Create GISO File for IOS-XR Upgrade with Python 3

Contents

Introduction

This document describes the creation of Golden ISO (GISO) for upgrades in routers that run Cisco IOS® XR software in eXR versions.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco IOS XR software
- Cisco IOS XR software installation and upgrade procedures
- Linux basic commands and navigation of command line

Components Used

This document is not restricted to specific hardware versions, this document apply for all routers running IOS XR 64 Bit.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Related Products

This document can also be used with these hardware versions:

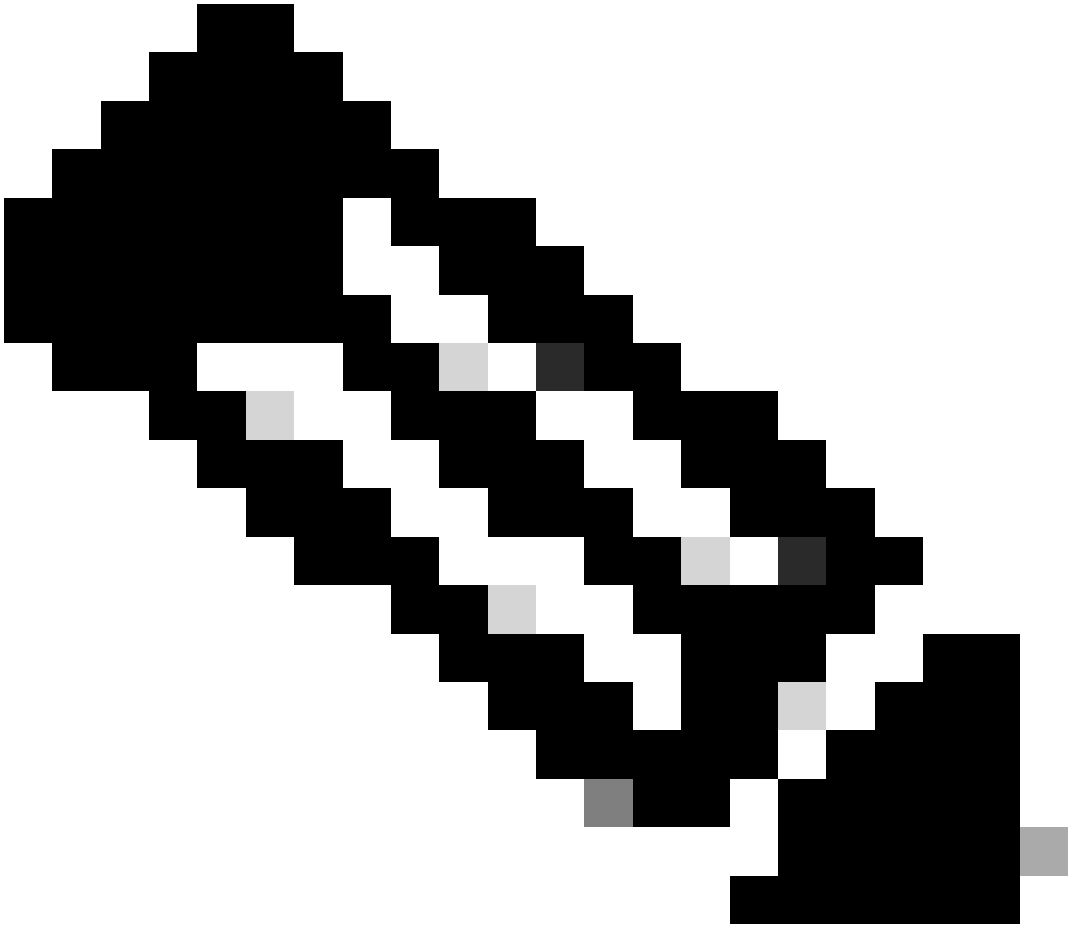
- Cisco 9000 Series Routers
- Cisco 8000 Series Routers

- Cisco NCS 5700 Series Routers
- Cisco NCS 5500 Series Routers
- NCS 540 & 560 Routers

Golden ISO for XR Upgrades

Typically, Cisco releases IOS-XR software as a mini/base ISO which contains mandatory IOS-XR packages for a given platform, a set of optional packages to run additional features and software patches for any bug fixes in form of Software Maintenance Upgrade (SMU). Optional package and SMU are in RPM packaging format.

The Golden ISO tool creates an ISO containing the full contents of the mini/base ISO together with optional packages and SMU of the user choice. Once the Golden ISO is created it can be used either for iPXE boot or used for SU (system upgrade) from the current running version to a new version of IOS-XR.



Note: Mini/base ISO is mandatory for any Golden ISO creation

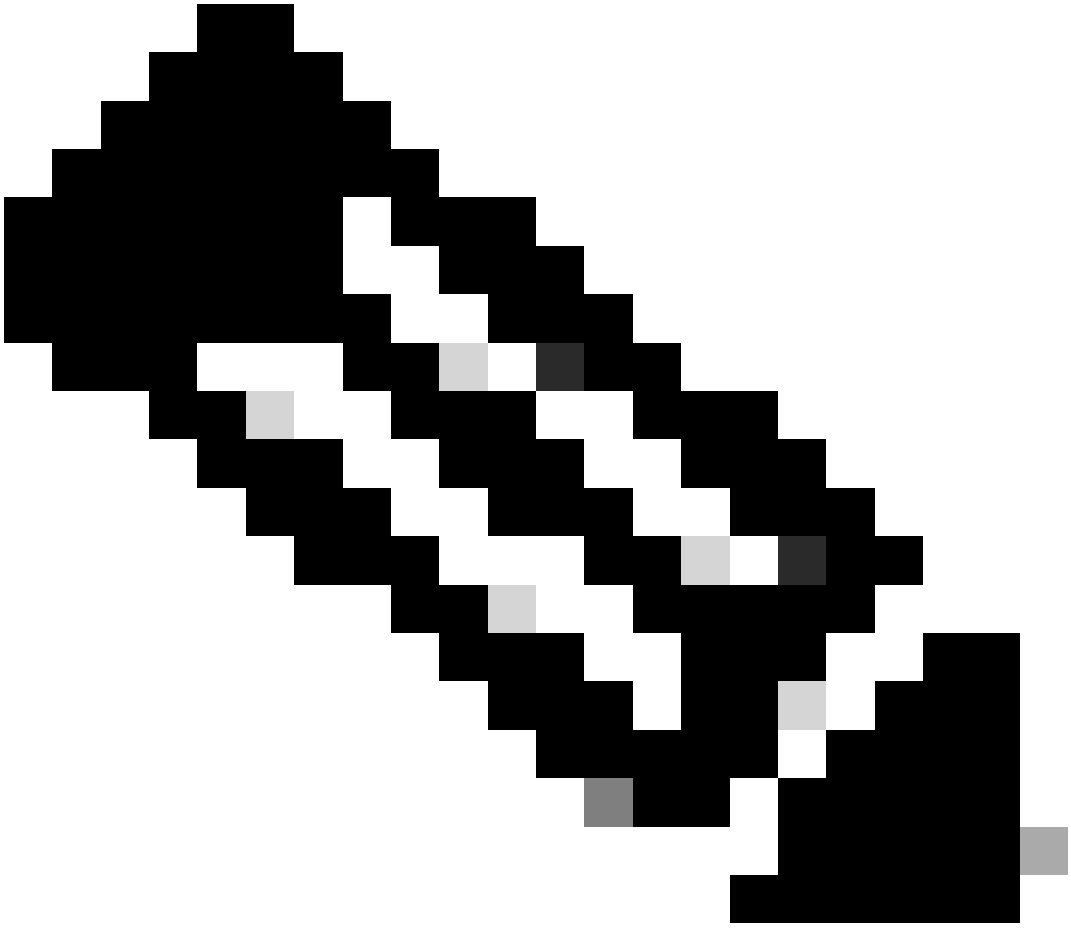
---

gisobuild Python Tool

This tool can be run natively on a Linux host. Alternatively, the tool can also be run on a Linux system with Docker enabled and the ability to pull the published 'cisco-xr-gisobuild' image from Docker Hub, for this example is suggested to work in Debian 11.8

This tool has the next executable requirements:

- python3 >= 3.6
- rpm >= 4.14
- cpio >= 2.10
- gzip >= 1.9
- createrepo_c
- file
- isoinfo
- mkisofs
- mksquashfs
- openssl
- unsquashfs
- 7z (Optional - but functionality can be reduced without)
- iso-read (Optional - but functionality can be reduced without)
- zip (Optional - but functionality can be reduced without)
- unzip (Optional - but functionality can be reduced without)



Note: On a native Linux system, which does not have all dependencies met, the tool dependencies can be installed on supported distributions by running the next command (possibly via sudo): ./setup/prep_dependency.sh

It also requires the next Python (>= 3.6) modules:

- dataclasses
- defusedxml
- distutils
- packaging
- rpm
- yaml

To run natively on a Linux host, the next distributions have been tested, specifically for this scenario, Debian.

- Alma Linux 8
- Fedora 34
- Debian 11.2

Preparation

Is imperative to first confirm what optional packages are needed and why. Installing unnecessary or excessive packages can produce disk space utilization problems and failures during the installation. Confirm each platform pre-requisites and requirements for disk space prior to the creation of the GISO.

To download the necessary software consult the Official Software Download site: Cisco Software Download

Script can help us to unify a great range of files, for example, iso, bridge fixes, SMUs and so on.

Is needed to copy the gisobuild.py script to a particular location on the server. The script is located at gisobuild Git Site

GISO Creation

Here is a summary of arguments that we can use for this script usage:

```
usage: gisobuild.py [-h] [--iso ISO] [--repo REPO [REPO ...]]
                    [--bridging-fixes BRIDGE_FIXES [BRIDGE_FIXES ...]]
                    [--xrconfig XRCONFIG] [--ztp-ini ZTP_INI] [--label LABEL]
                    [--no-label] [--out-directory OUT_DIRECTORY]
                    [--create-checksum] [--yamlfile CLI_YAML] [--clean]
                    [--pkglist PKGLIST [PKGLIST ...]] [--script SCRIPT]
                    [--docker] [--x86-only] [--migration] [--optimize]
                    [--full-iso]
                    [--remove-packages REMOVE_PACKAGES [REMOVE_PACKAGES ...]]
                    [--skip-usb-image] [--copy-dir COPY_DIRECTORY]
                    [--clear-bridging-fixes] [--verbose-dep-check] [--debug]
                    [--isoinfo ISOINFO] [--image-script IMAGE_SCRIPT]
                    [--version]
```

For this example, a GISO for ASR 9901 is created, ISIS and OSPF packages are used for brevity, mini for 7.9.21 version and config file also added to the GISO.

As visible in the next command, packages, mini and config file are copied in the /src directory, this to create a cleared version of the script run command.

```
root@debian:/gisobuild-master/src# ls
asr9k-9000v-nV-x64-1.0.0.0-r7921.x86_64.rpm
asr9k-mcast-x64-1.0.0.0-r7921.x86_64.rpm
asr9k-bng-ipoe-x64-1.0.0.0-r7921.x86_64.rpm
asr9k-mgbl-x64-1.0.0.0-r7921.x86_64.rpm
asr9k-bng-pppoe-x64-1.0.0.0-r7921.x86_64.rpm
asr9k-mini-x64-7.9.21.iso
asr9k-bng-supp-x64-1.0.0.0-r7921.x86_64.rpm
asr9k-mpls-te-rsvp-x64-1.0.0.0-r7921.x86_64.rpm
asr9k-bng-x64-1.0.0.0-r7921.x86_64.rpm
asr9k-mpls-x64-1.0.0.0-r7921.x86_64.rpm
asr9k-optic-x64-1.0.0.0-r7921.x86_64.rpm
asr9k-eigrp-x64-1.0.0.0-r7921.x86_64.rpm
asr9k-ospf-x64-1.0.0.0-r7921.x86_64.rpm
asr9k-isis-x64-1.0.0.0-r7921.x86_64.rpm
```

```
asr9k-services-x64-1.0.0.0-r7921.x86_64.rpm
asr9k-li-x64-1.0.0.0-r7921.x86_64.rpm
asr9k-m2m-x64-1.0.0.0-r7921.x86_64.rpm
lnt
lntmod
output_gisobuild
utils
validate
wrappers
exrmod
running-config-ASR9K
gisobuild.py

--------GISO CREATION--------
root@debian:/gisobuild-master/src# ./gisobuild.py --iso asr9k-mini-x64-7.9.21.iso --pkglist asr9k-isis-
System requirements check [PASS]

Platform: asr9k Version: 7.9.21

XR-Config file (/gisobuild-master/src/running-config-ASR9K) will be encapsulated in Golden ISO.
Warning: No RPMS or Optional Matching 7.9.21 packages found in repository

Building Golden ISO...
Summary .....

XR Config file:
    router.cfg

    ...Golden ISO creation SUCCESS.

Golden ISO Image Location: /gisobuild-master/src/output_gisobuild/asr9k-golden-x64-7.9.21-firstGiso.iso
Creating USB Boot zip...
Skipping USB Boot Zip creation: Not supported for platform: asr9k
USB BOOT ZIP NEEDED?: Contact asr9k team to add support.
root@debian:/gisobuild-master/src#
```

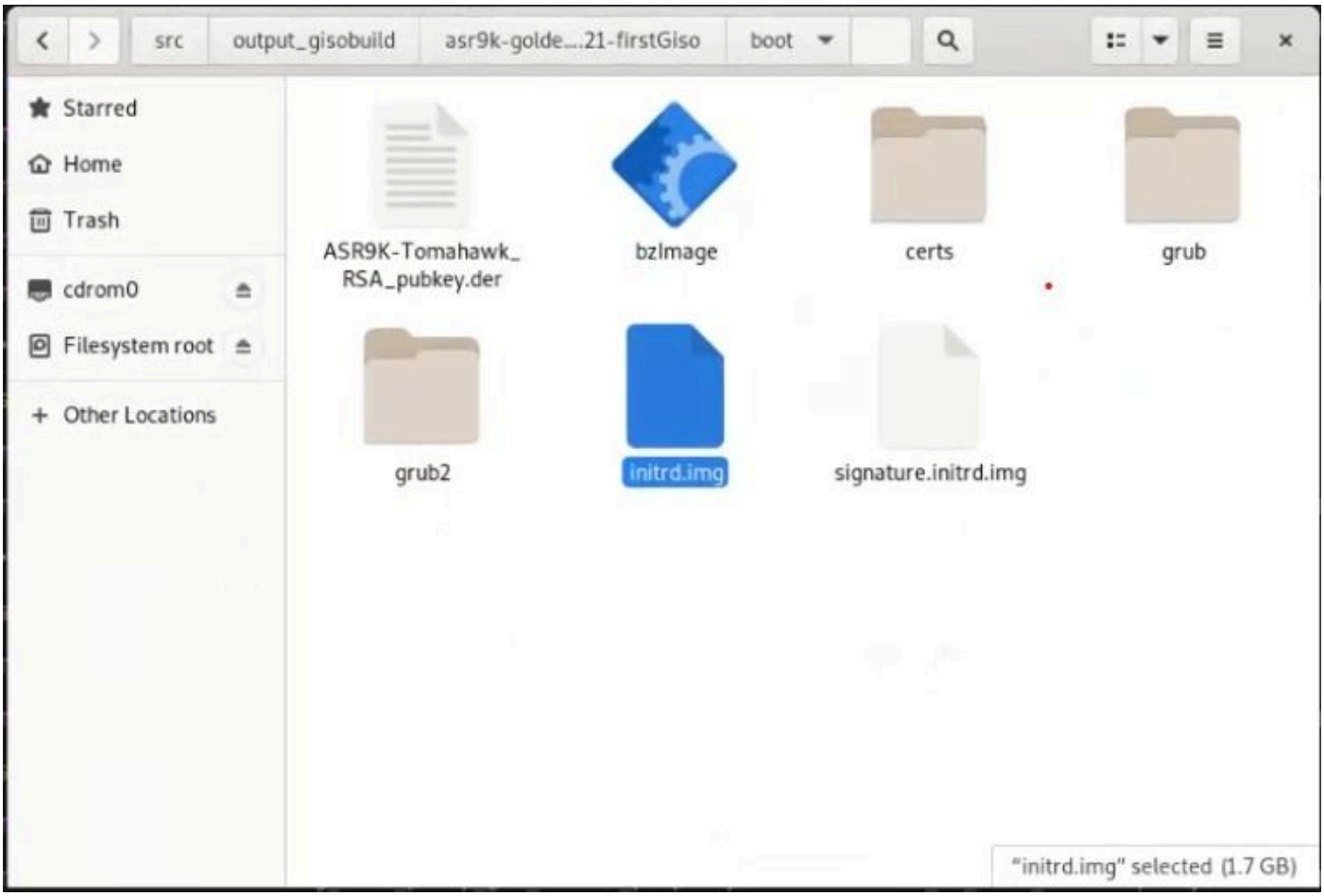
Here is a brief description of the parameters used for this GISO creation:

--iso ISO	Path to Mini.iso/Full.iso file
--xrconfig XRCONFIG	Path to XR config file
--label LABEL, -l LABEL	Golden ISO Label
--clean	Delete output dir before proceeding
--skip-image	Do not build the USB image (not supported for ASR9K platform)

Verify

As the installation logs mention we can confirm our GISO creation by reviewing the location, for this example. Golden ISO Image Location: /gisobuild-master/src/output_gisobuild/asr9k-golden-x64-7.9.21-firstGiso.iso

As per next image, initrd.img is 1.7GB containing the installation packages that we included.



GISO in Debian GUI