



## 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 7](#).

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

- Migration from IOS XR 32-bit to IOS XR 64-bit
- 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 7](#)

## Limitations

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

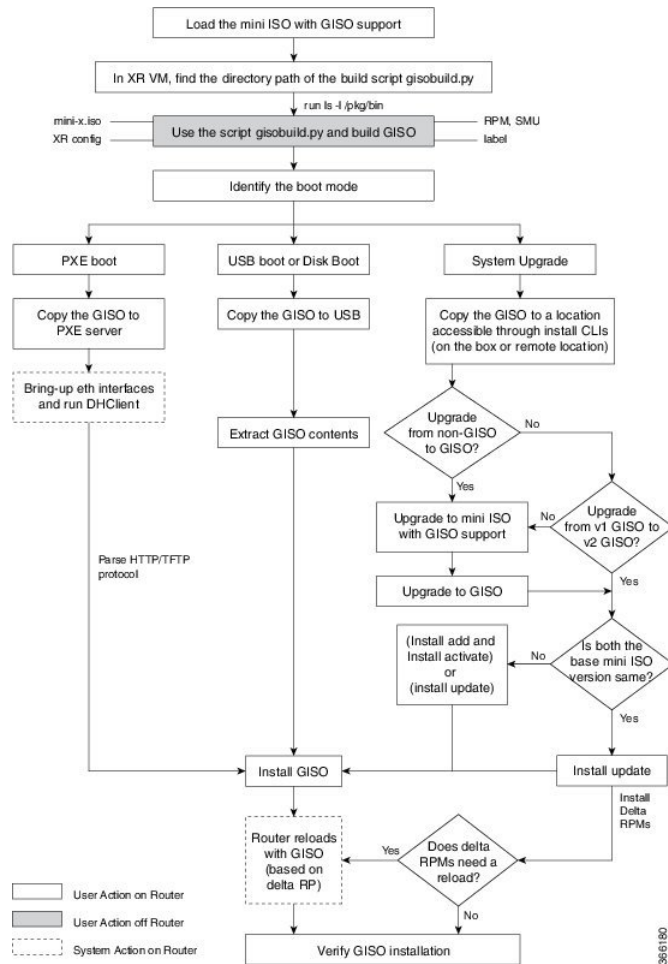
- GISO image size more than 1.8 GB is not supported. The maximum image size for RSP880-LT-SE/TR is 1.599 GB.

- 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.
- Migrating from IOS XR 32-bit to 64-bit OS using GISO involves the following restrictions:
  - The IOS XR 32-bit to 64-bit conversion script does not support file names exceeding 48 characters.
  - The IOS XR 32-bit OS has a maximum file size limit of 2 GB. Ensure that GISO does not exceed that limit.

For more information about migration methods and system requirements, see the [Migration Guide for Cisco ASR 9000 Series Routers](#).

## Golden ISO Workflow

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



# 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 `asr9000-mpls-te-rsvp` is dependent on the parent RPM `asr9000-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 ASR 9000 series routers, upgrade to release 6.1.3 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 asr9k-mini-x.iso] [-r <rpm repository>]
[-c <config-file>] [-l <giso label>] [-m] [-v]
Golden ISO build process starting...
```

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

```
Platform: asr9k Version: <version>
```

```
Info: Migration option is provided so migration tar will be generated
```

```
Scanning repository [repo]...
```

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

```
Total 56 RPM(s) present in the repository path provided in CLI
```

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

```
(+) asr9k-diags-x64-1.0.0.0-r<version>.x86_64.rpm
(+) asr9k-parser-x64-2.0.0.0-r<version>.x86_64.rpm
(+) asr9k-mgbl-x64-3.0.0.0-r<version>.x86_64.rpm
(+) asr9k-k9sec-x64-2.2.0.0-r<version>.x86_64.rpm
(+) asr9k-os-suppl-64-4.0.0.1-r<version>.CSChu77777.x86_64.rpm
(+) asr9k-mpis-x64-2.1.0.0-r<version>.x86_64.rpm
(+) asr9k-k9sec-x64-2.2.0.1-r<version>.CSCxr33333.x86_64.rpm
```

```
-----Truncated
```

```
Skipping following rpms from repository since they are already present in base ISO:
```

```
(-) asr9k-parser-x64-2.0.0.0-r<version>.x86_64.rpm
(-) asr9k-bgp-x64-1.0.0.0-r<version>.x86_64.rpm
(-) asr9k-diags-x64-1.0.0.0-r<version>.x86_64.rpm
```

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

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

```
(+) asr9k-sysadmin-system-<version>-r<version>.CSCcv11111.x86_64.rpm
(+) asr9k-sysadmin-shared-<version>-r<version>.CSCcv22222.x86_64.rpm
(+) asr9k-sysadmin-system-<version>-r<version>.CSCcv44444.x86_64.rpm
```

```
-----Truncated
```

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

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

```
(+) asr9k-sysadmin-hostos-<version>-r<version>.CSChu77777.host.x86_64.rpm
(+) cisco-klm-mifpga-0.1.pl-r0.0.CSCTpl2345.host.x86_64.rpm
(+) kernel-modules-3.14-r0.1.host.x86_64.rpm
```

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

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

```
Summary .....
```

```

XR rpms:
  asr9k-k9sec-x64-2.2.0.0-r<version>.x86_64.rpm
  iosxr-infra-asr9k-64-4.0.0.2-r<version>.CSCxr11111.x86_64.rpm
  asr9k-mlps-te-rsvp-x64-1.2.0.0-r<version>.x86_64.rpm

HOST rpms:
  asr9k-sysadmin-hostos-<version>-r<version>.CSCChu77777.host.x86_64.rpm
  cisco-klm-mifpga-0.1.pl-r0.0.CSCtpl2345.host.x86_64.rpm
  kernel-modules-3.14-r0.1.host.x86_64.rpm

...Golden ISO creation SUCCESS.

Golden ISO Image Location: <repo>/asr9k-goldenk9-x64.iso-<version>.v1

Building Migration tar...

Migration tar creation SUCCESS.

Migration tar Location: <repo>/asr9k-goldenk9-x64-migrate_to_eXR.tar-<version>.v1

Detail logs: <repo>/Giso_build.log-2016-12-18:03:53:54.837680
[03:54:53]-[router:<repo>]$ ls -ltr asr9k-goldenk9-x64*
-rw-r--r-- 1 root root 1274669056 Dec 18 03:54 asr9k-goldenk9-x64.iso-<version>.v1
-rw-r--r-- 1 root root 1425008640 Dec 18 03:54 asr9k-goldenk9-x64-migrate_to_x64.tar-<version>.v1
-----Truncated

```

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.

The following example shows the logs from installation of GISO using PXE boot:

```
...
Fri Dec 2 19:18:03 UTC 2016: ---Starting to prepare host logical volume---
...
Fri Dec 02 19:18:14 UTC 2016: Skipping tp base rpm(openssh-scp-6.6p1-r0.0.host.x86_64.rpm) from
installation
Fri Dec 02 19:18:14 UTC 2016: Skipping tp base rpm(kernel-modules-3.14-r0.1.host.x86_64.rpm) from
installation
Fri Dec 02 19:18:15 UTC 2016: Installing asr9k-sysadmin-hostos-6.1.3-r613.CSChu77777.host.x86_64
[SUCCESS]
...
Fri Dec 2 19:18:23 UTC 2016: ---Starting to prepare calvados logical volume---
...
Fri Dec 02 19:18:48 UTC 2016: Skipping tp base rpm(openssh-scp-6.6p1-r0.0.admin.x86_64.rpm) from
installation
Fri Dec 02 19:18:48 UTC 2016: Skipping tp base rpm(kernel-modules-3.14-r0.1.admin.x86_64.rpm)
from installation
Fri Dec 02 19:18:49 UTC 2016: Installing asr9k-sysadmin-system-6.1.3-r613.CSCcv44444.x86_64
[SUCCESS]
Fri Dec 02 19:18:50 UTC 2016: Installing asr9k-sysadmin-shared-6.1.3-r613.CSCcv33333.x86_64
[SUCCESS]
Fri Dec 02 19:18:51 UTC 2016: Installing asr9k-sysadmin-hostos-6.1.3-r613.CSChu77777.admin.x86_64
[SUCCESS]
...
Fri Dec 2 19:19:07 UTC 2016: ---Starting to prepare repository---
Fri Dec 2 19:19:11 UTC 2016: File system creation on /test took 3 seconds
Fri Dec 2 19:19:11 UTC 2016: Copying /iso/host.iso to repository /iso directory
Fri Dec 2 19:19:11 UTC 2016: Copy Host rpms to repository
Fri Dec 2 19:19:13 UTC 2016: Copying /iso/asr9k-sysadmin.iso to repository /iso directory
Fri Dec 2 19:19:13 UTC 2016: Copy Sysadmin rpms to repository
Fri Dec 2 19:19:16 UTC 2016: Copy HostOs rpms to repository
Fri Dec 2 19:19:16 UTC 2016: Copy XR rpms to repository
```

```
Fri Dec 2 19:19:16 UTC 2016: Copy giso_info.txt to repository
Fri Dec 2 19:19:17 UTC 2016: Copying /iso/asr9k-xr.iso to repository /iso directory
Fri Dec 2 19:19:21 UTC 2016: Copying all ISOs to repository took 10 seconds
```

...

- **USB boot or Disk Boot:** when the USB mode is detected during boot, and GISO is identified, the additional RPMs and XR configuration files are extracted and installed.
- **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.iss0 /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.
- **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
-----
asr9k-sysadmin-6.1.1

asr9k-sysadmin-hostos-6.1.1-r611.CSCcv10001.admin.x86_64
asr9k-sysadmin-system-6.1.1-r611.CSCcv10005.x86_64
....

XR repository
-----
asr9k-iosxr-mgbl-3.0.0.0-r611.x86_64
asr9k-xr-6.1.1
....

Host repository
-----
host-6.1.1

```

**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 asr9k-goldenk9-x64-6.1.3

Sun Dec 4 13:52:48.279 UTC
This may take a while ...
ISO Name: asr9k-goldenk9-x64-6.1.3
ISO Type: bundle
ISO Bundled: asr9k-mini-x64-6.1.3
Golden ISO Label: temp
ISO Contents:
ISO Name: asr9k-xr-6.1.3
ISO Type: xr
rpms in xr ISO:
iosxr-os-asr9k-64-5.0.0.0-r613
iosxr-ce-asr9k-64-3.0.0.0-r613
iosxr-infra-asr9k-64-4.0.0.0-r613
iosxr-fwding-asr9k-64-4.0.0.0-r613
iosxr-routing-asr9k-64-3.1.0.0-r613
...

ISO Name: asr9k-sysadmin-6.1.3

```

```
ISO Type: sysadmin
rpms in sysadmin ISO:
  asr9k-sysadmin-topo-6.1.3-r613
  asr9k-sysadmin-shared-6.1.3-r613
  asr9k-sysadmin-system-6.1.3-r613
  asr9k-sysadmin-hostos-6.1.3-r613.admin
...

ISO Name: host-6.1.3
ISO Type: host
rpms in host ISO:
  asr9k-sysadmin-hostos-6.1.3-r613.host

Golden ISO Rpms:
  xr rpms in golden ISO:
    asr9k-k9sec-x64-2.2.0.1-r613.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
    asr9k-mpls-x64-2.1.0.0-r613.x86_64.rpm
    asr9k-k9sec-x64-2.2.0.0-r613.x86_64.rpm

  sysadmin rpms in golden ISO:
    asr9k-sysadmin-system-6.1.3-r613.CSCcv11111.x86_64.rpm
    openssh-scp-6.6p1-r0.0.admin.x86_64.rpm
    openssh-scp-6.6p1.p1-r0.0.CSCtp12345.admin.x86_64.rpm

  host rpms in golden ISO:
    openssh-scp-6.6p1-r0.0.host.x86_64.rpm
    openssh-scp-6.6p1.p1-r0.0.CSCtp12345.host.x86_64.rpm
```

---

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