



## 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. For more information about the build script and the steps to build GISO, see [Build Golden ISO, on page 3](#).

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 6](#).

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 6](#)
- [Install Replace with Golden ISO, on page 9](#)

## Limitations

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

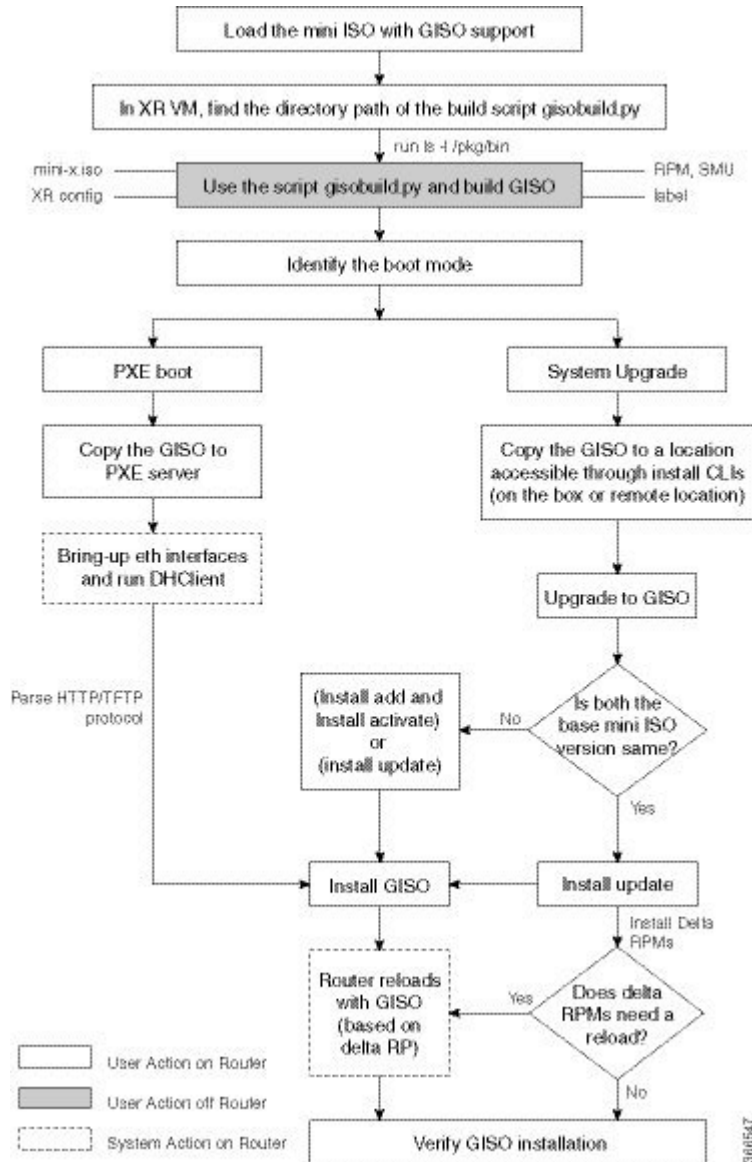
- Building and booting GISO for asynchronous package (a package of different release than the ISO) is not supported.

- 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.

# 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.
- Skips and removes Cisco RPMs that do not match the mini-x.iso version.
- 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.

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** Golden ISO can be built only from mini ISO. The full or fullk9 bundle ISO is not supported.

Use the following naming convention when building GISO:

GISO Build	Format	Example
GISO without k9sec RPM	<code>&lt;platform-name&gt;-golden-x.iso-&lt;version&gt;.&lt;label&gt;</code>	<code>&lt;platform-name&gt;-golden-x64.iso-&lt;version&gt;.v1</code>
	<code>&lt;platform-name&gt;-golden-x-&lt;version&gt;.iso.&lt;label&gt;</code>	<code>&lt;platform-name&gt;-golden-x64-&lt;version&gt;.iso.v1</code>
GISO with k9sec RPM	<code>&lt;platform-name&gt;-goldenk9-x.iso-&lt;version&gt;.&lt;label&gt;</code>	<code>&lt;platform-name&gt;-goldenk9-x64.iso-&lt;version&gt;.v1</code>
	<code>&lt;platform-name&gt;-goldenk9-x-&lt;version&gt;.iso.&lt;label&gt;</code>	<code>&lt;platform-name&gt;-goldenk9-x64-&lt;version&gt;.iso.v1</code>



**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 non-GISO to 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` 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. Ensure that all RPMs and SMUs are present in the same directory. The number of RPMs and SMUs that can be used to build the Golden ISO is 128.

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

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

The following example shows the script output:

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

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

```
Platform: ncs5000 Version: 6.2.2
```

```
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-r622.x86_64.rpm
```

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

Following SYSADMIN x86\_64 rpm(s) will be used for building Golden ISO:

```
(+) ncs5000-sysadmin-system-6.2.2-r622.CSCcv44444.x86_64.rpm
```

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

```
(+) ncs5000-sysadmin-system-6.2.2-r622.CSCcv44444.arm.rpm
```

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

Building Golden ISO...

Summary .....

XR rpms:

```
ncs5000-mgbl-3.0.0.0-r622.x86_64.rpm
```

SYSADMIN rpms:

```
ncs5000-sysadmin-system-6.2.2-r622.CSCcv44444.x86_64.rpm
```

```
ncs5000-sysadmin-system-6.2.2-r622.CSCcv44444.arm.rpm
```

```
...Golden ISO creation SUCCESS.
```

Golden ISO Image Location: <directory-path>/ncs5000-golden-x.iso-6.2.2

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.

GISO is built with the RPMs placed in respective folders in the specified directory and also includes the log files `giso_summary.txt` and `gisobuild.log-<timestamp>`. The XR configuration file is placed as `router.cfg` in the directory.



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

---

**What to do next**

Install the golden ISO 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 instal 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 <absoulte-path-of-Golden-ISO>
```

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 Rpm:

```

xr rpms in golden ISO:
  ncs5000-k9sec-x64-2.2.0.1-r622.CSCxr33333.x86_64.rpm
  openssh-scp-6.6pl.p1-r0.0.CSCTpl2345.xr.x86_64.rpm
  openssh-scp-6.6pl-r0.0.xr.x86_64.rpm
  ncs5000-mpis-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-6.6pl-r0.0.admin.x86_64.rpm
  openssh-scp-6.6pl-r0.0.admin.arm.rpm
  openssh-scp-6.6pl.p1-r0.0.CSCTpl2345.admin.x86_64.rpm
  openssh-scp-6.6pl.p1-r0.0.CSCTpl2345.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

Golden ISO (GISO) upgrades the router to a version that has a predefined list of software maintenance update (SMUs) with a single operation. However, to update to the same version with a different set of SMUs requires a two-step process.

To avoid this two-step process, use the **install replace** command to replace the currently active version with the full package including the image and SMUs from the newly added GISO.

The process involves upgrading the GISO to add the delta SMUs, and manually deactivating the SMUs that are not in use. In addition, this is the only method to upgrade to GISO containing different optional RPMs, which is a subset of the running set of optional RPMs. For example, consider V1 of GISO is the running version with V1 mini and optional RPMs V1 mpls, V1 mpls-te, V1 mgbl, and V1 k9sec. If V2 of GISO does not contain V2 k9sec, then use **install replace** to upgrade to the optional RPMs in V2.



### 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

The **install replace** command is supported only with GISO, but not with .mini and .rpm packages directly.

### 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.