX-OS software.

Optionality in Cisco NX-OS Software

Beginning with Cisco NXOS Release 9.2(1), Cisco NX-OS software image supports modular package management. Cisco NX-OS software now provides flexibility to add, remove, and upgrade the features selectively without changing the base NX-OS software.

The advantages for using modular Cisco NX-OS software are:

- Lean NX-OS software

- Asynchronous delivery of the features and the fixes: Quick fixes are provided that are independent of the releases, including new features.

- Reduced footprint of binaries and libraries at run time

Cisco NX-OS software is provisioned to boot the NX-OS software in two modes as described in the following illustration:

- Base NX-OS mode

- Full NX-OS mode

Figure 1: Optionality in Cisco NX-OS Software
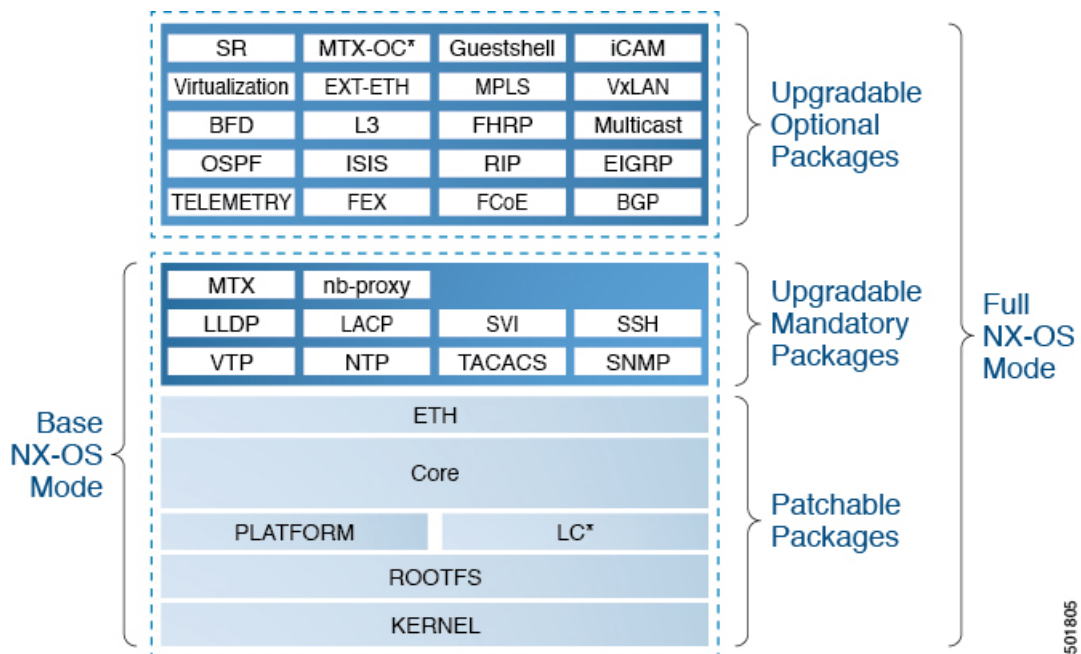
- Base NX-OS mode contains:

  - Upgradable mandatory packages

  - Patchable packages

- Full NX-OS mode contains:

  - Upgradable optional packages

  - Upgradable mandatory packages

  - Patchable packages

Note The default mode is full NX-OS mode.

In base NX-OS mode, basic Layer 2 and Layer 3 features are available. All dynamic routing features (for example, BGP, OSPF, EIGRP, RIP, and ISIS) and other optional feature RPMs are not available by default. You have to install the optional feature RPMs on top of the base image.

In full NX-OS mode, all feature RPMs are installed during boot time when Ethernet plugin is activated by the plugin manager. There is no change in the user behavior as compared to the previous releases.

Using Modular Packages

The Cisco NX-OS software image is traditionally constructed with the packaging that forms a Cisco Linux distribution. It makes upgrading certain packages difficult as each package is large in size.

This section describes a new package management for the Cisco NX-OS software image. Beginning with Cisco NX-OS Release 9.2(1), some NXOS features are considered as optional, for example, BGP, OSPF, VXLAN, MPLS, Segment Routing.

Each modular package has the following important characteristics:

- Upgrade functionality: The modular packages can be independently upgraded. The modular packages should be used from the same release as performing upgrades on these packages across multiple releases is not supported.

- Optionality: The modular packages are optional, for example, these packages can be removed or uninstalled at run time. The removal of the modular packages does not affect bringing-up the system and it does not affect any other functionality of the switches.

Note All APIs exported by the modular package should be used only after the installation of the feature.

RPM and YUM

RPM (Red Hat Package Manager) is the package management system used for packaging in the Linux Standard Base (LSB). The RPM command options are grouped into three subgroups for:

- Querying and verifying packages

- Installing, upgrading, and removing packages

- Performing miscellaneous functions

rpm is the command name for the main command that is used with RPM, whereas .rpm is the extension that is used for the RPM files.

YUM (Yellowdog Updater, Modified) is an open source command-line tool for RPM based Linux systems. It allows users and system administrators to easily install, update, remove, or search software packages on the systems. YUM adds the automatic updates and the package management, including dependency management, to the RPM systems. In addition to understanding the installed packages on a system, YUM works with the repositories that are collections of the packages and they are typically accessible over a network connection.

List of Cisco NX-OS Software Packages

The Cisco NX-OS software image consists of the third party packages.

- Upgradable optional packages

- Upgradable mandatory packages

- Patchable packages

Upgradable Optional Packages

Each upgradable optional package has the following important characteristics:

- It can be independently upgraded.

- These packages are optional, for example, these packages can be removed or uninstalled at runtime. The removal of the upgradable optional packages does not affect bringing-up the system and it does not affect any other functionality on the switches.

> **Note** Use all the APIs that are exported by the optional package only after detecting the presence of the feature.

- These packages can be upgraded, downgraded, activated, or deactivated.

The upgradable optional packages contain some of the following items:

- BGP
- BFD
- EIGRP
- Ext-Eth
- FCoE
- FEX
- FHRP
- Guestshell
- ISIS
- L3
- MPLS
- MTX-OC
- Multicast
- OPENSSH
- OSPF
- RIP
- SR
- TACACS+
- Telemetry
- Virtualization
- VXLAN

Upgradable Mandatory Packages

The mandatory packages can only be upgraded or downgraded, but they cannot be deactivated. The status of the package can be active/install or inactive/uninstall. The upgradable mandatory packages contain the following:

- LACP

- LLDP

- MTX

- nb-proxy

- NTP

- SNMP

- SSH

- SVI

- TACACS

- VTP

Patchable Packages

The patchable packages contain the following:

- Kernel

- LC

- Network-infra (aka Eth)

- Platform

- Rootfs

- System-infra (aka Core)

Booting the NX-OS Image in Base or Full Mode

You can now boot the NX-OS image in base or full mode. The full boot mode installs the complete NX-OS software which is similar to the software of the previous releases. This is the default boot mode. The base boot mode has no optional RPMs installed.

To use the command line option, see the following steps:

- Use the **install reset nxos base** option to install the NX-OS image in the base boot mode using the VSH prompt. After reload, the switch is in the base mode with no optional packages installed.

- Use the **install reset nxos full** option to install the NX-OS image in the full boot mode using the VSH prompt. After reload, the switch is in the full mode with the optional packages automatically installed.

For more information, see Using Install CLIs for Feature RPM Operation section.

Support for ISSU

Beginning with Cisco NX-OS Release 9.2(1), there is no change in the ISSU on Cisco Nexus 9000 Series switches.

Information About RPMs

RPMs can be upgraded or downgraded to a new software version using NXOS install commands or by using YUM commands. An upgradable RPM can be optional or mandatory.

See the following sections for more information about optional and mandatory RPMs.

Optional RPMs and Their Associated Features

The optional RPMs are the RPMs that can be installed to enable the features without affecting the native NXOS behavior or they can be removed using the **install deactivate** command from the switch.

Optional RPMs, for example, EIGRP are not a part of the base software. They can be added, upgraded, and removed as required using either **yum** or **install** CLI commands from the switch.

See the following list of the optional RPMs and their associated features:

Table 9: List of Optional RPMs and Their Associated Features

Package Name	Associated Features
BGP	feature bgp
BFD	feature bfd
Container-tracker	feature container-tracker
EIGRP	feature eigrp
Ext-Eth	• feature openflow • feature evb • feature imp • feature netflow • feature sla_sender • feature sla_responder • feature sla twamp-server • feature sflow
FCoE	• feature-set fcoe • feature-set fcoe-npv

Package Name	Associated Features
FEX	feature-set fex
FHRP	• feature hsrp • feature vrrpv3
iCAM	feature icam
ISIS	feature isis
MPLS	• feature mpls segment-routing • feature mpls evpn
Multicast	• feature pim • feature pim6 • feature msdp • feature ngmvpn
OSPF	• feature ospf • feature ospfv3
RIP	feature rip
Services	feature catena
SR	feature mpls segment-routing traffic-engineering
TELEMETRY	feature telemetry
Virtualization	NA
VXLAN	• feature nv overlay • feature fabric forwarding

Guidelines for NX-OS Feature RPM Installation

See the following NX-OS system RPM repositories that are present in the Cisco NX-OS Series switches for the RPM management.

Note Avoid manually copying the RPMs to system repositories. Instead use the install or YUM commands.

Table 10: RPM Repositories That Are Present in the Switches

Repository Name	Repository Path	Description
groups-repo	/rpms	Part of the bundled NX-OS image. It is used to keep all the RPMs that are bundled as part of the NX-OS image. All RPMs based in this repository are known as base RPMs.
localdb	/bootflash/.rpmstore/patching/localrepo	Used for RPM persistency. When a user adds a NX-OS feature RPM as part of **install add** command, the RPM is copied to this location and it is persisted during the reloads. User has the responsibility to clean the repository.
		To add a RPM to this repository, use **install add** command.
		To remove a RPM from this repository, use **install remove** command.
		YUM commands can be used to populate the repository too.
		The maximum space for the repository is 200Mb along with the patching repository for Cisco Nexus 9000 Series switches except Cisco Nexus 3000 Series switches. For Cisco Nexus 3000 Series switches, the maximum space for the repository is 20 Mb only.
patching	/bootflash/.rpmstore/patching/patchrepo	Used for RPM persistency. When a user adds a NX-OS patch RPM to the switch, the patch RPM is copied to this repository.
thirdparty	/bootflash/.rpmstore/thirdparty	Used for RPM persistency when a user adds a third party RPM.

The **groups-repo** and **localdb** repositories hold the NX-OS feature RPMs that should be installed during the system boot or during activation. YUM commands or **install** command can be used for the installation or the removal of these RPMs.

The following rules are applied to the feature RPM installation procedure during boot or install time:

- Only RPMs with the same NX-OS release number should be selected for the installation.

- Base RPMs cannot be added to the **localdb** repository.

List of NX-OS Mandatory RPMs That Can Be Patched

See the list of the NX-OS mandatory RPMs that can be patched.

Table 11: List of the NX-OS Mandatory RPMs that can be Patched

Serial Number	RPM Name	Description
1	Core	NX-OS infrastructure software.
2	Platform	Cisco NX-OS platform specific software and some Linux modified software.
3	Eth	Cisco NX-OS features that are tightly coupled with the infrastructure.
4	Linecard	Cisco NX-OS arm based line cards in Cisco Nexus 9000 platform switches and x86_64 line cards in Cisco Nexus 9508 platform switches with -R series line cards.
5	Linecard 2	Cisco NX-OS x85_64 line card in Cisco Nexus 9000 Series switches.
6	TOR	Cisco NX-OS Top of Rack switches.

Using Install CLIs for Feature RPM Operation

See the following reference table for using install CLIs for the feature RPM operations:

Table 12: Reference for Install CLIs for the Feature RPM Operations

CLI	Description
install reset	This operation removes all the patches, persisted configurations, upgraded packages, third party installed packages, unsaved configurations, and reloads the switch's previous mode (Full/Base) with the default packages.
	The **install reset** command also performs write erase operation. The following message is displayed at the prompt:
	`switch(config)# install reset`

	`WARNING!!This operation will remove all pactches, upgraded packages,`
	`persisted etc configs, third party packages installed,`
	`startup configuration(write erase) and reload the switch with default packages.`

	`Do you want to proceed with reset operation? (y/n)? [n]`
install reset nxos base	This operation installs NXOS in base mode by removing all patches, upgraded packages, persisted etc configurations, third party packages installed, startup configuration (write erase), and reloads the switch with the default packages.
install reset nxos full	This operation installs NXOS with full mode by removing all patches, upgraded packages, persisted etc configs, third party packages installed, startup configuration (write erase), and reloads the switch with the default packages (with mandatory and optional RPMs).
install add <>	Adds an RPM file to respective repository and updates the repository (**patch/feature/third-party**).
install activate *<rpm name>*	Installs an RPM that is present in the repository.
install commit *<rpm name>*	Used for the patch RPMs. Makes the patch persist during reload.
install deactivate *<rpm name>*	Un-installs an RPM.
install remove *<rpm name>*	Removes an RPM file from the repository and updates the repository.
sh install active	Displays the list of the installed RPMs in the system apart from base rootfs RPMs. (features/patch/third-party).

CLI	Description
sh install inactive	Displays the list of the RPMs that are present in the repository but they are not installed.
sh install packages	Lists all the RPMs that are installed including rootfs RPMs.

Using Install CLIs for Digital Signature Support

See the following section for more information on using the install CLIs for digital signature support.

Procedure

Step 1   switch# **install add bootflash**:*<keyfile>* **gpg-key**

Example:

```
install add bootflash:RPM-GPG-KEY-puppetlabs gpg-key
[####################] 100%
Install operation 304 completed successfully at Thu Apr 19 16:40:28 2018
```

Cisco release RPMs are signed with Cisco GPG (GNU Privacy Guard) key. The public GPG key is present at **/etc/pki/rpm-gpg/arm-Nexus9k-rel.gpg**. To add other public keys from different sources, use the steps in this section.

Step 2   switch#**install verify package** *<package-name>* OR switch#**install verify bootflash**:*<RPM file>*

Example:

```
switch# install verify bootflash:vxlan-2.0.0.0-9.2.1.lib32_n9000.rpm

RSA signed
switch#
```

Displays the CLI to verify whether the RPM file is a signed or non-signed file.

Step 3   **show install packages**

Displays all packages with the signed or unsigned information.

Example:

```
switch# sh install packages

Boot Image:
NXOS Image: bootflash:/nxos.9.2.1.bin

----------------------------------------------------
Installed Packages
attr.x86_64 2.4.47-r0.0 installed Unsigned
aufs-util.x86_64 3.14+git0+b59a2167a1-r0.0 installed Unsigned
base-files.n9000 3.0.14-r89.0 installed Unsigned
base-passwd.lib32_x86 3.5.29-r0.1.0 installed Unsigned
bash.lib32_x86 4.3.30-r0.0 installed Unsigned
bfd.lib32_n9000 2.0.0.0-9.2.1 installed Signed
```

```
bgp.lib32_n9000 2.0.0.0-9.2.1 installed Signed
binutils.x86_64 2.25.1-r0.0 installed Unsigned
bridge-utils.x86_64 1.5-r0.0 installed Unsigned
busybox.x86_64 1.23.2-r0.0 installed Unsigned
busybox-udhcpc.x86_64 1.23.2-r0.0 installed Unsigned
bzip2.x86_64 1.0.6-r5.0 installed Unsigned
ca-certificates.all 20150426-r0.0 installed Unsigned
cgroup-lite.x86_64 1.1-r0.0 installed Unsigned
chkconfig.x86_64 1.3.58-r7.0 installed Unsigned
container-tracker.lib32_n9000 2.0.0.0-9.2.1 installed Signed
containerd-docker.x86_64 0.2.3+gitaa8187dbd3b7ad67d8e5e3a15115d3eef43a7ed1-r0.0
installed Unsigned
core.lib32_n9000 2.0.0.0-9.2.1 installed Signed
coreutils.lib32_x86 8.24-r0.0 installed Unsigned
cpio.x86_64 2.12-r0.0 installed Unsigned
cracklib.lib32_x86 2.9.5-r0.0 installed Unsigned
cracklib.x86_64 2.9.5-r0.0 installed Unsigned
createrepo.x86_64 0.4.11-r9.0 installed Unsigned
cronie.x86_64 1.5.0-r0.0 installed Unsigned
curl.lib32_x86 7.60.0-r0.0 installed Unsigned
db.x86_64 6.0.30-r0.0 installed Unsigned
dbus-1.lib32_x86 1.8.20-r0.0 installed Unsigned
dhcp-client.x86_64 4.3.2-r0.0 installed Unsigned
dhcp-server.x86_64 4.3.2-r0.0 installed Unsigned
switch#
```

Querying All Installed RPMs

Complete the following step to query all the installed RPMs:

Procedure

	Command or Action	**Purpose**
Step 1	**show install packages** **Example:** switch# **show install packages** Boot Image: NXOS Image: bootflash:/nxos.9.2.1.bin --- Installed Packages attr.x86_64 2.4.47-r0.0 installed Unsigned aufs-util.x86_64 3.14+git0+b59a2167a1-r0.0 installed Unsigned base-files.n9000 3.0.14-r89.0 installed Unsigned base-passwd.lib32_x86 3.5.29-r0.1.0 installed Unsigned bash.lib32_x86 4.3.30-r0.0 installed Unsigned bfd.lib32_n9000 2.0.0.0-9.2.1 installed Signed	Queries all the installed RPMs.

Command or Action	Purpose
| ```
bgp.lib32_n9000 2.0.0.0-9.2.1 installed
 Signed
binutils.x86_64 2.25.1-r0.0 installed
Unsigned
bridge-utils.x86_64 1.5-r0.0 installed
Unsigned
busybox.x86_64 1.23.2-r0.0 installed
Unsigned
busybox-udhcpc.x86_64 1.23.2-r0.0
installed Unsigned
bzip2.x86_64 1.0.6-r5.0 installed
Unsigned
ca-certificates.all 20150426-r0.0
installed Unsigned
cgroup-lite.x86_64 1.1-r0.0 installed
Unsigned
chkconfig.x86_64 1.3.58-r7.0 installed
Unsigned
container-tracker.lib32_n9000
2.0.0.0-9.2.1 installed Signed
containerd-docker.x86_64
0.2.3+gitaa8187dbd3b7ad67d8e5e3a15115d3eef43a7ed1-r0.0
installed Unsigned
core.lib32_n9000 2.0.0.0-9.2.1 installed
 Signed
coreutils.lib32_x86 8.24-r0.0 installed
 Unsigned
cpio.x86_64 2.12-r0.0 installed Unsigned
cracklib.lib32_x86 2.9.5-r0.0 installed
 Unsigned
cracklib.x86_64 2.9.5-r0.0 installed
Unsigned
createrepo.x86_64 0.4.11-r9.0 installed
 Unsigned
cronie.x86_64 1.5.0-r0.0 installed
Unsigned
curl.lib32_x86 7.60.0-r0.0 installed
Unsigned
db.x86_64 6.0.30-r0.0 installed Unsigned
dbus-1.lib32_x86 1.8.20-r0.0 installed
Unsigned
dhcp-client.x86_64 4.3.2-r0.0 installed
 Unsigned
dhcp-server.x86_64 4.3.2-r0.0 installed
 Unsigned
switch#
``` | |

Querying Only Installed Featured NX-OS RPMs

Complete the following step to query only the installed featured NX-OS RPMs:

Procedure

	Command or Action	Purpose
Step 1	**show install packages	grep** *<lib32_n9000>* **Example:**	Queries only the installed featured NX-OS RPMs.

Command or Action	Purpose

switch# **show install packages** | **grep** *lib32_n9000*
core.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
eth.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
lacp.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
linecard2.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
lldp.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
mtx-device.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
mtx-grpc-agent.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
mtx-infra.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
mtx-netconf-agent.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
mtx-restconf-agent.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
mtx-telemetry.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
ntp.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
nxos-ssh.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
platform.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
snmp.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
svi.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
tacacs.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
tor.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed
vtp.lib32_n9000
2.0.0.0-9.2.1   installed
       Signed	

Querying Only Installed Third Party RPMs

Complete the following step to query only the installed third party RPMs:

Procedure

	Command or Action	Purpose
Step 1	**show install packages	grep** *<x86_64>*	Queries the installed third-party RPMs.
	Example:	Queries all the installed RPMs.

	switch# **show install packages	grep** *x86_64*
		attr.x86_64
		2.4.47-r0.0 installed
		     Unsigned
		aufs-util.x86_64
		3.14+git0+b59a2167a1-r0.0 installed
		     Unsigned
		base-passwd.x86_64
		3.5.29-r0.1.0 installed
		     Unsigned
		binutils.x86_64
		2.25.1-r0.0 installed
		     Unsigned
		bridge-utils.x86_64
		1.5-r0.0 installed
		     Unsigned
		busybox.x86_64
		1.23.2-r0.0 installed
		     Unsigned
		busybox-udhcpc.x86_64
		1.23.2-r0.0 installed
		     Unsigned
		bzip2.x86_64
		1.0.6-r5.0 installed
		     Unsigned
		cgroup-lite.x86_64
		1.1-r0.0 installed
		     Unsigned	

Installing the RPMs Using One Step Procedure

The CLIs for both install and upgrade RPMs are the same. See the following step to install the RPMs using one step procedure:

Procedure

	Command or Action	Purpose
Step 1	**install add** *<rpm>* **activate**	Installs and activates the RPM.
	Example:	
	switch# **install add** *bootflash:chef.rpm* **activate**	

Command or Action	Purpose
| ```
Adding the patch (/chef.rpm)
[###################] 100%
Install operation 868 completed
successfully at Tue May  8 11:20:10 2018


Activating the patch (/chef.rpm)
[###################] 100%
Install operation 869 completed
successfully at Tue May  8 11:20:20 2018
``` | |

Example

```
switch# show install active
Boot Image:
        NXOS Image: bootflash:/nxos.9.2.1.bin

Active Packages:
bgp-2.0.1.0-9.2.1.lib32_n9000
chef-12.0.0alpha.2+20150319234423.git.1608.b6eb10f-1.el5.x86_64

Active Base Packages:
        lacp-2.0.0.0-9.2.1.lib32_n9000
        lldp-2.0.0.0-9.2.1.lib32_n9000
        mtx-device-2.0.0.0-9.2.1.lib32_n9000
        mtx-grpc-agent-2.0.0.0-9.2.1.lib32_n9000
        mtx-infra-2.0.0.0-9.2.1.lib32_n9000
        mtx-netconf-agent-2.0.0.0-9.2.1.lib32_n9000
        mtx-restconf-agent-2.0.0.0-9.2.1.lib32_n9000
        mtx-telemetry-2.0.0.0-9.2.1.lib32_n9000
        ntp-2.0.0.0-9.2.1.lib32_n9000
        nxos-ssh-2.0.0.0-9.2.1.lib32_n9000
        snmp-2.0.0.0-9.2.1.lib32_n9000
        svi-2.0.0.0-9.2.1.lib32_n9000
        tacacs-2.0.0.0-9.2.1.lib32_n9000
        vtp-2.0.0.0-9.2.1.lib32_n9000
switch(config)#
```

Installing the RPMs Using Two Steps Procedure

The CLIs for both install and upgrade RPMs are the same. See the following steps to install the RPMs using two steps procedure:

Procedure

	Command or Action	Purpose
Step 1	**install add** *<rpm>* **Example:** switch# **install add**	Installs the RPM.

	Command or Action	Purpose
	bootflash:vxlan-2.0.1.0-9.2.1.1ib32_n9000.rpm [####################] 100% Install operation 892 completed successfully at Thu Jun 7 13:56:38 2018 switch(config)# **sh install inactive | grep vxlan** vxlan-2.0.1.0-9.2.1.1ib32_n9000	
Step 2	**install activate** *<rpm>* **Example:**	Activates the RPM.

Example

```
switch#install activate vxlan

[####################] 100%
Install operation 891 completed successfully at Thu Jun  7 13:53:07 2018

switch# show install active | grep vxlan

vxlan-2.0.0.0-9.2.1.1ib32_n9000

switch# sh install inactive | grep vxlan

switch#
```

Upgrading the RPMs Using One Step

The CLIs for both install and upgrade RPMs are the same. See the following steps to upgrade the RPMs:

Procedure

	Command or Action	Purpose
Step 1	**install add** *<rpm>***activate upgrade** **Example:** switch(config)# **install add bootflash:bgp-2.0.2.0-9.2.1.1ib32_n9000.rpm activate upgrade** Adding the patch (/bgp-2.0.2.0-9.2.1.1ib32_n9000.rpm) [####################] 100%	Installs the RPM.

Command or Action	Purpose
| ```
Install operation 870 completed
successfully at Tue May 8 11:22:30 2018

Activating the patch
(/bgp-2.0.2.0-9.2.1.lib32_n9000.rpm)
[####################] 100%
Install operation 871 completed
successfully at Tue May 8 11:22:40 2018
``` | |

Example

```
switch(config)# show install active

Boot Image:
 NXOS Image: bootflash:/nxos.9.2.1.bin

Active Packages:
bgp-2.0.2.0-9.2.1.lib32_n9000
chef-12.0.0alpha.2+20150319234423.git.1608.b6eb10f-1.el5.x86_64

Active Base Packages:
 lacp-2.0.0.0-9.2.1.lib32_n9000
 lldp-2.0.0.0-9.2.1.lib32_n9000
 mtx-device-2.0.0.0-9.2.1.lib32_n9000
 mtx-grpc-agent-2.0.0.0-9.2.1.lib32_n9000
 mtx-infra-2.0.0.0-9.2.1.lib32_n9000
 mtx-netconf-agent-2.0.0.0-9.2.1.lib32_n9000
 mtx-restconf-agent-2.0.0.0-9.2.1.lib32_n9000
 mtx-telemetry-2.0.0.0-9.2.1.lib32_n9000
 ntp-2.0.0.0-9.2.1.lib32_n9000
 nxos-ssh-2.0.0.0-9.2.1.lib32_n9000
 snmp-2.0.0.0-9.2.1.lib32_n9000
 svi-2.0.0.0-9.2.1.lib32_n9000
 tacacs-2.0.0.0-9.2.1.lib32_n9000
 vtp-2.0.0.0-9.2.1.lib32_n9000
```

Downgrading the RPMs

The downgrade procedure needs a special CLI attribute. See the following step to downgrade the RPMs using the one step procedure:

Procedure

	Command or Action	Purpose
| Step 1 | **install add** *<rpm>***activate downgrade** <br><br> **Example:** <br><br> ```
switch(config)# install add
bootflash:bgp-2.0.1.0-9.2.1.lib32_n9000.rpm
 activate downgrade
``` | Downgrades the RPM. |

Command or Action	Purpose
| ```
Adding the patch
(/bgp-2.0.1.0-9.2.1.lib32_n9000.rpm)
[####################] 100%
Install operation 872 completed
successfully at Tue May 8 11:24:43 2018

Activating the patch
(/bgp-2.0.1.0-9.2.1.lib32_n9000.rpm)
[####################] 100%
Install operation 873 completed
successfully at Tue May 8 11:24:52 2018
``` | |

Example

```
switch(config)# show install active
Boot Image:
 NXOS Image: bootflash:/nxos.9.2.1.bin

Active Packages:
 bgp-2.0.1.0-9.2.1.lib32_n9000
 chef-12.0.0alpha.2+20150319234423.git.1608.b6eb10f-1.el5.x86_64

Active Base Packages:
 lacp-2.0.0.0-9.2.1.lib32_n9000
 lldp-2.0.0.0-9.2.1.lib32_n9000
 mtx-device-2.0.0.0-9.2.1.lib32_n9000
 mtx-grpc-agent-2.0.0.0-9.2.1.lib32_n9000
 mtx-infra-2.0.0.0-9.2.1.lib32_n9000
 mtx-netconf-agent-2.0.0.0-9.2.1.lib32_n9000
 mtx-restconf-agent-2.0.0.0-9.2.1.lib32_n9000
 mtx-telemetry-2.0.0.0-9.2.1.lib32_n9000
 ntp-2.0.0.0-9.2.1.lib32_n9000
 nxos-ssh-2.0.0.0-9.2.1.lib32_n9000
 snmp-2.0.0.0-9.2.1.lib32_n9000
 svi-2.0.0.0-9.2.1.lib32_n9000
 tacacs-2.0.0.0-9.2.1.lib32_n9000
 vtp-2.0.0.0-9.2.1.lib32_n9000
switch(config)#
```

Removing the RPMs

See the following steps to remove the RPMs:

Procedure

	Command or Action	Purpose
| **Step 1** | **install remove** *<rpm>*<br><br>**Example:**<br><br>```
switch(config)# show install
inactive | grep vxlan
``` | Removes the RPM from the repository. |

Command or Action	Purpose
```vxlan-2.0.0.0-9.2.1.lib32_n9000 switch(config)# install remove vxlan Proceed with removing vxlan? (y/n)? [n] y [###################] 100% Install operation 890 Removal of base rpm package is not permitted at Thu Jun 7 13:52:15 2018```	

Format of the RPM

The general format of a RPM is <name>-<version>-<release>.<arch>.rpm. The same format is followed for NXOS feature RPMS.

- Name: package name, for example, BGP

- Version in <x.y.x.b> format: <major.minor.patch.build_number>, for example, 2.0.1.0

- Release: The branch from which the RPM is created, for example, 9.2.1

- Arch: The architecture type of the RPM, for example, lib32_n9000

See the following table for more information on the naming convention, for example, fex-2.0.0.0-9.2.1.lib32_n9000.rpm:

Table 13: RPM Naming Convention

RPM Naming Convention Example: fex-2.0.0.0-9.2.1.lib32_n9000.rpm	Description
fex	Indicates the name of the component.
2	Indicates that the RPM is not backward compatible. Configuration loss takes place during an upgrade.
0	Indicates the incremental API changes/CLI changes/Schema changes with backward compatibility. It is applicable to the new features on top of the existing capabilities. No configuration is lost during an upgrade.
0	Indicates a bug fix without any functionality change. No configuration is lost during an upgrade.
0	This number tracks how many times the component has changed during the development cycle of a release. This value will be 0 for all the release images.

RPM Naming Convention Example: fex-2.0.0.0-9.2.1.lib32_n9000.rpm	Description
9.2.1	Indicates the release number or the distribution version for the RPM. It aligns to the NVR format. Since the feature RPM is only applicable to a NXOS release, this field has NXOS release version number present.
lib32_n9000	Indicates the architecture type of the RPM.

Rules for Managing RPM Version During Installation

The **groups-repo** and **localdb** repositories hold the NX-OS feature RPMs that should be installed during the system boot or during activation. The **localdb** repository holds all the persisted RPMs from the old installation. All inactive RPMs that are present in **localdb** that are not required any more, should be removed to make space for the new RPMs.

YUM commands or **install** commands can be used for the installation or the removal of these RPMs.

The following rules are applied to the feature RPM installation procedure during boot or install time:

- The RPM files with the release number that is same as the NXOS release are the compatible files to be activated.

- Users are not allowed to add a RPM to the **localdb** repository if the RPM is present in **groups-repo** repository. Any RPM version other than the **groups-repo** repository should be allowed.

- If a RPM is present in **groups-repo** and it is also present in **localdb** repository with the same version, the RPM from the **groups-repo** repository is considered for the installation during boot time and install time. (This step is needed because in old releases, adding the base RPM allowed the installation to the **localdb** repository. This step is needed for the backward compatibility.)

- When a non-base feature RPM is added, activated, deactivated, or removed, the respective entry should be present in **inactive_feature_rpms.inf**.

- When a base RPM is deactivated or activated, the entry should be present in **/bootflash/.rpmstore/nxos_preinstall_rpms_removed** and **/bootflash/.rpmstore/nxos_rpms_persisted** respectively.

- If a base RPM entry is not present in the **/bootflash/.rpmstore/nxos_rpms_persisted** file, any RPM version present in the **groups-repo** or **localdb** repository should not be considered for the installation during boot time. In this case, the removed entry should be part of **/bootflash/.rpmstore/nxos_preinstall_rpms_removed**.

Note Avoid manually copying the RPMs to the system repositories. Instead, use the **install** or YUM commands.

Information About YUM Commands

See the following sections for more information about YUM commands.

> **Note** YUM commands do not support ctrl+c. Install commands do support ctrl+c. If YUM commands are aborted using ctrl+c, manual cleanup must be performed using "/isan/bin/patching_utils.py --unlock".

Performing Package Operations Using the YUM Commands

See the following sections for performing package operations using the YUM commands:

> **Note** YUM commands are accessed only from the BASH shell on the box and they are not allowed from the NXOS VSH terminal.

> **Note** Make sure that as a sudo user, you have access to the super user privileges.

Finding the Base Version RPM of the Image

Use the **ls /rpms** command to find the base version RPM of the image. The base RPM version is the pre-installed RPM that is archived in the system image.

```
#ls /rpms

bfd-2.0.0.0-9.2.1.lib32_n9000.rpm
ins_tor_sdk_t2-1.0.0.0-9.2.0.77.lib32_n9000.rpm
mtx-netconf-agent-2.0.0.0-9.2.1.lib32_n9000.rpm   snmp-2.0.0.0-9.2.1.lib32_n9000.rpm
bgp-2.0.0.0-9.2.1.lib32_n9000.rpm
ins_tor_sdk_t3-1.0.0.0-9.2.0.77.lib32_n9000.rpm
mtx-restconf-agent-2.0.0.0-9.2.1.lib32_n9000.rpm  sr-2.0.0.0-9.2.1.lib32_n9000.rpm
container-tracker-2.0.0.0-9.2.1.lib32_n9000.rpm  isis-2.0.0.0-9.2.1.lib32_n9000.rpm
        mtx-telemetry-2.0.0.0-9.2.1.lib32_n9000.rpm      svi-2.0.0.0-9.2.1.lib32_n9000.rpm
eigrp-2.0.0.0-9.2.1.lib32_n9000.rpm                lacp-2.0.0.0-9.2.1.lib32_n9000.rpm
        nbproxy-2.0.0.0-9.2.1.lib32_n9000.rpm
tacacs-2.0.0.0-9.2.1.lib32_n9000.rpm
ext-eth-2.0.0.0-9.2.1.lib32_n9000.rpm              lldp-2.0.0.0-9.2.1.lib32_n9000.rpm
        ntp-2.0.0.0-9.2.1.lib32_n9000.rpm
telemetry-2.3.4.0-9.2.1.lib32_n9000.rpm
fcoe-2.0.0.0-9.2.1.lib32_n9000.rpm                 mcast-2.0.0.0-9.2.1.lib32_n9000.rpm
        nxos-ssh-2.0.0.0-9.2.1.lib32_n9000.rpm
virtualization-2.0.0.0-9.2.1.lib32_n9000.rpm
fex-2.0.0.0-9.2.1.lib32_n9000.rpm                  mpls-2.0.0.0-9.2.1.lib32_n9000.rpm
        ospf-2.0.0.0-9.2.1.lib32_n9000.rpm                 vtp-2.0.0.0-9.2.1.lib32_n9000.rpm
fhrp-2.0.0.0-9.2.1.lib32_n9000.rpm                 mtx-device-2.0.0.0-9.2.1.lib32_n9000.rpm
        repodata
vxlan-2.0.0.0-9.2.1.lib32_n9000.rpm
guestshell-2.0.0.0-9.2.1.lib32_n9000.rpm           mtx-grpc-agent-2.0.0.0-9.2.1.lib32_n9000.rpm
        rip-2.0.0.0-9.2.1.lib32_n9000.rpm
icam-2.0.0.0-9.2.1.lib32_n9000.rpm                 mtx-infra-2.0.0.0-9.2.1.lib32_n9000.rpm
        services-2.0.0.0-9.2.1.lib32_n9000.rpm
```

Checking the List of the Installed RPMs

Use the **yum list installed** command to query the feature and third party RPMs and grep a specific RPM. See the following example for feature RPMs:

```
bash-4.2# yum list installed | grep lib32_n9000

bfd.lib32_n9000                      2.0.0.0-9.2.1          @groups-repo
core.lib32_n9000                     2.0.0.0-9.2.1          installed
eth.lib32_n9000                      2.0.0.0-9.2.1          installed
guestshell.lib32_n9000               2.0.0.0-9.2.1          @groups-repo
lacp.lib32_n9000                     2.0.0.0-9.2.1          installed
linecard2.lib32_n9000                2.0.0.0-9.2.1          installed
lldp.lib32_n9000                     2.0.0.0-9.2.1          installed
mcast.lib32_n9000                    2.0.0.0-9.2.1          @groups-repo
mtx-device.lib32_n9000               2.0.0.0-9.2.1          installed
mtx-grpc-agent.lib32_n9000           2.0.0.0-9.2.1          installed
mtx-infra.lib32_n9000                2.0.0.0-9.2.1          installed
mtx-netconf-agent.lib32_n9000        2.0.0.0-9.2.1          installed
mtx-restconf-agent.lib32_n9000       2.0.0.0-9.2.1          installed
mtx-telemetry.lib32_n9000            2.0.0.0-9.2.1          installed
nbproxy.lib32_n9000                  2.0.0.0-9.2.1          installed
ntp.lib32_n9000                      2.0.0.0-9.2.1          installed
nxos-ssh.lib32_n9000                 2.0.0.0-9.2.1          installed
ospf.lib32_n9000                     2.0.0.0-9.2.1          @groups-repo
platform.lib32_n9000                 2.0.0.0-9.2.1          installed
snmp.lib32_n9000                     2.0.0.0-9.2.1          installed
svi.lib32_n9000                      2.0.0.0-9.2.1          installed
tacacs.lib32_n9000                   2.0.0.0-9.2.1          installed
tor.lib32_n9000                      2.0.0.0-9.2.0.77       installed
virtualization.lib32_n9000           2.0.1.0-9.2.1          @localdb
vtp.lib32_n9000                      2.0.0.0-9.2.1          installed
vxlan.lib32_n9000                    2.0.0.0-9.2.1          @groups-repo
...
```

Getting Details of the Installed RPMs

The **yum info** <*rpmname*> command lists out the detailed info of the installed RPM.

```
yum info vxlan

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo

                  | 1.1 kB    00:00 ...
localdb

                  |  951 B    00:00 ...
patching

                  |  951 B    00:00 ...
thirdparty

                  |  951 B    00:00 ...
Installed Packages
Name        : vxlan
Arch        : lib32_n9000
Version     : 2.0.0.0
Release     : 9.2.1
```

```
Size        : 6.4 M
Repo        : installed
From repo   : groups-repo
Summary     : Cisco NXOS VxLAN
URL         : http://cisco.com/
License     : Proprietary
Description : Provides VxLAN support
```

Installing the RPMs

Installing the RPMs downloads the RPMs and copies the respective program to the switches. See the following example for installing the RPMs from a remote server (that is reachable in the network):

```
bash-4.3# yum install
http://10.0.0.2/modularity/rpms/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo
                                                    | 1.1 kB     00:00 ...
localdb
                                                    |  951 B     00:00 ...
localdb/primary
                                                    |  886 B     00:00 ...
localdb
                                                                  1/1
patching
                                                    |  951 B     00:00 ...
thirdparty
                                                    |  951 B     00:00 ...
Setting up Install Process
vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm
                                                    | 1.6 MB     00:00
Examining /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm:
vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================

 Package               Arch                    Version
        Repository                           Size
================================================================================
Installing:
vxlan                 lib32_n9000             2.0.1.0-9.2.1
    /vxlan-2.0.1.0-9.2.1.lib32_n9000         6.4 M
Transaction Summary
================================================================================
Install     1 Package

Total size: 6.4 M
Installed size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
```

```
Running Transaction
  Installing : vxlan-2.0.1.0-9.2.1.lib32_n9000
                                                                       1/1
starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete

Installed:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1


Complete!
```

See the following example for installing the RPMs from local bootflash:

sudo yum install /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm

```
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo

                       | 1.1 kB      00:00 ...
localdb

                       |  951 B      00:00 ...
patching

                       |  951 B      00:00 ...
thirdparty

                       |  951 B      00:00 ...
Setting up Install Process
Examining /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm: vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm as an update to
vxlan-2.0.0.0-9.2.1.lib32_n9000
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================

 Package                              Arch
Version                                               Repository
                                   Size
================================================================================
Updating:
 vxlan                                lib32_n9000
2.0.1.0-9.2.1                                      /vxlan-2.0.1.0-9.2.1.lib32_n9000
                          6.4 M

Transaction Summary
================================================================================
Upgrade       1 Package

Total size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
```

```
Transaction Test Succeeded
Running Transaction
  Updating   : vxlan-2.0.1.0-9.2.1.lib32_n9000

                                      1/2
starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete
  Cleanup    : vxlan-2.0.0.0-9.2.1.lib32_n9000

                                      2/2

Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1



Complete!
```

See the following example for installing the RPM if it is available in a repository:

```
yum install eigrp
```

Upgrading the RPMs

See the following example for upgrading the RPMs from a remote server (that is reachable in the network):

```
bash-4.3# yum upgrade
http://10.0.0.2/modularity/rpms/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo
                                                  | 1.1 kB    00:00 ...
localdb
                                                  |  951 B    00:00 ...
patching
                                                  |  951 B    00:00 ...
thirdparty
                                                  |  951 B    00:00 ...
Setting up Upgrade Process
vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm
                                                  | 1.6 MB    00:00
Examining /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm:
vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm as an update to
vxlan-2.0.0.0-9.2.1.lib32_n9000
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

===============================================================================

 Package               Arch                   Version
       Repository                             Size
===============================================================================
```

```
Updating:
vxlan                    lib32_n9000                    2.0.1.0-9.2.1
    /vxlan-2.0.1.0-9.2.1.lib32_n9000                    6.4 M
Transaction Summary
================================================================================
Upgrade      1 Package

Total size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
** Found 1 pre-existing rpmdb problem(s), 'yum check' output follows:
busybox-1.23.2-r0.0.x86_64 has missing requires of busybox-syslog
  Updating   : vxlan-2.0.1.0-9.2.1.lib32_n9000
                                                                        1/2
starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete
  Cleanup    : vxlan-2.0.0.0-9.2.1.lib32_n9000
                                                                        2/2

Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1


Complete!
```

See the following example for upgrading the RPMs from local bootflash:

sudo yum upgrade /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm

```
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo

                | 1.1 kB     00:00 ...
localdb

                | 951 B     00:00 ...
patching

                | 951 B     00:00 ...
thirdparty

                | 951 B     00:00 ...
Setting up Upgrade Process
Examining /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm: vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm as an update to
vxlan-2.0.0.0-9.2.1.lib32_n9000
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved
```

```
 Package                                        Arch
Version                                                     Repository
                              Size
================================================================================
Updating:
vxlan                                   lib32_n9000
2.0.1.0-9.2.1                                       /vxlan-2.0.1.0-9.2.1.lib32_n9000
                              6.4 M

Transaction Summary
================================================================================
Upgrade       1 Package

Total size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : vxlan-2.0.1.0-9.2.1.lib32_n9000

                                    1/2
starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete
  Cleanup    : vxlan-2.0.0.0-9.2.1.lib32_n9000

                                    2/2

Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1



Complete!
```

See the following example for upgrading the RPMs if it is available in any repository:

yum upgrade eigrp

Downgrading the RPMs

See the following example for downgrading the RPMs from a remote server (that is reachable in the network):

sudo yum downgrade vxlan-2.0.0.0-9.2.1.lib32_n9000

```
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
Setting up Downgrade Process
groups-repo

                    | 1.1 kB     00:00 ...
localdb

                    |  951 B     00:00 ...
localdb/primary
```

```
                      | 1.3 kB     00:00 ...
localdb

                                           2/2
patching

                      |  951 B     00:00 ...
thirdparty

                      |  951 B     00:00 ...
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be a downgrade
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be erased
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================

 Package                                  Arch
         Version                                          Repository
                                 Size
================================================================================
Downgrading:
vxlan                                     lib32_n9000
         2.0.0.0-9.2.1                                    groups-repo
                                 1.6 M
Transaction Summary
================================================================================

Downgrade    1 Package

Total download size: 1.6 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : vxlan-2.0.0.0-9.2.1.lib32_n9000

                                  1/2
starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete
  Cleanup    : vxlan-2.0.1.0-9.2.1.lib32_n9000

                                  2/2

Removed:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1



Installed:
  vxlan.lib32_n9000 0:2.0.0.0-9.2.1



Complete!
```

See the following example for downgrading the RPMs from local bootflash:

yum downgrade /bootflash/eigrp-2.0.0-9.2.1.lib32_n9000.rpm

See the following example for downgrading the RPMs if it is available in any repository:

yum downgrade eigrp

Deleting the RPMs

Deleting the RPMs de-installs the RPMs and removes any configuration CLI of the feature. Use the **yum erase** *<rpm>* command to delete the RPMs.

```
bash-4.2# sudo yum erase vxlan

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
Setting up Remove Process
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be erased
--> Finished Dependency Resolution

Dependencies Resolved

===============================================================================

 Package                 Arch                                     Version
                             Repository                                 Size
===============================================================================
Removing:
 vxlan                   lib32_n9000                              2.0.1.0-9.2.1
                             @/vxlan-2.0.1.0-9.2.1.lib32_n9000           6.4 M
Transaction Summary
===============================================================================
Remove        1 Package

Installed size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Erasing    : vxlan-2.0.1.0-9.2.1.lib32_n9000

                                1/1
starting pre-remove package version mgmt for vxlan
pre-remove for vxlan complete

Removed:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1


Complete!
```

Support for YUM Groups

The support for YUM groups is part of the package management. It simplifies the management of the packages for the administrators and it provides greater flexibility.

The administrators can group a list of packages (RPMs) into a logical group and they can perform various operations. YUM supports the following group commands:

- grouplist

- groupinfo

- groupinstall

- groupremove

- groupupdate

YUM groups can be broadly classified as L2, L3, routing, and management.

Using the grouplist Command

In Linux, number of packages are bundled to particular group. Instead of installing individual packages with yum, you can install particular group that will install all the related packages that belongs to the group. For example to list all the available groups, use the **yum grouplist** command:

```
bash-4.2# sudo yum grouplist

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
Setting up Group Process
groups-repo

                        | 1.1 kB     00:00 ...
localdb

                        |  951 B     00:00 ...
patching

                        |  951 B     00:00 ...
thirdparty

                        |  951 B     00:00 ...
groups-repo/group

                        | 1.6 kB     00:00 ...
Installed Groups:
   L2
   L3
   management
Available Groups:
   routing
Done

bash-4.3$
```

Using the groupmembers Command

Use **yum groupinfo** command to display the description and the contents of a package group. The command lists out the feature members of the group.

```
bash-4.2# sudo yum groupinfo l2

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
Setting up Group Process
groups-repo

                       | 1.1 kB     00:00 ...
localdb

                       |  951 B     00:00 ...
patching

                       |  951 B     00:00 ...
thirdparty

                       |  951 B     00:00 ...

Group: L2
 Mandatory Packages:
   lacp
   lldp
   svi
   vtp
```

Using the groupinstall Command

This command is for both install & upgrade of the members RPM. If the member is not installed, it will install the highest version available. If the member is already installed and higher RPM is available, it will upgrade that member.

```
bash-4.2# sudo yum groupinstall routing

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo

                       | 1.1 kB     00:00 ...
localdb

                       |  951 B     00:00 ...
patching

                       |  951 B     00:00 ...
thirdparty

                       |  951 B     00:00 ...
Setting up Group Process
Package ospf-2.0.0.0-9.2.1.lib32_n9000 already installed and latest version
Resolving Dependencies
--> Running transaction check
---> Package bgp.lib32_n9000 0:2.0.0.0-9.2.1 will be installed
```

```
---> Package eigrp.lib32_n9000 0:2.0.0.0-9.2.1 will be installed
---> Package isis.lib32_n9000 0:2.0.0.0-9.2.1 will be installed
---> Package rip.lib32_n9000 0:2.0.0.0-9.2.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================

 Package                 Arch                                     Version
                                          Repository              Size
================================================================================
Installing:
 bgp                     lib32_n9000                              2.0.0.0-9.2.1
                                          groups-repo             2.4 M
 eigrp                   lib32_n9000                              2.0.0.0-9.2.1
                                          groups-repo             428 k
 isis                    lib32_n9000                              2.0.0.0-9.2.1
                                          groups-repo             1.2 M
 rip                     lib32_n9000                              2.0.0.0-9.2.1
                                          groups-repo             214 k

Transaction Summary
================================================================================
Install      4 Packages

Total download size: 4.2 M
Installed size: 19 M
Is this ok [y/N]: y
Downloading Packages:
--------------------------------------------------------------------------------
Total

          132 MB/s | 4.2 MB     00:00
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : rip-2.0.0.0-9.2.1.lib32_n9000

                                  1/4
starting pre-install package version mgmt for rip
pre-install for rip complete
starting post-install package version mgmt for rip
post-install for rip complete
  Installing : isis-2.0.0.0-9.2.1.lib32_n9000

                                  2/4
starting pre-install package version mgmt for isis
pre-install for isis complete
starting post-install package version mgmt for isis
post-install for isis complete
  Installing : eigrp-2.0.0.0-9.2.1.lib32_n9000

                                  3/4
starting pre-install package version mgmt for eigrp
pre-install for eigrp complete
starting post-install package version mgmt for eigrp
post-install for eigrp complete
  Installing : bgp-2.0.0.0-9.2.1.lib32_n9000

                                  4/4
starting pre-install package version mgmt for bgp
pre-install for bgp complete
starting post-install package version mgmt for bgp
```

```
post-install for bgp complete

Installed:
  bgp.lib32_n9000 0:2.0.0.0-9.2.1                    eigrp.lib32_n9000 0:2.0.0.0-9.2.1
              isis.lib32_n9000 0:2.0.0.0-9.2.1                    rip.lib32_n9000
0:2.0.0.0-9.2.1

Complete!
```

Using the groupupdate Command

Use the **yum groupupdate** command to update any existing installed group packages.

```
bash-4.3# yum groupupdate routing

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo

                        | 1.1 kB      00:00 ...
localdb

                        |  951 B      00:00 ...
localdb/primary

                        | 1.9 kB      00:00 ...
localdb

                                        6/6
patching

                        |  951 B      00:00 ...
thirdparty

                        |  951 B      00:00 ...
Setting up Group Process
Resolving Dependencies
--> Running transaction check
---> Package bgp.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package bgp.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
---> Package eigrp.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package eigrp.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
---> Package isis.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package isis.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
---> Package ospf.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package ospf.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
---> Package rip.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package rip.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================

 Package              Arch                                          Version
                                   Repository        Size
================================================================================

Updating:
 bgp                  lib32_n9000                               2.0.1.0-9.2.1
                                   localdb           2.4 M
```

```
eigrp                      lib32_n9000                                                           2.0.1.0-9.2.1
                                                  locald             428 k
isis                       lib32_n9000                                                           2.0.1.0-9.2.1
                                                  local              1.2 M
ospf                       lib32_n9000                                                           2.0.1.0-9.2.1
                                                  localdb            2.8 M
rip                        lib32_n9000                                                           2.0.1.0-9.2.1
                                                  localdb            214 k
Transaction Summary
================================================================================
Upgrade       5 Packages

Total download size: 7.0 M
Is this ok [y/N]: y
Downloading Packages:
--------------------------------------------------------------------------------
Total

              269 MB/s | 7.0 MB      00:00
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : eigrp-2.0.1.0-9.2.1.lib32_n9000


                                  1/10
starting pre-install package version mgmt for eigrp
pre-install for eigrp complete
starting post-install package version mgmt for eigrp
post-install for eigrp complete
  Updating   : ospf-2.0.1.0-9.2.1.lib32_n9000


                                  2/10
starting pre-install package version mgmt for ospf
pre-install for ospf complete
starting post-install package version mgmt for ospf
post-install for ospf complete
  Updating   : rip-2.0.1.0-9.2.1.lib32_n9000


                                  3/10
starting pre-install package version mgmt for rip
pre-install for rip complete
starting post-install package version mgmt for rip
post-install for rip complete
  Updating   : isis-2.0.1.0-9.2.1.lib32_n9000


                                  4/10
starting pre-install package version mgmt for isis
pre-install for isis complete
starting post-install package version mgmt for isis
post-install for isis complete
  Updating   : bgp-2.0.1.0-9.2.1.lib32_n9000


                                  5/10
starting pre-install package version mgmt for bgp
pre-install for bgp complete
starting post-install package version mgmt for bgp
post-install for bgp complete
  Cleanup    : bgp-2.0.0.0-9.2.1.lib32_n9000


                                  6/10
  Cleanup    : isis-2.0.0.0-9.2.1.lib32_n9000


                                  7/10
```

```
    Cleanup    : rip-2.0.0.0-9.2.1.lib32_n9000

                                   8/10
    Cleanup    : ospf-2.0.0.0-9.2.1.lib32_n9000

                                   9/10
    Cleanup    : eigrp-2.0.0.0-9.2.1.lib32_n9000

                                   10/10

Updated:
  bgp.lib32_n9000 0:2.0.1.0-9.2.1        eigrp.lib32_n9000 0:2.0.1.0-9.2.1
isis.lib32_n9000 0:2.0.1.0-9.2.1      ospf.lib32_n9000 0:2.0.1.0-9.2.1      rip.lib32_n9000
  0:2.0.1.0-9.2.1

Complete!
```

Using the grouperase Command

Use the **yum grouperase** command to delete the groups or all the RPM members of the group.

```
bash-4.3$ sudo yum grouperase routing

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
Setting up Group Process
groups-repo

                        | 1.1 kB     00:00 ...
localdb

                        |  951 B     00:00 ...
patching

                        |  951 B     00:00 ...
thirdparty

                        |  951 B     00:00 ...
Resolving Dependencies
--> Running transaction check
---> Package bgp.lib32_n9000 0:2.0.0.0-9.2.1 will be erased
---> Package eigrp.lib32_n9000 0:2.0.0.0-9.2.1 will be erased
---> Package isis.lib32_n9000 0:2.0.0.0-9.2.1 will be erased
---> Package ospf.lib32_n9000 0:2.0.0.0-9.2.1 will be erased
---> Package rip.lib32_n9000 0:2.0.0.0-9.2.1 will be erased
--> Finished Dependency Resolution

Dependencies Resolved

=================================================================================


 Package              Arch                                            Version
                                      Repository          Size
=================================================================================

Removing:
 bgp                  lib32_n9000                                     2.0.0.0-9.2.1
                                      @groups-repo        11 M
 eigrp                lib32_n9000                                     2.0.0.0-9.2.1
                                      @groups-repo        2.0 M
 isis                 lib32_n9000                                     2.0.0.0-9.2.1
                                      @groups-repo        5.7 M
 ospf                 lib32_n9000                                     2.0.0.0-9.2.1
                                      @groups-repo        15 M
```

```
 rip                lib32_n9000                                    2.0.0.0-9.2.1
                                          @groups-repo          1.0 M

Transaction Summary
========================================================================================

Remove        5 Packages

Installed size: 34 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Erasing    : isis-2.0.0.0-9.2.1.lib32_n9000

                                  1/5
starting pre-remove package version mgmt for isis
pre-remove for isis complete
  Erasing    : ospf-2.0.0.0-9.2.1.lib32_n9000

                                  2/5
starting post-remove package version mgmt for isis
post-remove for isis complete
starting pre-remove package version mgmt for ospf
pre-remove for ospf complete
  Erasing    : eigrp-2.0.0.0-9.2.1.lib32_n9000

                                  3/5
starting post-remove package version mgmt for ospf
post-remove for ospf complete
starting pre-remove package version mgmt for eigrp
pre-remove for eigrp complete
  Erasing    : rip-2.0.0.0-9.2.1.lib32_n9000

                                  4/5
starting post-remove package version mgmt for eigrp
post-remove for eigrp complete
starting pre-remove package version mgmt for rip
pre-remove for rip complete
  Erasing    : bgp-2.0.0.0-9.2.1.lib32_n9000

                                  5/5
starting post-remove package version mgmt for rip
post-remove for rip complete
starting pre-remove package version mgmt for bgp
pre-remove for bgp complete

Removed:
  bgp.lib32_n9000 0:2.0.0.0-9.2.1      eigrp.lib32_n9000 0:2.0.0.0-9.2.1
isis.lib32_n9000 0:2.0.0.0-9.2.1     ospf.lib32_n9000 0:2.0.0.0-9.2.1      rip.lib32_n9000
 0:2.0.0.0-9.2.1

Complete!
```

Finding Repositories

This command lists the repositories that the switch has along with the number of RPMs it has to those repositories.

```
bash-4.3# yum repolist all
```

```
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo

                      | 1.1 kB     00:00 ...
localdb

                      | 951 B      00:00 ...
patching

                      | 951 B      00:00 ...
thirdparty

                      | 951 B      00:00 ...
repo id
       repo name
                                   status
groups-repo
       Groups-RPM Database
                                   enabled: 37
localdb
       Local RPM Database
                                   enabled:  6
patching
       Patch-RPM Database
                                   enabled:  0
thirdparty
       Thirdparty RPM Database
                                   enabled:  0
open-nxos
       open-nxos
                                   disabled
repolist: 43
```

Finding the Installed YUM Version

See the following example for listing the installed YUM version:

yum --version

```
3.4.3
  Installed: rpm-5.4.14-r0.0.x86_64 at 2018-06-02 13:04
  Built    : Wind River <info@windriver.com> at 2018-04-27 08:36
  Committed: Wind River <info@windriver.com> at 2018-04-27

  Installed: yum-3.4.3-r9.0.x86_64 at 2018-06-02 13:05
  Built    : Wind River <info@windriver.com> at 2018-04-27 08:36
  Committed: Wind River <info@windriver.com> at 2018-04-27
```

Mapping the NX-OS CLI to the YUM Commands

See the following table for mapping the NX-OS CLI to the YUM commands:

Table 14: Patching Command Reference

NX-OS CLI Commands	YUM Commands
show install inactive	**yum list --patch-only available**

NX-OS CLI Commands	YUM Commands
show install active	**yum list --patch-only installed**
show install committed	**yum list --patch-only committed**
show install packages	**yum list --patch-only**
show install pkg-info	**yum info --patch-only**
show install log	**yum history --show-patch-log** where log_cmd: • opid= - Log that is specific to an operation ID. • last - Shows the latest operation log. • reverse – Shows the log in reverse order. • detail – Show detailed log. • from= - Shows logging from a specific operation ID.
clear install log	**yum history --clear-patch-log=** where clear_log_cmd: • all - Clears the complete log. • - Clears the logs above this operation ID.
install add	**yum install --add bootflash:/**
install remove	**yum install --remove**
install remove inactive	**yum install --remove all**
install activate	**yum install --no-persist --nocommit** **Note** By default, all packages are activated and committed.
install deactivate	**yum erase --nocommit** **Note** By default, all packages are de-activated and committed.
install commit	**yum install --commit**
Install commit	**yum install --commit all**

Creating User Roles for Install Operation

The **install** command is only available to the users of admin role. The **install** command can be available to a user by RBAC. See RBAC configuration guidelines for the same.