Additional Install Operations

After you upgrade your system, based on your requirement, you can perform additional install operations:

View the Version of Installed Packages

The router is shipped with a pre-installed operating system. You can view the version of the installed software and the active packages. If you have upgraded your system, installed additional packages or bug fixes, you can view the version of the committed packages.

Review the software version information:

- Package name and version
- User who built the package
- Time the package was built
- Build workspace
- Build host
- ISO label:
  - Label is present if GISO boots using PXE boot
  - Label is present if GISO is installed using the `install replace` method
  - Label reverts to default (only release version) if there is any change since the time the image with the label was installed.
  - Label is nullified and reverts to default if an RPM is added or removed on top of an existing GISO.
  - Label is repopulated if an RPM from the GISO is added or removed and a rollback operation is performed.

- Copyright information
- Hardware information

Step 1 View the version of the Cisco IOS XR software, and its various software components that are installed on the router.

Example:

The following example shows the version information for a non-GISO image:

NCS 5500 Series:

```
Router#show version
Cisco IOS XR Software, Version 7.5.2 LNT
Copyright (c) 2013-2022 by Cisco Systems, Inc.
Build Information:
Built By     : xyz
Built On     : Tue June 07 19:43:44 UTC 2021
Build Host   : iox-lnx-064
Workspace    : ../ncs5500/ws
Version      : 7.5.2
Label        : 7.5.2

cisco NCS5500L (D-1563N @ 2.00GHz)
cisco NCS-55A1-36H-S (D-1563N @ 2.00GHz) processor with 32GB of memory
ios uptime is 3 weeks, 1 day, 10 hours, 11 minutes
NCS-55A1-36H-S Fixed Scale HW Flexible Consumption Need Smart Lic
```

The following example shows the version information for a GISO image. The customer label is appended to the `Label` field in the GISO image:

NCS 5500 Series:

```
Router#show version
Cisco IOS XR Software, Version 7.5.2 LNT
Copyright (c) 2013-2022 by Cisco Systems, Inc.
Build Information:
Built By     : xyz
Built On     : Tue June 07 19:43:44 UTC 2021
Build Host   : iox-lnx-064
Workspace    : ../ncs5500/ws
Version      : 7.5.2
Label        : 7.5.2-Customer_Label

cisco NCS5500L (D-1563N @ 2.00GHz)
cisco NCS-55A1-36H-S (D-1563N @ 2.00GHz) processor with 32GB of memory
ios uptime is 3 weeks, 1 day, 10 hours, 11 minutes
NCS-55A1-36H-S Fixed Scale HW Flexible Consumption Need Smart Lic
```

You can also use the `get` RPC on the `install.version` data model.

Step 2 View the active packages.

Example:

NCS 5500 Series:

```
Router#show install active summary
Mon Jun  6 04:18:26.684 UTC
Active Packages:    XR: 145    All: 1157
Label:              7.5.2
Software Hash:      28dd70ef227aeca3d3fd3ecf8d1792a4f51fabb299ec7d38725869575fd9cfaf

Optional Packages                                                         Version
```

```
---------------------------------------------------- ---------------------------
xr-bgp                                                         7.5.2v1.0.0-1
xr-ipsla                                                       7.5.2v1.0.0-1
xr-is-is                                                       7.5.2v1.0.0-1
xr-lldp                                                        7.5.2v1.0.0-1
xr-mcast                                                       7.5.2v1.0.0-1
xr-mpls-oam                                                    7.5.2v1.0.0-1
xr-ncs5500l-mcast                                              7.5.2v1.0.0-1
xr-ncs5500l-netflow                                            7.5.2v1.0.0-1
xr-netflow                                                     7.5.2v1.0.0-1
xr-ospf                                                        7.5.2v1.0.0-1
xr-perf-meas                                                   7.5.2v1.0.0-1
xr-perfmgmt                                                    7.5.2v1.0.0-1
xr-track                                                       7.5.2v1.0.0-1
```

You can also use the `get` RPC on the `install.packages.active.node` and the `install.packages.active.node.summary` data models.

To understand the data model structure and its arguments, see the Obtain Data Models for Install Operation topic.

Build a Golden ISO

Table 1: Feature History Table

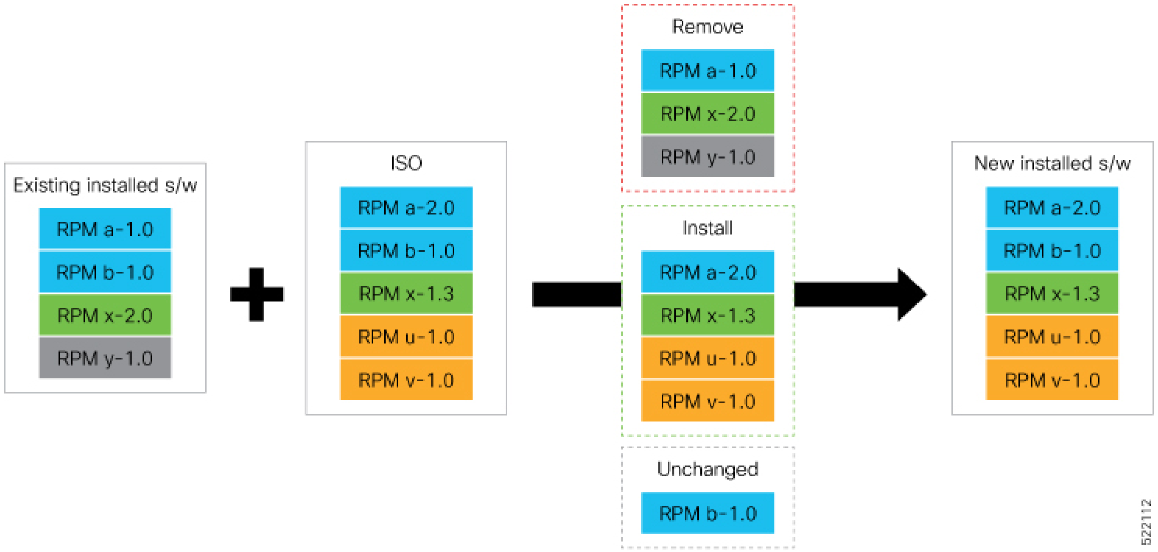
Feature Name	Release Information	Description
Build Golden ISO (GISO) Using `gisobuild.py` Tool	Release 7.6.1	This feature allows you to build your GISO image without support from Cisco. You can now select the install files, add your RPMs, repackage them as a custom image, and install the image. In previous releases, you had to contact Cisco to get your GISO built.

Golden ISO (ISO) upgrades the router to a version that has a predefined set of RPMs with a single operation. For example, you can create a customized ISO with the base OS package and specific optional RPMs based on your network requirements.

GISO supports automatic dependency management, and provides these functionalities:

- Builds RPM database of all the packages present in package repository.
- Skips and removes Cisco RPMs that do not match the base ISO version.
- Skips and removes third-party RPMs that are not part of already existing third-party base package in the base ISO.

Note Install operation over IPv6 is not supported.



Step 1 Contact Cisco Support to build the GISO image with the set of packages based on your requirement.

Step 2 Build GISO image using `gisobuild.py` tool.

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

- Base mini-x.iso (mandatory)
- Set of packages to install (Cisco signed packages)
- XR configuration file (optional)
- Label for golden ISO (optional)
- ZTP initialization `ztp.ini` file (optional)

Note GISO build tool verifies the RPM dependecnies and RPM signatures. GISO build fails if the RPM is unsigned or incorrectly signed.

a) Copy the repository from the Github location to an offline system or external server where the GISO will be built.

b) Run the script `gisobuild.py` and provide parameters to build the GISO image. Ensure that all RPMs and SMUs are present in the same directory or on a repository.

Example:

```
$ ./giso/src/gisobuild.py --iso <input iso> --repo <rpm repo1 rpm_repo2> --pkglist <pkg1 pkg2
pkg3>
    --xrconfig <config.cfg> --ztp-ini <ztp.ini> --label <label>
    --out-directory <out_directory> --clean
```

The following parameters can be provided as input to the GISO build tool:

- --iso: ISO path to mini.iso or full.iso file
- --xrconfig: XR configuration file
- --label: GISO label
- --repo: Path to repositories containing RPMs and tarballs

- --pkglist: Optional RPMs or SMUs to package
- --ztp-ini: Path to the ZTP initialization file
- --remove-packages: Remove RPMs from the GISO. To remove multiple RPMs, separate the RPM names using comma. For example, `--remove-packages xr-bgp,xr-mcast` command removes the xr-bgp and xr-mast packages from GISO
- --out-directory: Output directory to store output of the operations performed on the file
- --clean: Delete contents of the output directory
- --skip-dep-check: Skip dependency checking between files
- --version: Print version of the tool
- --pkglist: Optional RPM or SMU to package
- --yamlfile: Provide CLI arguments via YAML markup file
- --docker: Load and run pre-built docker image

The tool uses the input parameters to build the GISO image.

Step 3 Copy the GISO image to the `/harddisk:` location on the router.

Step 4 Upgrade the system to replace the current software with the .iso image, and install the RPMs.

Example:

```
Router#install replace <source location> <giso name.iso>
```

If you are using a configuration file in GISO, use the following command to extract and replace the configuration.

```
Router# install replace <source location> <GISO-with-cfg>-<platform>.iso
```

Note The default option is to replace the existing configuration. The install operation applies the configuration from a GISO, the router reboots to activate the configuration.

Step 5 View the version information for the GISO image. You can include a label to indicate the runing software version on the router. For example, create a label v1 for the current GISO version. When you rebuild GISO with additional RPMs, you can create a label v2 to distinguish the builds.

Example:

```
Router#show version
Cisco IOS XR Software, Version 7.5.2 LNT
Copyright (c) 2013-2022 by Cisco Systems, Inc.
Build Information:
Built By     : xyz
Built On     : Tue June 07 19:43:44 UTC 2021
Build Host   : iox-lnx-064
Workspace    : ../ncs5500/ws
Version      : 7.5.2
Label        : 7.5.2-Customer_Label

cisco NCS5500L (D-1563N @ 2.00GHz)
cisco NCS-55A1-36H-S (D-1563N @ 2.00GHz) processor with 32GB of memory
ios uptime is 3 weeks, 1 day, 10 hours, 11 minutes
NCS-55A1-36H-S
NCS55B1 Fixed Scale HW Flexible Consumption Need Smart Lic
```

Upgrade the System to Obtain Bug Fixes

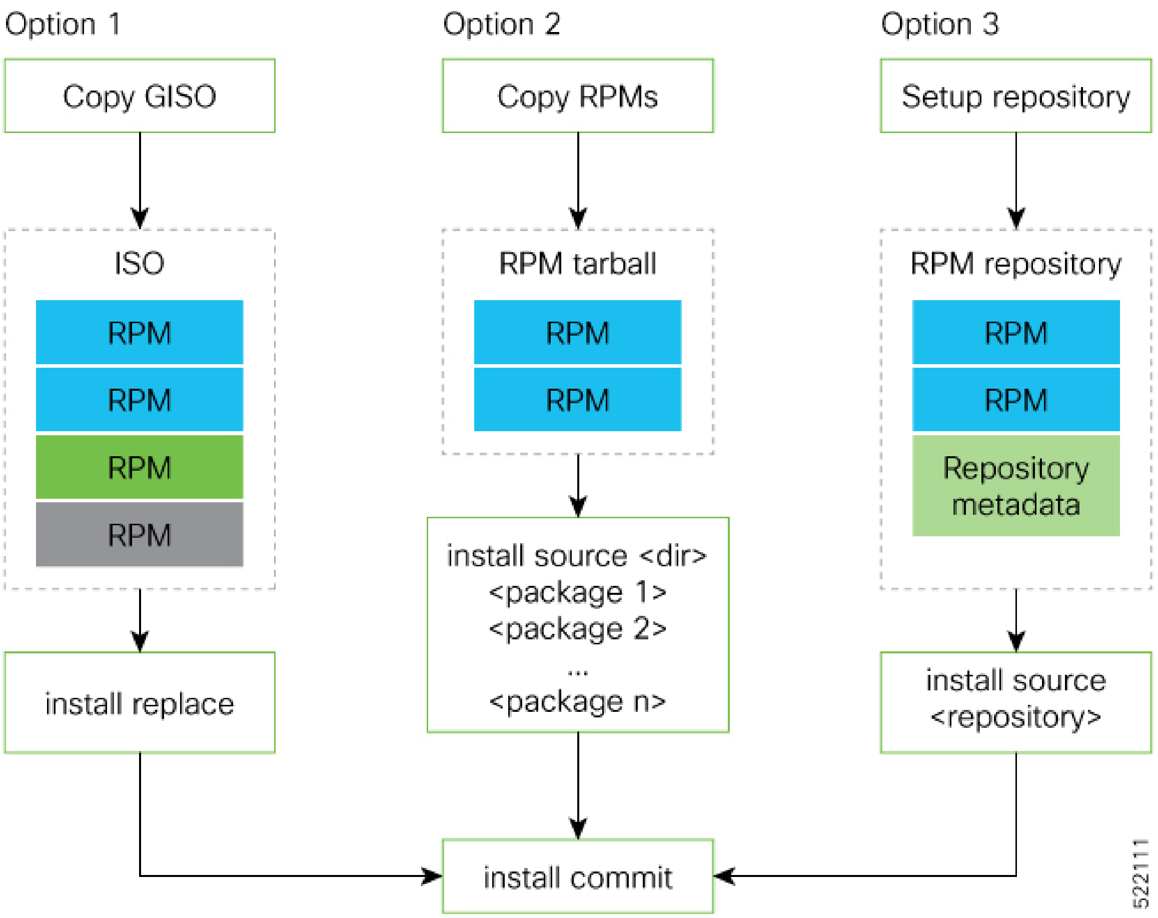
You can upgrade the system to obtain all available bug fixes or choose specific bug fix using bug ID.

Bug fixes are available as optional RPMs. The bug fixes replace packages in the base ISO without adding up to the image size. For example, even if you add 20 RPMs to the GISO, the GISO file size will remain the same as the initial ISO.

You can download the bug fix RPMs from the Cisco Software Download portal .

The `README` file provides the relevant information about the bug fix and also identifies the dependencies, if any, where other bug fix RPMs may be required for a complete fix.

The following image shows the options to install the bug fix RPMs.



The software is split into modular blocks and the package manager infrastructure computes the dependencies between the blocks. Each block of software has a top-level package and various partition-level packages. Bug fixes that span multiple blocks may lead to creating multiple dependent packages that are built as part of earlier bug fixes.

Note We recommend that you leverage the GISO workflow. Contact Cisco Support to build a new GISO with the required bug fixes. The RPMs are present in the initrd, which is the initial RAM disk for the boot loader, and this requires that the package is signed by Cisco. You can install GISO using a single `install replace` operation. For more information about GISO, see the *Build a Golden GISO* section.

However, if you do not prefer using the GISO, here are a few alternative ways to install bug fixes:

- Create tarballs to install one or more bug fixes. For example, if you are installing bug fixes CSCxx11111, CSCyy22222 and CSCzz33333, you can use the individual tarball files and create a single tarball file.
- Use a Dandified Yum- or DNF repository to install, update, or remove relevant bug fixes. See, Create Repository to Access Files for Installing IOS XR Software.

Note The RPMs can be installed from the TAR file using CLI or Yang data model.

- CLI—
  - Packaging operation: `install package add source /harddisk:/-<version>.tgz synchronous`
  - Packaging operation and atomic change: `install source /harddisk:/-<version>.tgz synchronous`
- Yang Data Model—

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:
base:1.0" message-id="101">
<install-package-add xmlns=
"http://cisco.com/ns/yang/Cisco-IOS-XR-install-augmented-act">
<source-type>tar</source-type>
<source>/misc/disk1/<platform>-<version>.tgz</source>
</install-package-add>
</rpc>
```

Note Use the RPM repository to harvest the benefits of package manager. The package manager queries the available packages, and downloads only those packages and their dependencies that are needed for installation.

Step 1 View the list of available bug fixes.

Example:

Step 2 Install the bug fix or package using one of the following options:

- Install the package where the bug fix is applied.

  This task can also be accomplished using Cisco-IOS-XR-install-augmented-oper data model.

  Apply the changes.

```
Router#install apply [reload | restart]
```

Note To identify if you must reload or restart the system while applying the changes, use one of these two methods:

- History of last transaction

  ```
  Router#show install history last transaction verbose
  2019-09-11 17:01:46 UTC    Transaction 3 started
  2019-09-11 17:01:46 UTC      Atomic change 3.1 started
  2019-09-11 17:01:46 UTC      Packaging operation 3.1.1 started
  2019-09-11 17:16:46 UTC    Transaction 3 complete

  Least impactful apply method: process restart
  ```

  The command also displays the information about the changes to files and processes because of the install operation, and the package operations carried out on each node.

- Show install request

  ```
  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
  ```

  Here, both `install apply restart` and `install apply reload` options are available. In this case, use `install apply restart` command because the impact on the system is the least. But when only an `install apply reload` option is available, then reload is the only option to apply the change.

- Install the optional package. Changes are applied automatically.

  Attention Automatic change may trigger a reload of the router depending on the package that is installed.

  ```
  Router#install source /harddisk:/files xr-<platform>-core-7.3.1v1.0.1-1
  ```

  Note Packages can also be installed using the package name.

  ```
  Router#install source /harddisk:/files  xr-<platform>-core
  ```

- This task can also be performed using YANG data models. Use `install` RPC on the Cisco-IOS-XR-install-act data model. Here is an example usage with an HTTP repository:

  ```
  <install>
     <packages>
       <packagename>pkg1</packagename>
     </packages>
     <source-type>http</source-type>
  ```

```
       <source><path-to-source></source>
    </install>
```

Step 3 View the state of the packaging operation.

Example:

```
Router#show install request
User request: install package upgrade xr-<platform>-core-<version> xr-core-<version>
Operation ID: 2.1.2
State:        In progress since

Current activity:    Initiate operation
Next activity:       Begin transaction
Time started:        2019-06-25 07:41:06

No per-location information.
```

Step 4 View the log to ensure that the installation is successful.

Example:

```
Router#show install log
2019-06-25 07:41:06 UTC    Transaction 1 started
2019-06-25 07:45:08 UTC    Upgrade (Success)
2019-06-25 07:45:08 UTC    xr-<platform>-core-<version>
2019-06-25 07:45:08 UTC    xr-core-<version>
2019-06-25 07:57:02 UTC    Atomic change 1.1 successfully applied by reload
```

Step 5 View the history of the install operation.

Example:

```
Router#show install history table
  Transaction             Atomic Change          Packaging Operations
---------------------  --------------------------  -----------------------------------
Id   Status          Id  Method   Status      Id  Operation   Inputs Status
---------------------  --------------------------  -----------------------------------
 1   In progress     1   Reload   Success      1   Upgrade        1  Success
```

The command can also be used to view more details if there is a failed operation.

Use **show install history id <operation-id>** command to filter the history of install information by ID. IDs are of the form `<transaction id>.<atomic id>.<packaging id>`.

```
Router#show install history id ?
  WORD Specify an operation ID (e.g. 1, 1.2, 1.2.3)
```

Use **show install history last** command to view the last packaging operation, atomic change, or transaction.

```
Router#show install history last ?
     atomic-change  Show the last atomic change
     package        Show the last packaging operation
     transaction    Show the last transaction
```

Step 6 After the operation is complete, verify that the packages and `and xr-core-7.5.2v1.0.1-1` are installed and active.

Example:

Example:

Example:

```
Router#show install active summary
```

The version has changed. The version `1.0.1-1` indicates that the bug fix is installed.

This task can also be accomplished using data models. Use the `get` RPC for `install.fixes.active` operation using Cisco-IOS-XR-install-augmented-oper data model.

Step 7 Commit the changes for the changes to persist after a reload operation.

Example:

```
Router#install commit
```

Step 8 View the list of bug IDs for which fixes are committed.

Example:

```
Router#show install fixes committed
```

This task can also be accomplished using data models. Use the `get` RPC for `install.fixes.committed` operation using Cisco-IOS-XR-install-augmented-oper data model.

Step 9 View the list of active bug fix RPMs.

Example:

```
Router#show install fixes active
```

This task can also be accomplished using data models. Use the `get` RPC for `install.fixes.active` operation using Cisco-IOS-XR-install-augmented-oper data model.

Downgrade to a Previously Installed Package

You can downgrade a package to a previously installed version. By default, the subsequent previous version (version previous to the current version) is installed. Also, you can downgrade the software to a specific version of interest. To remove a bug fix RPM from the installed packages, downgrade the package to a version where the fix was not applied.

For release specific downgrade caveats see Caveats, on page 11.

Note While downgrading, you can choose any previous version, including the base version of the RPM. However, when downgrading a bug fix RPMs, ensure that you also consider all dependencies of the current version.

Bug fix RPM is an upgrade to the existing package. The action of removing a bug fix RPM either removes the entire feature, or fails if the package is mandatory.

The following example shows the package is downgraded to . The path to source can be a local location or a configured repository.

Before you begin

Ensure you have access to the previously installed package and its source.

Step 1 Downgrade the package using one of the following options:

- Downgrade the package where the fix was applied. When multiple older versions of the package are present in the configured repositories, the immediate previous version of the package is installed. Use caution when using this command as the current version of the package is removed completely.

  ```
  Router#install package downgrade xr-telnet
  ```

  Apply the changes.

  ```
  Router#install apply [reload | restart]
  ```

  Attention To identify whether to reload the router or restart the affected processes as part of the apply operation, use either **show install history last transaction verbose** command or **show install request** command.

- Install a specific earlier version of the optional package. The changes are applied automatically.

  Attention An automatic change may trigger a reload of the router depending on the package being downgraded.

  ```
  Router#install source <path-to-source> xr-telnet-7.0.1v1.0.0
  ```

- Use `install` RPC on the Cisco-IOS-XR-install-act data model. Here is an example usage with a local repository:

  ```
  <install>
   <packages>
     <packagename>


  </packagename>
   </packages>
     <source>file://<path-to-source>/</source>
  </install>
  ```

  The package version is downgraded to .

Step 2 Commit the operation.

Example:

```
Router#install commit
```

Caveats

Table 2: Upgrade Caveats

From	To	Bridge SMUs Required	Caveats
7.5.2	7.10.1 and later	Yes	1*, 2*
7.6.1	7.10.1 and later	Yes	1*, 2*
7.6.2	7.10.1 and later	None	1*
7.7.1	7.10.1 and later	None	1*

From	To	Bridge SMUs Required	Caveats
7.7.2	7.10.1 and later	None	1*
7.8.1	7.10.1 and later	None	1*
7.8.2	7.10.1 and later	None	1*
7.9.1	7.10.1 and later	None	1*
7.9.2	7.10.1 and later	None	1*

1*: You can't roll back using the **install rollback** command.

2*: Ensure that a reload bridging SMU (CSCwd71524) is installed.

Note CSCwd71524:

- When you upgrade from earlier than Release 7.10.1 to Release 7.10.1, system supports the installation process seemlessly.
- When you downgrade from Release 7.10.1, system preserves the present configuration and the install history from last transaction.

The following downgrade caveats are applicable for Release 7.10.1 and later:

Table 3: Downgrade Caveats

From	To	Bridge SMUs Required	Caveats
7.10.1 and later	7.5.2	Yes	***, A*, B*
7.10.1 and later	7.6.1	Yes	***, A*, B*
7.10.1 and later	7.6.2	Yes	***, A*, B*
7.10.1 and later	7.7.1	Yes	***, A*, B*
7.10.1 and later	7.7.2	Yes	***, A*, B*
7.10.1 and later	7.8.1	Yes	***, A*, B*
7.10.1 and later	7.8.2	Yes	***, A*, B*
7.10.1 and later	7.9.1	Yes	***
7.10.1 and later	7.9.2	Yes	***

***

- You don't need to run the **install commit** command after a downgrade operation because the operation is automatically committed.
- You can't roll back after a downgrade. To revert to the previous IOS XR previous version, replace or reimage to the relevant ISO.
- IOS XR configuration history is lost after a downgrade, but the router preserves the latest configuration.
- Install history from the last transaction is preserved after a downgrade operation.
- Downtime takes a longer time as the operation is performed through reimage.
- You can't downgrade using the **install package replace** command. Instead, use the **install replace** command to downgrade.
- Ensure that you reinstall third-party application once you complete the downgrade.
- You must re-install the *Type 6 masterkey* and reapply the configuration encrypted by it because they are lost after the downgrade.
- You must regenerate crypto keys and certificates after a downgrade.
- PXE recovery is required if the image downgrading isn’t bootable.

A*: You can't downgrade to the base ISO. You can downgrade to a GISO containing the bridge SMU (CSCwd71524).

B*: You must recover the router through PXE if a power cycle occurs during the downgrade.

Use the **show install upgrade-matrix running** command to view the caveats.

Roll Back Software to a Previously Saved Installation Point

You can roll your system software back to a previous version. This could be used to discard an ongoing install operation, or to undo an install operation that has already been committed. After each commit operation, the system saves a record of the committed software packages. Each record is a restoration point, and is assigned a unique ID. This ID is known as a transaction ID. You can use the transaction ID to roll back the software to a restoration point associated with this ID. Up to 900MB of space is allowed for rollback points, instead of a specific number of rollback points.

Note

- Use transaction ID 0 to roll back to the software that was present after the system booted for the first time.
- If you commit an install transaction using **install commit** command, the GISO ZTP configuration is saved along with the rest of the software changes. This means that if you use the **install rollback** or **install package rollback** command to revert the software to the state of a previous transaction, the GISO ZTP configuration is also reverted to its previous state. To undo this install operation, as well as the change in GISO ZTP configuration, use **install package abort** command. If there is no GISO ZTP configuration saved at the end of the install transaction, the existing GISO ZTP configuration is reverted to the previous state when the rollback operation of that transaction is applied.

Step 1 View the list of available transaction IDs.

Example:

```
Router# show install rollback list-ids
```

Step 2 Explore the main packages that can be installed if you roll the software back to the specific transaction ID.

Example:

```
Router# show install rollback id <id>
```

Step 3 View the relative changes that are made to the currently installed software if it is rolled back to a transaction ID.

Example:

```
Router# show install rollback id <id> changes
```

To perform these tasks using data models, use the `get` RPC on the Cisco-IOS-XR-install-augmented-oper data model.

```
<rpc>
  <get>
   <filter type="subtree">
      <install xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-install-augmented-oper">
      <rollback/>
     </install>
   </filter>
  </get>
</rpc>
```

Step 4 Roll back to the software associated with the specific transaction ID.

Example:

```
Router# install rollback <id> [commit]
```

If you want to apply the change and roll back to the associated transaction ID, commit the change. You can also include the keyword `noprompt` in the command to enable the system to bypass your permission to reload the router.

Attention This roll back operation installs the previous software and also applies the change automatically. This may reload the router depending on the package that is rolled back.

Alternatively, use the **install package rollback** command to only roll back the package but not apply the changes. You can check whether the router will reload or restart if you apply the change using the **show install history last transaction verbose** command or **show install request** command. Based on the command output, you can take the appropriate action using **install apply reload | restart** command to either reload or restart the system. Use the **install commit** command to commit the transaction.

To perform this task using data models, use the `install-rollback` RPC on the Cisco-IOS-XR-install-augmented-oper data model.

```
<rpc>
 <install-rollback xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-install-augmented-act">
  <commit>true</commit>
   <transaction-id>0</transaction-id>
 </install-rollback>
</rpc>
```

To understand the data model structure and its arguments, see the Access the Install-Related Data Model.

Step 5 Commit the operation.

Example:

```
Router#install commit
```

Telemetry Support for Install Operations

Table 4: Feature History Table

Feature Name	Release Information	Description
Stream Telemetry Data about Install Operations	Release 7.5.2	You can stream telemetry data for install-related details such as active and committed packages, view the progress of install operations, retrieve the image version, and view the error messages with recovery information when an operation fails.

To stream telemetry data that is related to software installation, you must create subscriptions to the sensor paths in the YANG data models. See *Obtain Data Models for Install Operation* for the list of supported data models. For information about establishing a telemetry session and creating subscriptions, see the *Telemetry Configuration Guide for Cisco NCS 5500 Series Routers*.

Stream Telemetry Data About	Description	YANG Path

Summary of active packages	Data is streamed after a successful **apply** operation. An active package is the software currently running on the system.	`Cisco-IOS-XR-install-oper:`
		`install/packages/active/summary`

Summary of committed packages	Data is streamed after a successful **commit** operation. A package that is committed remains active following a system reload.	`Cisco-IOS-XR-install-oper:`
		`install/packages/committed/summary`

Status of the last request operation	Data is streamed when starting a new request and also when entering an `idle` state. If the operation has failed, this includes error messages along with recovery state.	`Cisco-IOS-XR-install-oper:`
		`install/request`

Image version and GISO label	Data is streamed after a successful **apply** operation.	`Cisco-IOS-XR-install-oper:`
		`install/version`

Packaging information	Data is streamed at the start and end of a packaging operation.	`Cisco-IOS-XR-install-augmented-oper:`
		`install/history/latest-packaging-operation`

Stream Telemetry Data About	Description	YANG Path

Atomic information	Data is streamed at the start and end of **apply** operation.	`Cisco-IOS-XR-install-augmented-oper:`
		`install/history/latest-atomic-change`

Transaction information	Data is streamed at the start, in progress, and end of a **commit** operation.
	Note After a transactional rollback, some of the data such as summary of active packages, image version can change. However, telemetry events are not sent after the reload operation.
		`Cisco-IOS-XR-install-augmented-oper:`
		`install/history/latest-transaction`

View Features and Capabilities Supported on a Platform

Table 5: Feature History Table

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

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

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

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

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

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

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

View the List of Supported Features

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

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

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

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

View the List of Supported ACL Features

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

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

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

View the List of Supported ACL Features for Specific RP

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

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

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