



Workflow for Installing Cisco IOS XR7 Software

The router is shipped with a pre-installed version of the Cisco IOS XR (XR7) software. When the router is powered ON for the first time, the pre-installed software starts functioning automatically. You configure the router for network capabilities. When a new version of the software is available, you can upgrade the system using these tasks:



Note For instructions to upgrade image-specific software, navigate to the [CCO Software Download](#) portal, select the product and refer to the `ncs5700-x64-<version>.docs.tar` file for the release.

- [Obtain Data Models for Install Operation, on page 1](#)
- [Create Repository to Access Files for Installing IOS XR Software, on page 3](#)
- [Upgrade the System, on page 6](#)
- [Install Optional Packages to Provide Additional Functionality, on page 16](#)

Obtain Data Models for Install Operation

You can use YANG data models to install and upgrade XR7 software. The data models are packaged with the release image in the `/pkg/yang` directory.

The models are in the `.yang` format. Each data model can be identified as one of the following functionalities:

- `-oper` in the model name indicates an operational model. For example, `Cisco-IOS-XR-install-oper.yang` and `Cisco-IOS-XR-install-augmented-oper.yang` are operational models for the install operation.
- `-cfg` indicates a configuration model. For example, `Cisco-IOS-XR-install-cfg.yang` is a configuration model for the install operation.
- `-act` indicates an action model. For example, `Cisco-IOS-XR-install-augmented-act.yang` and `Cisco-IOS-XR-install-act.yang` are action models for the install operation.

Procedure

- Step 1** View the install-related data models on the router. You can also access the supported data models to install Cisco IOS XR software from the [Github](#) repository.

Example:

```
node0_RP0_CPU0:/pkg/yang]$ls -ltr *install*
-rw-r--r--. 1 root root 8646 Jul 2 01:59 Cisco-IOS-XR-install-act.yang
-rw-r--r--. 1 root root 7267 Jul 2 01:59 Cisco-IOS-XR-install-search-act.yang
-rw-r--r--. 1 root root 10664 Jul 2 01:59 Cisco-IOS-XR-install-augmented-act.yang
-rw-r--r--. 1 root root 2511 Jul 2 02:00 Cisco-IOS-XR-um-install-cfg.yang
-rw-r--r--. 1 root root 2270 Jul 2 02:04 Cisco-IOS-XR-install-cfg.yang
-rw-r--r--. 1 root root 6222 Jul 2 02:04 Cisco-IOS-XR-install-oper.yang
-rw-r--r--. 1 root root 14009 Jul 2 02:04 Cisco-IOS-XR-install-augmented-oper.yang
-rw-r--r--. 1 root root 13726 Jul 2 02:04 Cisco-IOS-XR-install-oper-sub1.yang
-rw-r--r--. 1 root root 2440 Jul 2 02:04 Cisco-IOS-XR-install-augmented-oper-sub2.yang
-rw-r--r--. 1 root root 59866 Jul 2 02:04 Cisco-IOS-XR-install-augmented-oper-sub1.yang
```

The following table describes the function of the install-related data models:

| Data Model | Description |
|-------------------------------------|--|
| Cisco-IOS-XR-um-install-cfg model | Unified data model that contains a collection of YANG definitions for Cisco IOS XR install package configuration, and augments the modules with configuration data |
| Cisco-IOS-XR-install-oper | Operational data model to view details that are related to basic package information, active and committed packages, and fixes |
| Cisco-IOS-XR-install-cfg | Configuration data model to specify the location of the install source |
| Cisco-IOS-XR-install-act | Action model to perform basic install operations and software upgrade |
| Cisco-IOS-XR-install-search-act | Action model that contains a collection of YANG definitions for install actions related to searching for package information |
| Cisco-IOS-XR-install-augmented-oper | Augmented operational model that displays information about packaging, atomic changes, and history of the install operation on the router |
| Cisco-IOS-XR-install-augmented-act | Action model to perform flexible install operations, including controlling the exact timing of system reloads and rolling back to a previous commit |
| Cisco-IOS-XR-shellutil-copy-act | Action model to copy files on the router from a source location |

Step 2

Use a manageability agent like NETCONF or gRPC to connect and communicate with the router. You can send Remote Procedure Calls (RPC) requests to configure or retrieve operational data from the router. The router processes the request and responds to the request through an RPC response. You use the RPCs to send requests to install the software by populating the relevant container and leaf in the data model.

For more information about understanding the data model structure and using data models, see the *Programmability Configuration Guide for Cisco NCS 5500 Series Routers*.

Create Repository to Access Files for Installing IOS XR Software



Note If only Golden ISO (GISO) is used, you do not need to create a repository.

To install packages (RPM), code upgrades, and updates in XR7, you need a repository of RPMs for the router to download the RPMs for installation. The repository can be local to the router, or accessed remotely through FTP, HTTP, or HTTPS.



Important The repository must be created specific to each platform and release. Do not create repositories with a mix of platforms and releases.

When the repository is accessed remotely, you must provide a repository URL from where the install files are fetched. The URL contains:

- IP address of the server
- Port number of the server
- (Optional) Virtual Routing and Forwarding (VRF) name

The repository can be configured to be reachable using a non-default VRF table. If the repository is reachable through an address in a VRF, specify the name of the VRF.

The format of the repository URL is one of the following:

- FTP: `ftp://<server>[;<vrf>]/<path-to-repository>`
- HTTP: `http://<server>[;<vrf>]/<path-to-repository>`
- HTTPS: `https://<server>[;<vrf>]/<path-to-repository>`
- Local: `file:///<path-to-repository>`. The path to the repository must be under `/harddisk:/` location.

For example, the URL for HTTP server is `http://172.16.0.0:3333/`.



Note Username and password are not supported for HTTP and FTP repositories.

Create and Configure a Local Repository

The router can serve as repository to host the RPMs. You must be a `root-lr` user with access to the router shell. Remote repository is the recommended method to access the RPMs. However, if remote repository is not your preferred option, then you can use the router as a repository to host the RPMs.

Using a local repository removes the need to setup an external server for software installation. In this method, the image files are copied directly to the router, and used to create a repository locally. However, on the downside, the files for future updates must be copied to each router individually.

This section provides the procedure for setting up a local RPM repository on the router.

Procedure

- Step 1** Create a directory locally on the router's `/harddisk:`. Copy the required RPMs and ISO files (using `copy` or `scp` command) from the server to the local directory on the router.
- Step 2** Access the shell of the router using `run` command and untar the RPMs.

Example:

```
Router#run
[node:~]$cd <directory-with-rpms>
[node:~]$tar -xvzf <rpm-name>.tgz
```

- Step 3** Exit from the shell.
- Step 4** Configure the local repository.

Example:

```
Router#config
Router(config)#install repository local-repo url file:///harddisk:/<directory-with-rpms>
Router(config)#commit
<data and time stamp> UTC: config[67543]: Configuration committed by user.
Router(config)#end
```

where, `local-repo` is the repository name, `file:///harddisk:/<directory-with-rpms>` is the local repository URL.

- Step 5** Check the contents of the repository.

Example:

NCS 5700 Series:

```
Router#show install available
Trying to access repositories...
Package                Architecture      Version           Repository
-----
xr-ncs5700-core        x86_64           7.3.1v1.0.1-1    local-repo
xr-core                x86_64           7.3.1v1.0.1-1    local-repo
```

Note

Only the top-level packages are displayed. The contents of the repository is displayed only when the configured repository is valid and the RPMs are present in the repository. It displays only the packages that are available in the repository and not part of active system.

Create and Configure an External Repository

To create an external repository, use a server that can be reached over HTTP, HTTPS or FTP. The following instructions are applicable to Linux distribution systems.

Using an external repository provides a central common repository to be used across devices. This eliminates the need to copy files for future updates to each router individually. It also serves as a single source when new RPMs (bug fixes, packages, updates) are made available. This is the recommended method to setup a repository.

Before you begin

Ensure that you have completed the following tasks:

- Set up your HTTP, HTTPS or FTP server. Ensure that the server is reachable as specified in the note above.
- Install `createrepo` utility on the Linux distribution system (if not installed already).

Procedure

Step 1 Create a directory on the server and copy all the RPMs to a directory. This directory hosts the repository and must be accessible to the HTTP, HTTPS or FTP server that the router will use to access the repository. For example, `/var/www/html`, is the directory where the repository will be created.

If the RPM files are archived (.tar format) or compressed (.tgz or .gz format), extract the files. The files hierarchically arrange in sub directories under the main directory that is used as a repository.

Step 2 Convert the directory to a repository using `createrepo` utility on the Linux server. This creates a directory named `repodata` with the metadata of all the RPMs.

Example:

```
[node]$createrepo --database /var/www/html/  
Saving Primary metadata  
Saving file lists metadata  
Saving other metadata  
Generating sqlite DBs  
Sqlite DBs complete  
  
[node]$cd /var/www/html/  
[node]$ls  
repodata
```

If you add new packages to the repository, change or remove packages from the repository, you must run `createrepo` command again to update the metadata. This ensures that the package manager chooses the correct packages.

Step 3 Configure the external repository.

Example:

```
Router#config  
Router(config)#install repository remote-repo url http://10.194.88.104/<directory-with-rpms>  
Router(config)#commit  
<data and time stamp> UTC: config[67542]: Configuration committed by user 'cisco'.  
Router(config)#end
```

For FTP, the repository is configured as follows:

```
Router#config  
Router(config)#install repository remote-repo url ftp://10.194.88.104/<directory-with-rpms>  
Router(config)#commit  
<data and time stamp> UTC: config[67543]: Configuration committed by user 'cisco'.  
Router(config)#end
```

where, `remote-repo` is the repository name, `http://10.194.88.104/<directory-with-rpms>` is the HTTP repository URL, and `ftp://10.194.88.104/<directory-with-rpms>` is the FTP repository URL.

Step 4 Verify connectivity to the server, and check the contents of the repository.

Example:

NCS 5700 Series:

Router#show install available

Trying to access repositories...

| Package | Architecture | Version | Repository |
|-----------------|--------------|----------------|-------------|
| ----- | ----- | ----- | ----- |
| xr-ncs5700-core | x86_64 | 7.3.1v1.0.1-1 | local-repo |
| xr-core | x86_64 | 7.3.11v1.0.1-1 | remote-repo |

Note

Only the top-level packages are displayed. The contents of the repository are displayed only when the configured repository is valid and the RPMs with the updated metadata are present in the repository. It displays only the packages that are available in the repository and not part of active system.

Upgrade the System

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

You can replace the currently active software on your system with the software from a specified ISO image or GISO image. Only a minimal set of changes is performed to upgrade to the new software. Packages are not removed and reinstalled if they have the same name and version. For example, an upgrade that differs by only one package, removes and install only that one modified package.



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

This section shows replacing the current software version with .iso image. To understand the phases of install operation, see [Software Deliverables and Terminologies](#). For information about repositories, see [Create Repository to Access Files for Installing IOS XR Software, on page 3](#).

View Supported Software Upgrade or Downgrade Versions

Table 1: Feature History Table

| Feature Name | Release Information | Description |
|---|---------------------|--|
| Supported Software Upgrade or Downgrade IOS XR Versions | Release 7.5.2 | <p>You can determine whether a software version can be upgraded or downgraded to another version using this functionality. Before an actual upgrade or downgrade process, you can also view the hardware or software limitations that could cause the upgrade or downgrade to fail. This feature helps you plan successful software upgrades or downgrades.</p> <p>This feature introduces the show install upgrade-matrix command.</p> |

Your Cisco router comes preinstalled with IOS XR software. You either upgrade the software release to use new features and software fixes, or you downgrade the software. To leverage new features that are added or software fixes that are provided, it is important that you upgrade your router to a current version.

To help you select a Cisco IOS XR software release that aligns with Cisco-certified upgrade and downgrade paths, this feature provides answers to the following questions:

- What upgrade or downgrade releases are supported for the current release?
- I plan to upgrade from Release X to Release Y. Does my router support upgrade to Release Y?
- Are there any bridging SMUs that must be installed before I upgrade the software?

This feature provides a mechanism to determine whether the current release supports an upgrade to a target release. This task is run at the start of a software upgrade or downgrade through the **install replace** command. If the validation fails, the software upgrade is blocked, and the system notifies the reason for the failure. This feature allows you to proactively examine whether you can upgrade or downgrade to a certain release, saving time and effort involved in planning and upgrading the software.

The feature provides the following information to help you understand the prerequisites or limitations related to the specific software upgrade or downgrade:

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

You can overwrite the automatic validation using the **force** keyword in the **install replace** command. With this option, the system displays warning messages when the upgrade fails but does not block the software upgrade. Use the **force ?** keyword to understand any other impact to system functionalities apart from the disabling of this process that determines the supported releases for software upgrade or downgrade.

You can view the support information using the following **show** commands or through the operational data.

| Command | Description |
|---|--|
| show install upgrade-matrix running | Displays all supported software upgrades from the current version according to the support data installed on the running system |
| show install upgrade-matrix running v1 v2 | Displays details about the software upgrades from version 1 to version 2 according to the support data installed on the running system |
| show install upgrade-matrix running all | Displays all supported software upgrades from any version according to the support data installed on the running system |
| show install upgrade-matrix iso <i>path-to-ISO</i> | Displays details about the software upgrade from the current version to the version of the target ISO according to the support data in both the running system and the ISO image |
| show install upgrade-matrix iso <i>path-to-ISO</i> v1 v2 | Displays details about the software upgrade from version 1 to version 2 according to the support data in the target ISO image |
| show install upgrade-matrix iso <i>path-to-ISO</i> all | Displays all supported software upgrades from any version according to the support data in the target ISO image |
| show install upgrade-matrix iso <i>path-to-ISO</i> running | Displays details about the software upgrade from the current version to the version of ISO according to the support matrices in both the running system and the target ISO image |
| show install upgrade-matrix rollback <i>ID</i> | Displays details about the software upgrade from the current version to a version of a specific rollback point (indicated by an ID) according to the support matrices in both the running system and the rollback ID |
| show install upgrade-matrix rollback <i>ID</i> v1 v2 | Displays details about the software upgrade from version 1 to version 2 according to the support data in the specific rollback ID |
| show install upgrade-matrix rollback <i>ID</i> all | Displays all supported software upgrades from any version according to the support data in the specific rollback ID |
| show install upgrade-matrix rollback <i>ID</i> running | Displays details about the software upgrade from the current version to the version of the specific rollback ID according to the support matrices in both the running system and the rollback ID |

View All Supported Software Upgrade from Running Version

The following example shows all supported releases for upgrade from the current version 7.5.2 on the router:

```
Router#show install upgrade-matrix running
Tue Jun  7 06:36:06.865 UTC
This may take a while ...
```

The current software [7.5.2] can be upgraded from and downgraded to the following releases:

```
=====
From      To      Bridge SMUs Required      Caveats
=====
7.5.2     7.6.1     None                      - Caveat string
-----
```

View the Release Support for Software Upgrade Between Two Releases

The following example shows the support information for an upgrade from one release to another—7.5.2 and 7.3.1:

```
Router#show install upgrade-matrix running 7.5.2 7.3.1
Tue May 10 19:33:59.135 UTC
```

```
-----
Upgrade matrix information for system upgrade: 7.5.2->7.3.1
-----
```

XR system upgrade is supported, with the following restrictions:

The following fixes must be installed if any version of the package is installed.

```
-----
Ddts      Name      Version
-----
CSCab54345  xr-bgp    7.5.2
```

View All Releases that Support Software Upgrade from Current Software

The following example shows all releases that support upgrades or downgrades for the current release:

```
Router#show install upgrade-matrix running all
Tue May 10 19:33:59.135 UTC
```

```
-----
Matrix: XR Version: 7.5.2, File version: 1.0
```

The upgrade matrix indicates that the following system upgrades are supported:

```
-----
From      To      Restrictions
-----
7.3.0     7.5.2     -
7.3.1     7.5.2     -
7.3.2     7.5.2     -
7.3.3     7.5.2     -
7.3.4     7.5.2     -
7.5.1     7.5.2     -
7.5.2     7.3.0     -
7.5.2     7.3.1     Bridging fixes
7.5.2     7.3.2     -
7.5.2     7.3.3     -
```

| | | |
|-------|-------|---|
| 7.5.2 | 7.3.4 | - |
| 7.5.2 | 7.5.1 | - |
| 7.5.2 | 7.6.1 | - |
| 7.6.1 | 7.5.2 | - |

View Supported Releases to Upgrade Software From Current Version to Target Version

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

```
Router#show install upgrade-matrix iso /harddisk\:\ncs5500-mini-x-7.5.2.iso
```

```
-----
Upgrade matrix information for system upgrade: 7.5.2->7.6.1
-----
```

```
XR system upgrade is supported with no additional restrictions
```

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

View Supported Releases from Current Version to an ISO Version

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

```
Router#show install upgrade-matrix iso /harddisk\:\ncs5500-x64-7.5.2.iso all
Tue Jun  7 06:40:43.973 UTC
This may take a while ...
```

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

| From | To | Bridge SMUs Required | Caveats |
|-------|-------|----------------------|---------|
| 7.6.1 | 7.5.2 | None | None |
| 7.4.2 | 7.5.2 | None | None |
| 7.4.1 | 7.5.2 | None | None |
| 7.3.2 | 7.5.2 | None | None |
| 7.3.1 | 7.5.2 | None | None |
| 7.2.2 | 7.5.2 | None | None |
| 7.2.1 | 7.5.2 | None | None |
| 7.5.2 | 7.6.1 | None | None |
| 7.5.2 | 7.4.2 | None | None |
| 7.5.2 | 7.4.1 | None | None |
| 7.5.2 | 7.3.2 | None | None |
| 7.5.2 | 7.3.1 | None | None |

| | | | |
|-------|-------|------|------|
| 7.5.2 | 7.2.2 | None | None |
| 7.5.2 | 7.2.1 | None | None |

Replacing a Software Release with an Unsupported Release

In this example, consider you attempt to replace version 7.4.2 with an unsupported version 7.5.1. This operation fails and a message is displayed to notify the reason for the failure.

```
Router#install package replace /harddisk\:/ncs5500-x64-7.4.2.iso synchronous
Tue May 10 19:33:59.135 UTC
Starting:
  install package replace /harddisk:/ncs5500-x64-7.4.2.iso
Packaging operation 1.1.1
Press Ctrl-C to return to the exec prompt. This will not cancel the install operation

Current activity: Initializing ...
Current activity: Verify input and download ISO .....

Packaging operation 1.1.1: 'install package replace /harddisk:/ncs5500-x64-7.4.2.iso' failed
-
Software upgrade 7.5.1->7.4.2 not listed as permitted.
Automatically recovered after failure, ready for next user request.
```

View All Releases that Support Rollback from Current Software

The following example shows all releases that support upgrades or downgrades for the current release:

```
Router#show install upgrade-matrix rollback 130 all
Mon Jun  6 10:42:23.013 UTC
Matrix: XR version: 7.5.2, File version: 1.0
```

The upgrade matrix indicates that the following system upgrades are supported:

| From | To | Restrictions |
|-------|-------|----------------|
| 7.3.1 | 7.5.2 | Bridging fixes |
| 7.3.2 | 7.5.2 | - |
| 7.4.1 | 7.5.2 | - |
| 7.4.2 | 7.5.2 | - |
| 7.5.1 | 7.5.2 | - |
| 7.6.1 | 7.5.2 | - |
| 7.5.2 | 7.3.1 | - |
| 7.5.2 | 7.3.2 | - |
| 7.5.2 | 7.4.1 | - |
| 7.5.2 | 7.5.1 | - |

View the Release Support for Software Upgrade Between Two Releases with Specific Rollback ID

The following example shows the support information for an upgrade from one release to another—7.5.2 and 7.3.1, with specific rollback ID 130:

```
Router#show install upgrade-matrix rollback 130 7.3.1 7.5.2
Mon Jun  6 10:42:55.052 UTC
-----
Upgrade matrix information for system upgrade: 7.3.1->7.5.2
-----
```

XR system upgrade is supported, with the following restrictions:

The following fixes must be installed if any version of the package is installed.

| Ddts | Name | Version |
|------------|------------|-------------|
| CSCvy66646 | xr-install | 7.3.1v1.0.1 |

Upgrade the Current Active Version of Cisco IOS XR Software

This section shows replacing the current software version with `.iso` image. The instructions in this section also apply to system downgrade.

Upgrade the Current Active Version

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

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

From Cisco IOS XR Release 7.5.1, you can use the Automatic Install of Bridging Bug Fix RPMs feature to install any bridging bug fix RPMs that are prerequisite for a system upgrade or a downgrade.

You need to add the required Bridging Bug Fix RPMs into the customized ISO built using Cisco Golden ISO (GISO) build script **gisobuild.py**, by going through the steps in [Build a Golden ISO](#). Ensure that you add the required bridging bug fix RPMs into your Golden ISO. The GISO can include bridging Bug Fix RPMs for multiple releases, and installs only the specific bridging Bug Fix RPMs required for the target release. The bridging bug fix RPMs can be used in the following scenarios:

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

The **install replace** command adds and applies the highest version of any bridging bugfix RPMs for the currently active XR version.

Procedure

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

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

Example:

```
Router#install package replace /harddisk:/ncs5700-x64.iso
```

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

Example:

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

Note

You can use a single command to perform both the packaging operation and activating the applying the changes using **install replace /harddisk:/ncs5500-mini-x-<release>.iso** command.

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

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

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

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

Step 4 View the install log.

Example:

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

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

Step 5 Verify that the image is activated successfully.

Example:

```
Router#show install request
```

Step 6 Commit the transaction.

Example:

```
Router#install commit
```

Note

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

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

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

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

Upgrade the System and Install RPMs

In this scenario, you replace the current software with the .iso image and have the possibility to install or remove optional RPMs before applying the changes. You can perform this operation while an atomic-change is already in progress. However, all packaging operations before this command are discarded. The installed software is an exact copy of the software in the ISO after this packaging operation is complete. You can perform all additional packaging operations after this operation and before applying and committing the changes.

Procedure

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

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

Example:

```
Router#install package replace /harddisk:/ncs5700-x64.iso
```

Step 3 Install other RPMs (packages) after the system upgrade operation.

- Configure a repository on the router. For instructions to create a local or a remote repository, see [Create Repository to Access Files for Installing IOS XR Software, on page 3](#).
- Check the available packages in the repository.

Example:

```
Router#show install available
```

- Install the RPMs.

Example:

```
Router#install package add <pkg1> <pkg2> <pkgn>
```

Step 4 Check the status of install operation.

Example:

```
Router#show install request
```

```
User request: install package add xr-bgp
```

```
State: In progress since <date and timestamp>
```

```
Current activity: Package add or other package operation
```

```
Next activity: Await user input
```

```
Time started: <date and timestamp>
```

```
Timeout in: 35m 8s
```

```
Locations responded: 0/1
```

| Location | Packaging operation stage | Notification Phase | Clients responded |
|------------|---------------------------|--------------------|-------------------|
| 0/RP0/CPU0 | Package operations | None in progress | N/A |

Note

The operation ID is a unique ID for each user request. This ID is constructed from the transaction ID, atomic change ID and packaging operation ID that was already used in the commands. For example, if the request is `install commit`, the operation ID is the transaction ID. If the request includes applying an atomic change but not committing the transaction (for example, `install replace /harddisk:/ncs5700_x64.iso`), the operation ID is the atomic change ID. An operation ID of 4.2 indicates a second atomic change in the fourth transaction.

This operation ID is also returned in the action RPC. If an error occurs while the request is initiated, an empty string is returned instead of an operation ID.

When the State changes to Success, activate the new image.

```
Router#show install request
```

```
Wed Sep 14 02:53:21.525 PDT
```

```
User request: install package abort latest
```

```
Operation ID: 2.1.2
```

```
State: Failure since 2022-09-14 02:48:15 UTC-07:00
```

```
Disk space check failed on nodes: 0/0/CPU0. Query 'show install history id 2.1.2 errors' for more details and next steps
```

```
Current activity: Await user input
```

```
Time started: 2022-09-14 02:48:20 UTC-07:00
```

```
The following actions are available:
```

```
install package add
```

```
install package remove
```

```
install package upgrade
```

```
install package downgrade
```

```
install package abort latest
```

```
install package abort all-since-apply
```

```
install apply restart
```

```
install apply reload
```

```
install reimage
```

Note

The **install apply restart** method has the least impact.

Step 5 Activate the new .iso image or RPM on the router by applying the changes.

Example:

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

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

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

Step 6 Verify the image and packages that are activated as part of `install package add` operation is activated successfully.

Example:

```
Router#show install request
```

Step 7 Commit the transaction.

Example:

```
Router#install commit
```

To perform the same step using data models, use the `install-package-replace` RPC on the [Cisco-IOS-XR-install-augmented-act](#) data model.

```
<install-replace>
  <file>iso-name</file>
  <source-type>local</source-type>
  <source>directory-containing-iso</source>
```

```
<commit>true</commit>
</install-replace>
```

Upgrade QDD Optical Modules Through CLI

The QDD optics firmware file needs to be copied to the router manually. Contact Cisco Support to check the QDD firmware version, IOS XR release compatibility, and to obtain the QDD optics firmware file.

Limitation: When ports share a common management interface, IOS XR serializes the firmware upgrade. Serializing and deserializing may delay the upgrade process.

Procedure

Step 1 From the router, copy the QDD firmware file to the hard disk using the following command:

Example: `scp user@10.1.1.1:/home/user/filename harddisk:/`

- When you are using VRF, use the following sample command:

```
scp user@10.1.1.1:/home/user/cl1.bin vrf MGMT harddisk:/

Tue Jan 25 02:57:22.762 UTC
Connecting to 10.1.1.1...
Password:
  Transferred 1484800 Bytes
  1484800 bytes copied in 0 sec (22161194)bytes/sec

RP/0/RP0/CPU0:8808#dir harddisk:/cl1.bin
Tue Jan 25 03:00:47.835 UTC

Directory of harddisk:/cl1.bin
 35 -rw-r--r--. 1 1484800 Jan 25 02:57 dp04qsdd_dp04sfp8_161_10_01.ackit

53461500 kbytes total (42983204 kbytes free)
```

- When you are not using VRF, remove the `vrf MGMT` command:

```
scp user@10.1.1.1:/home/user/cl1.bin harddisk:/
```

Step 2 Run the following commands to upgrade the FPD for QDD optical modules:

Multiport upgrade: `upgrade optics port 0,1,2,3,4 filename /harddisk:/cl1.bin location 0/1/CPU0`

Single port upgrade: `upgrade optics port 0 filename /harddisk:/cl1.bin location 0/1/CPU0`

You can check the firmware upgrade progress using the following command: `show optics firmware upgrade port 0,1,1,2,3,4 location 0/1/CPU0`

Install Optional Packages to Provide Additional Functionality

You can install one or more packages (RPM) that are not already present on the system. The packages are not mandatory for the software to function, but provide additional functionality. Based on your requirement, you

can install or remove these optional packages. The source file can be a repository name, repository url, local filepath, or path to a tar file.

You must specify only the top-level package name that you want to install. The associated dependencies of this package, in the form of card and partition-specific packages, are included automatically. By default, the latest available version of each package is installed. You can also explicitly install a specific version of a package.



Note All Cisco IOS XR images are signed to ensure the authenticity of the software.

This example shows the options to install the optional package `xr-telnet-7.3.17.5.2v1.0.1-1.x86_64.rpm`.

Procedure

Step 1 Install one or more optional packages using one of the following options:

- **Option 1:** Install the package from the local directory:

NCS 5700 Series:

```
Router#install source /harddisk:/files xr-telnet-7.3.1v1.0.1-1.x86_64.rpm
```

Note

The `install source` command automatically applies the changes. Use this command to install optional packages. To upgrade existing packages, see [Upgrade the System to Obtain Bug Fixes](#).

- **Option 2:** Install the package from a configured remote repository:

```
Router#install source install-repo xr-telnet
```

Here, `install-repo` is the name of the repository. For repository configuration, see [Create Repository to Access Files for Installing IOS XR Software, on page 3](#).

- **Option 3:** Install the package from a repository URL:

```
Router#install source http://72.16.0.0:3333/remote-repo xr-telnet
```

- **Option 4:** Add the package and apply the change. The package must be available in the repository.

NCS 5700 Series:

```
Router#install package add xr-telnet-7.3.1v1.0.1-1.x86_64.rpm
Router#install apply {restart | reload}
```

More than one package can be installed using a single packaging operation. Use the following command:

```
Router#install source <path-to-source> <package 1> <package 2> ... <package n>
```

For example,

NCS 5700 Series:

```
Router#install source /harddisk:/files xr-telnet-7.3.1v1.0.0-1.x86_64.rpm
xr-mcast-7.3.1v1.0.0-1.x86_64.rpm
```

To perform this task using data models, use the `install` RPC on the [Cisco-IOS-XR-install-act](#) data model. Here is an example with an HTTP repository:

```

<install>
  <packages>
    <packagename>pkg1</packagename>
    <packagename>pkg2</packagename>
    ...
    <packagename>pkgn</packagename>
  </packages>
  <source-type>http</source-type>
  <source><path-to-source></source>
</install>

```

Step 2 Commit the operation.

Example:

```
Router#install commit
```

Step 3 Check the status of install operation.

Example:

NCS 5700 Series:

```

Router# show install history last package
Tue Jul 14 11:09:19.748 UTC
2021-07-14 11:08:12 UTC      Packaging operation 1.1.1 started
2021-07-14 11:08:12 UTC      Add
2021-07-14 11:08:12 UTC      xr-telnet
2021-07-14 11:09:10 UTC      Packaging operation 1.1.1 success

Location 0/RP0/CPU0
  Add xr-telnet-7.3.1v1.0.0-1.x86_64
  Add xr-telnet-36cad6c174d48ffe-7.3.1v1.0.0-1.x86_64
  Add xr-telnet-ncs5700-7.3.1v1.0.0-1.x86_64
  Add xr-telnet-alf05ad3091205a8-7.3.1v1.0.0-1.x86_64

```
