



Customize Installation using Golden ISO

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

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

GISO is built using a build script `gisobuild.py` available on the github location [Github](#) location.

When a system boots with GISO, additional SMUs and RPMs in GISO are installed automatically, and the router is pre-configured with the XR configuration in GISO. For more information about downloading and installing GISO, see [Install Golden ISO, on page 5](#).

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

- Initial deployment of the router
- Software disaster recovery
- System upgrade from one base version to another
- System upgrade from same base version but with additional SMUs
- Install update to identify and update dependant packages
- [Limitations, on page 1](#)
- [Golden ISO Workflow, on page 2](#)
- [Build Golden ISO, on page 3](#)
- [Install Golden ISO, on page 5](#)
- [Install Replace with Golden ISO, on page 9](#)
- [Upgrade the System Using IOS XR Configuration File in GISO, on page 11](#)

Limitations

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

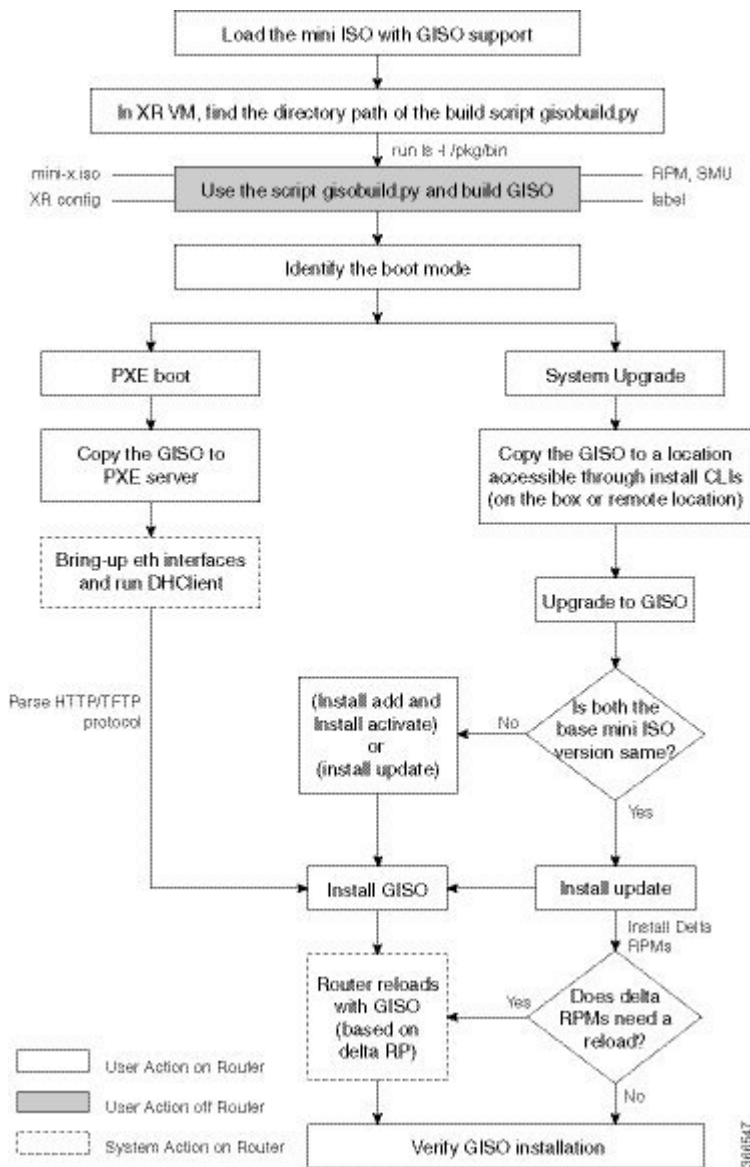
- Building and booting GISO for asynchronous package (a package of different release than the ISO) is not supported.

- Verifying the XR configuration is not supported in the GISO build script `gisobuild.py`.
- Renaming a GISO build and then installing from the renamed GISO build is not supported.
- Install operation over IPv6 is not supported.

Golden ISO Workflow

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

Figure 1: Golden ISO Workflow



Build Golden ISO

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

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

- Builds RPM database of all the packages present in package repository.
- Scans the repositories and selects the relevant Cisco RPMs that matches the input iso.
- Skips and removes third-party RPMs that are not SMUs of already existing third-party base package in mini-x.iso.
- Displays an error and exits build process if there are multiple base RPMs of same release but different versions.
- Performs compatibility check and dependency check for all the RPMs. For example, the child RPM `ncs5000-mpls-te-rsvp` is dependent on the parent RPM `ncs5000-mpls`. If only the child RPM is included, the Golden ISO build fails.

Build Golden ISO Using Script

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

- Base mini-x.iso (mandatory)
- XR configuration file (optional)
- one or more Cisco-specific SMUs for host, XR and System admin (mandatory)
- one or more third-party SMUs for host, XR and System admin (mandatory)
- Label for golden ISO (optional)
- Optional RPMs



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

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

To build GISO, perform the following steps:

Before you begin

- To upgrade from a release that did not support GISO to a release supporting GISO version, it is mandatory to first upgrade to mini ISO with GISO support. For NCS 5001 series routers, upgrade to release 6.2.2 or later.
- The system where GISO is built must meet the following requirements:
 - System must have Python version 2.7 and later.

- System must have free disk space of minimum 3 to 4 GB.
 - Verify that the Linux utilities `mount`, `rm`, `cp`, `umount`, `zcat`, `chroot`, `mkisofs` are present in the system. These utilities will be used by the script. Ensure privileges are available to execute all of these Linux commands.
 - Kernel version of the system must be later than 3.16 or later than the version of kernel of Cisco ISO.
 - Verify that a `libyaml` rpm supported by the Linux kernel is available to successfully `import yaml` in the tool.
 - User should have proper permission for security rpm(k9sec-rpm) in rpm repository, else security rpm would be ignored for Golden ISO creation.
- The system from where the `gisobuild.py` script is executed must have root credentials.

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

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

Example:

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

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

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

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

```
System requirements check [PASS]
```

```
Platform: ncs5000 Version: <version>
```

```
Scanning repository [repository-path]...
```

```
Building RPM Database...
```

```
Total 3 RPM(s) present in the repository path provided in CLI
```

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

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

```
...RPM compatibility check [PASS]
```

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

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

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

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

```
...RPM compatibility check [PASS]
```

```
Building Golden ISO...
```

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

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

...Golden ISO creation SUCCESS.

Golden ISO Image Location: <directory-path>/ncs5000-golden-x.iso-<version>
Detail logs: <directory-path>/Giso_build.log-2017-07-10:18:17:43.395586
```

where:

- -i is the path to mini-x.iso
- -r is the path to RPM repository
- -c is the path to XR config file
- -l is the golden ISO label
- -h shows the help message
- -v is the version of the build tool `gisobuild.py`
- -m is to build the migration tar to migrate from IOS XR to IOS XR 64 bit

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

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



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

What to do next

Install the GISO image on the router.

Install Golden ISO

Golden ISO (GISO) automatically performs the following actions:

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

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

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

- **PXE boot:** when the router is booted, the boot mode is identified. After detecting PXE as boot mode, all available ethernet interfaces are brought up, and DHCPClient is run on each interface. DHCPClient script parses HTTP or TFTP protocol, and GISO is downloaded to the box.
- **System Upgrade** when the system is upgraded, GISO can be installed using **install add**, **install activate**, or using **install replace** commands.

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

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

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

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

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

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

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

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

```
Router#admin

sysadmin-vm:0_RSP0# run chvrf 0 ssh rp0_admin
[sysadmin-vm:0_RSP0:~]$ ssh my_host
[host:~]$ cd /misc/disk1/
[host:~]$ ./usb-install-712-or-latest.sh asr9k-goldenk9-x64-7.0.2-dr.iso /dev/sdc
EFI
```

```
Preparing USB stick for EFI
parted gpt: Failed to create partition - continuing ...
Create filesystem on /dev/sdc1
Mounting source iso at //misc/disk1/cdtmp.CnuKnA
Mounting destination /dev/sdc1 at //misc/disk1/usbdev.SSBb4R
Copying image to USB stick
Initrd path is //misc/disk1/cdtmp.CnuKnA/boot/initrd.img
Getting boot
3749342 blocks
Copying boot
Copying initrd.img
Copying signature.initrd.img
Copying certs
Creating grub files
Copying /misc/disk1/asr9k-goldenk9-x64-7.0.2-dr.iso in USB Stick
USB stick set up for EFI boot!
```

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

The options to upgrade the system are as follows:

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

```

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

Current label: <None>

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

Collecting software state..

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

Label within GISO: <label-name>

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

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

Install add operation successful

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

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

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

This install operation will reload the system, continue?

    [yes/no]:[yes] Install operation will continue in the background
exec-timeout is resumed.

```

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

- **system upgrade across releases from version1 GISO to version2 GISO:** Both the GISO images have different base versions. Use **install add** and **install activate** commands, or **install replace** command to perform the system upgrade. The router reloads after the upgrade with the version2 GISO image.

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

```
sysadmin-vm:0_RP0# show install repository all
Admin repository
-----
ncs5000-sysadmin-6.2.2
ncs5000-sysadmin-hostos-6.2.2-r622.CSCcv10001.admin.x86_64
ncs5000-sysadmin-hostos-6.2.2-r622.CSCcv10001.admin.arm
ncs5000-sysadmin-system-6.2.2-r622.CSCcv10005.x86_64
ncs5000-sysadmin-system-6.2.2-r622.CSCcv10005.arm
....
XR repository
-----
ncs5000-iosxr-mgbl-3.0.0.0-r622.x86_64
ncs5000-xr-6.2.2
....
Host repository
-----
host-6.2.2
```

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

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

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

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

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

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



```

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

Golden ISO Rpms:
  xr rpms in golden ISO:
    ncs5000-k9sec-x64-2.2.0.1-r622.CSCxr33333.x86_64.rpm
    openssh-scp-6.6p1.p1-r0.0.CSCTp12345.xr.x86_64.rpm
    openssh-scp-6.6p1-r0.0.xr.x86_64.rpm
    ncs5000-mpls-x64-2.1.0.0-r622.x86_64.rpm
    ncs5000-k9sec-x64-2.2.0.0-r622.x86_64.rpm

  sysadmin rpms in golden ISO:
    ncs5000-sysadmin-system-6.2.2-r622.CSCcv11111.x86_64.rpm
    ncs5000-sysadmin-system-6.2.2-r622.CSCcv11111.arm.rpm
    openssh-scp-6.6p1-r0.0.admin.x86_64.rpm
    openssh-scp-6.6p1-r0.0.admin.arm.rpm
    openssh-scp-6.6p1.p1-r0.0.CSCTp12345.admin.x86_64.rpm
    openssh-scp-6.6p1.p1-r0.0.CSCTp12345.admin.arm.rpm
    ncs5000-sysadmin-hostos-6.2.2-r622.CSCcv10001.admin.x86_64.rpm
    ncs5000-sysadmin-hostos-6.2.2-r622.CSCcv10001.admin.arm.rpm

  host rpms in golden ISO:
    openssh-scp-6.6p1-r0.0.host.x86_64.rpm
    openssh-scp-6.6p1-r0.0.host.arm.rpm
    openssh-scp-6.6p1.p1-r0.0.CSCTp12345.host.x86_64.rpm
    openssh-scp-6.6p1.p1-r0.0.CSCTp12345.host.arm.rpm

```

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

Install Replace with Golden ISO

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

Example:

```

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

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

```

```

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

```

```

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

```

```

Install operation 13 finished successfully

```

```

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

Router# Install operation 46 finished successfully

```

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

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

For example,

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

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

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

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

Step 2 show version

Example:

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

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

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

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

Upgrade the System Using IOS XR Configuration File in GISO

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

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

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



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

This section shows replacing the current software version with image.

Table 1: Feature History Table

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

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

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



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

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

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

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

Table 2: Configuration File in GISO

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

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

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

Example:

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

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

Example:

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

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

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

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

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

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

Step 4 View the install log.

Example:

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

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

Step 5 Verify that the image is activated successfully.

Example:

```
Router#show install request
```

Step 6 Commit the transaction.

Example:

```
Router#install commit
```

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

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

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

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