

Software Packaging and Architecture

The Cisco ASR 1000 Series Aggregation Services Routers introduce a new software packaging model and architecture.

This chapter discusses this new packaging and architecture and contains the following sections:

- Software Packaging on the Cisco ASR 1000 Series Routers, on page 1
- Image Signing and Bootup, on page 5
- Processes Overview, on page 7

Software Packaging on the Cisco ASR 1000 Series Routers

This section covers the following topics:

ASR 1000 Series Routers Software Overview

The Cisco ASR 1000 Series Routers run using Cisco IOS XE software. Cisco IOS XE is released using consolidated packages and optional subpackages.

Each consolidated package contains a collection of software subpackages. Each software subpackage is an individual software file that controls a different element or elements of the Cisco ASR 1000 Series Router. Each individual software subpackage can be upgraded individually, or all software subpackages for a specific consolidated package can be upgraded as part of a complete consolidated package upgrade. Importantly, IOS (the RPIOS subpackage) is considered one of the seven individual subpackages that makes up a complete consolidated package.

A collection of software subpackages packaged together creates a single consolidated package. A consolidated package allows users to upgrade all individual subpackages on the router with a single software image download. Consolidated packages can be downloaded from Cisco.com; users who want to run the router using individual subpackages that are part of the consolidated package must first download the image from Cisco.com and extract the individual subpackages from the image, which can be done by entering **request platform** command-line interface commands.

Beginning in Cisco IOS XE Release 2.4, support for a supplemental, optional subpackage of type called *sipspawma* is introduced to support the Cisco WebEx Node for ASR 1000 Series shared port adapter (SPA). Optional subpackages are installed similarly to individual subpackages; however, optional subpackages are not bundled as part of a consolidated package like prior support for the individual subpackages, and optional subpackages must be downloaded independently.

Consolidated Packages

A consolidated package is a single image composed of individual software subpackage files. A single consolidated package file is a bootable file, and the Cisco ASR 1000 Series Router can be run using the consolidated package.

Note Consolidated packages only contain the required individual subpackage files. These packages do not contain supplemental, optional subpackages, such as the "sipspawma" package for the Cisco WebEx Node for ASR 1000 Series.

Each consolidated package also contains a provisioning file. A provisioning file is used for booting in cases where the individual subpackages are extracted from the consolidated package, or optional subpackages are used to run the router. For additional information on the advantages and disadvantages of running a complete consolidated package, see the "Running the Cisco ASR 1000 Series Routers: An Overview" section on page 5-1 .

For information about the consolidated packages available in a specific version of Cisco IOS XE, see the release notes for that version of Cisco IOS XE. The Cisco IOS XE Software Release Notes contains the release notes for each version of Cisco IOS XE.

Important Information About Consolidated Packages

The important information about consolidated packages include:

- For each version of a consolidated package, the RPBase, RPControl, ESPBase, SIPSPA, and SIPBase subpackages are identical among consolidated packages.
- For each version of consolidated package, the RPIOS subpackage is always different among consolidated packages.
- A consolidated package file is a bootable file. If the router is configured to run using a the complete consolidated package, boot the router using the consolidated package file. If the router is configured to run using individual subpackages, boot the router using the provisioning file. For additional information on the advantages and disadvantages of running a complete consolidated package, see the "Running the Cisco ASR 1000 Series Routers: An Overview" section on page 5-1 .
- If you need to install optional subpackages, then you must boot the router using the individual subpackage provisioning file method.

Individual Software SubPackages Within a Consolidated Package

This section provides an overview of the Cisco ASR 1000 Series Routers subpackages and the purpose of each individual subpackage. Every consolidated package will have all of these individual subpackages. To see additional information about each individual subpackages in a particular Cisco IOS XE release, see *Cisco IOS XE Release Notes* for that release.

Table 1: Individual SubPackages

SubPackage	Purpose
RPBase	Provides the operating system software for the Route Processor.

SubPackage	Purpose
RPControl	Controls the control plane processes that interface between the IOS process and the rest of the platform.
RPAccess	Exports processing of restricted components, such as Secure Socket Layer (SSL), Secure Shell (SSH), and other security features.
RPIOS	Provides the Cisco IOS kernel, which is where IOS features are stored and run. Each consolidated package has a different RPIOS.
ESPBase	Provides the ESP operating system and control processes, and the ESP software.
SIPBase	Controls the SIP operating system and control processes.
SIPSPA	Provides the SPA driver and Field Programmable Device (FPD) images.

Important Notes About Individual SubPackages

The important information about individual subpackage include:

- Individual subpackages cannot be downloaded from Cisco.com individually. To get these individual subpackages, users must download a consolidated package and then extract the individual subpackages from the consolidated package using the command-line interface.
- If the router is being run using individual subpackages instead of being run using a complete consolidated package, the router must be booted using a provisioning file. A provisioning file is included in all consolidated packages and is extracted from the image along with the individual subpackages whenever individual subpackages are extracted.

Optional Software SubPackages Outside of Consolidated Packages

Beginning in Cisco IOS XE Release 2.4, the ASR 1000 Series Routers support a new type of subpackage—this is an optional software subpackage that is available as a separate, external package that is downloaded and installed along with the other required subpackages.

sipspawmak9 is an optional subpackage that provides the system software for the Cisco WebEx Node for ASR 1000 Series Routers.

Important Notes About Optional SubPackages

The important information about optional subpackages include:

- Optional subpackages are downloaded separately from consolidated package files. Optional subpackages are not contained within a consolidated package for a release.
- Optional package installation works similarly to the installation of individual subpackages using a provisioning file.
- Optional subpackages can be uninstalled to remove provisioning when the package no longer applies to an RP.
- Optional subpackages are easily supported by the standard ISSU upgrade process as long as the package is located in the directory of the provisioning file for each RP.

Provisioning Files



Note You must use the provisioning files to manage the boot process if you need to install optional subpackages.

Provisioning files manage the boot process when the Cisco ASR 1000 Series Router is configured to run using individual subpackages or optional subpackages (such as the package for the Cisco WebEx Node for ASR 1000 Series). When individual subpackages are being used to run the Cisco ASR 1000 Series Router, the router has to be configured to boot the provisioning file. The provisioning file manages the bootup of each individual subpackage and the Cisco ASR 1000 Series Router assumes normal operation.

Provisioning files are extracted automatically when individual subpackage files are extracted from a consolidated package.

Provisioning files are not necessary for running the router using the complete consolidated package; if you want to run the router using the complete consolidated package, simply boot the router using the consolidated package file.

See the "Running the Cisco ASR 1000 Series Routers: An Overview" section on page 5-1 for additional information on the advantages and disadvantages of running individual subpackages versus running a complete consolidated package.

Important Notes About Provisioning Files

The important information about provisioning files include:

- Each consolidated package contains two provisioning files. One of the provisioning files is always named "packages.conf", while the other provisioning file will have a name based on the consolidated package naming structure. In any consolidated package, both provisioning files perform the exact same function.
- In most cases, the "packages.conf" provisioning file should be used to boot the router. Configuring the router to boot using this file is generally easier because the router can be configured to boot using "packages.conf", so no changes have to be made to the boot statement when Cisco IOS XE is upgraded (the **boot system** *file-system*:**packages.conf** configuration command can remain unmodified before and after an upgrade).
- The provisioning file and individual subpackage files must be kept in the same directory. The provisioning file does not work properly if the individual subpackage files are in other directories.
- The provisioning filename can be renamed; the individual subpackage filenames cannot be renamed.
- After placing the provisioning file and the individual subpackage files in a directory and booting the router, it is highly advisable not to rename, delete, or alter any of these files. Renaming, deleting, or altering the files can lead to unpredictable router problems and behaviors.

ROMmon Image

An independent ROMmon image is released periodically separate from consolidated packages or any other software releases.

See the documentation that accompanies the ROMmon image for information on each ROMmon image. For additional information on ROMmon, see the *Cisco ASR 1000 Series Routers Maintain and Operate Guide*

File to Upgrade Field Programmable Hardware Devices

Starting in Cisco IOS XE Release 3.1.0S, a hardware programmable package file used to upgrade field programmable hardware devices is released as needed . A package file is provided for the field programmable device to customers in cases where a field upgrade is required. If the Cisco ASR 1000 Series Router contains an incompatible version of the hardware programmable firmware on the Cisco ASR1000-RP, Cisco ASR1000-SIP, or Cisco ASR1000-ESP, then that firmware may need to be upgraded.

Generally an upgrade is only necessary in cases where a system message indicates one of the field programmable devices on the Cisco ASR 1000 Series Router needs an upgrade or a Cisco technical support representative suggests an upgrade.

In Cisco IOS XE Release 3.1.0S, a package file that contains a new version of the Complex Programmable Logic Device (CPLD) code is available for users who need to upgrade old versions of firmware on a Cisco ASR1000-RP2 or Cisco ASR1000-SIP10 in a Cisco ASR 1013 Router.

For more information on upgrading field programmable hardware devices, see the *Upgrading Field Programmable Hardware Devices for Cisco ASR 1000 Series Routers* document.

Image Signing and Bootup

The Cisco build servers generate the Cisco IOS XE images. The Cisco IOS XE images use the Abraxas image signing system to sign these images securely with the Cisco private RSA keys.

When you copy the Cisco IOS XE image onto a Cisco ASR 1000 Series Router, the Cisco's ROMMON Boot ROM verifies the image using Cisco release keys. These keys are public keys that correspond to the Cisco release private key that is stored securely on the Abraxas servers. The release key is stored in the ROMMON.

All the new Cisco ASR 1000 Series platforms support Cisco's Secure Boot technology. The Cisco Secure Boot technology serves as a hardware trust anchor which validates the ROMMON software to ensure that the ROMMON software is not tampered with.

The Cisco IOS XE image is digitally signed during the build time. An SHA-512 hash is generated over the entire binary image file, and then the hash is encrypted with a Cisco RSA 2048-bit private key. The ROMMON verifies the signature using the Cisco public key. If the software is not generated by a Cisco build system, the signature verification fails. The Cisco ASR 1000 Series ROMMON rejects the image and stops booting. If the signature verification is successfully, the Cisco ASR 1000 Series Router boots the image to the Cisco IOS XE runtime environment.

The ROMMON follows these steps when it verifies a signed Cisco IOS XE image during the boot up process:

1. Loads the Cisco IOS XE image into the CPU memory.
2. Examines the Cisco IOS XE package header.
3. Runs a non-secure integrity check on the image to ensure that there is no unintentional file corruption from the disk or TFTP. This is performed using a non-secure SHA-1 Hash.
4. Copies the Cisco's RSA 2048-bit public release key from the ROMMON storage and validates that the Cisco's RSA 2048-bit public release key is not tampered.
5. Extracts the Code Signing signature (SHA-512 Hash) from the package header and verifies it using Cisco's RSA 2048-bit public release key.
6. Performs the Code Signing validation by calculating the SHA-512 hash of the Cisco IOS XE package and compares it with the Code Signing signature. The Signed package is now validated.

7. Examines the Cisco IOS XE package header to validate the platform type and CPU architecture for compatibility.
8. Extracts the Cisco IOS XE software from the Cisco IOS XE package and boots it.

Note In above process, the step 3 is a non-secure check of the image which is intended to confirm the image against inadvertent corruption due to disk errors, file transfer errors, or copying errors. This is not part of the image code signing. This check is not intended to detect deliberate image tampering.

Image Code Signing validation occurs in steps 4, 5, and 6. This is a secure code signing check of the image using an SHA-512 Hash that is encrypted with a 2048-bit RSA key. This check is intended to detect deliberate image tampering.

During this process, the device displays the following:

```
  Initializing Hardware ...
System integrity status: 90170400 12030107

System Bootstrap, Version 16.12(8r), RELEASE SOFTWARE
Copyright (c) 1994-2020  by cisco Systems, Inc.

Current image running: Boot ROM0
Last reset cause: LocalSoft

ASR1001-HX platform with 8388608 Kbytes of main memory

File size is 0x32e9b97c
Located asr1000-universalk9.17.01.01.SPA.bin
Image size 854178172 inode num 34, bks cnt 208540 blk size 8*512
################################################################################
################################################################################
################################################################################  <---- (*) STEP
1
Boot image size = 854178172 (0x32e9b97c) bytes                                  <---- (*) STEP
1

ROM:RSA Self Test Passed
ROM:Sha512 Self Test Passed

Package header rev 1 structure detected                                         <---- (*) STEP
2
Calculating SHA-1 hash...done                                                   <---- (*) STEP
3
validate_package_cs: SHA-1 hash:                                                <---- (*) STEP
3
 calculated 3971e404:1211e83e:87ecc2bb:4f80bd9b:bacad0d7            <---- (*) STEP 3
 expected   3971e404:1211e83e:87ecc2bb:4f80bd9b:bacad0d7            <---- (*) STEP 3
Validating main package signatures                                              <---- (*) STEP
4 & 5

RSA Signed RELEASE Image Signature Verification Successful.                     <---- (*) STEP
6
Image validated                                                                 <---- (*) STEP
7


                Restricted Rights Legend                                         <---- (*) STEP
 8
```

```
Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

           Cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706



Cisco IOS Software [Amsterdam], ASR1000 Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Version
 17.1.1, RELEASE SOFTWARE (fc3)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2019 by Cisco Systems, Inc.
Compiled Fri 22-Nov-19 03:43 by mcpre
```

Processes Overview

Cisco IOS XE has numerous components that run entirely as separate processes on the Cisco ASR 1000 Series Routers. This modular architecture increases network resiliency by distributing operating responsibility among separate processes rather than relying on Cisco IOS software for all operations.

This section covers the following topics:

The following table provides a list of the important individual processes for the Cisco ASR 1000 Series Routers. These processes run in the background, and the CLI on Cisco ASR 1000 Series Routers using Cisco IOS-XE is identical in look, feel, and usage to the Cisco IOS CLI on most platforms. This information is provided because it may be useful for checking router state and troubleshooting, but understanding this information is not essential to understanding most router operations.

Table 2: Individual Processes

Process	Purpose	Affected FRUs	SubPackage Mapping
Chassis Manager	Responsible for all chassis management functions, including management of the HA state, environmental monitoring, and FRU state control.	RP (one instance per RP)SIP (one instance per SIP)ESP (one instance per ESP)	RPControlSIPBaseESPBase
Host Manager	Provides an interface between the IOS process and many of the information-gathering functions of the underlying platform kernel and operating system.	RP (one instance per RP)SIP (one instance per SIP)ESP (one instance per ESP)	RPControlSIPBaseESPBase
Logger	Provides IOS facing logging services to processes running on each FRU.	RP (one instance per RP)SIP (one instance per SIP)ESP (one instance per ESP)	RPControlSIPBaseESPBase

Process	Purpose	Affected FRUs	SubPackage Mapping
Interface Manager	Provides an interface between the IOS process and the per-SPA interface processes on the SIP.	RP (one instance per RP)SIP (one instance per SIP)	RPControlSIPBase
IOS	The IOS process implements all forwarding and routing features for the router.	RP (one per software redundancy instance per RP). Maximum of two instances per RP.	RPIOS
Forwarding Manager	Manages the downloading of configuration to each of the ESPs and the communication of forwarding plane information, such as statistics, to the IOS process.	RP (one per software redundancy instance per RP). Maximum of two instances per RP.ESP (one per ESP)	RPControl ESPBase
Pluggable Services	The integration point between platform policy application, such as authentication and the IOS process.	RP (one per software redundancy instance per RP). Maximum of two instances per RP.	RPControl
Shell Manager	Provides all user interface features and handling related to features in the nonIOS image of the consolidated package.	RP (one instance per RP)	RPControl
SPA driver process	Provides an isolated process driver for a specific SPA.	SPA (one instance per SPA per SIP)	SIPSPA
CPP driver process	Manages the CPP hardware forwarding engine on the ESP.	ESP (one instance per ESP)	ESPBase
CPP HA process	Manages HA state for the CPP hardware forwarding engine.	ESP (one instance per ESP)	ESPBase
CPP SP process	Performs high-latency tasks for the CPP-facing functionality in the ESP instance of the Forwarding Manager process.	ESP (one instance per ESP)	ESPBase

IOS as a Process

In almost all previous Cisco router platforms, an overwhelming majority of the internal software processes are run using Cisco IOS memory.

The Cisco ASR 1000 Series Routers introduce a distributed software architecture that moves many operating system responsibilities out of the IOS process. In this architecture, IOS, which previously was responsible for almost all of the internal software processes, now runs as one of many Linux processes while allowing other Linux processes to share responsibility for running the router. This architecture allows for better allocation of memory so the router can run more efficiently.

Dual IOS Processes

The Cisco ASR 1000 Series Router introduces a dual IOS process model that allows for increased high availability at all times.

Using SSO or RPR, a second IOS process can be enabled on a Cisco ASR 1002 or 1004 Router. On Cisco ASR 1000 Series Routers configured with dual Route Processors, the second IOS process runs on the standby Route Processor.

The state of these dual IOS processes can be checked by entering the **show platform** command.

The advantages of a second IOS process includes:

- Increased fault tolerance—In the event of an active IOS failure, the second IOS process immediately becomes the active IOS process with little to no service disruption.
- No downtime software upgrades—IOS and other software on the router can be upgraded using the In Service Software Upgrade (ISSU) feature in the standby IOS process, thereby allowing the network to remain active during the software upgrade. See the "Router#" section on page 5-20 for additional information on when ISSU can and cannot be used to perform no downtime software upgrades.

File Systems on the Cisco ASR 1000 Series Router

The following table provides a list of file systems that can be seen on the Cisco ASR 1000 Series Routers.

Table 3: File Systems

File System	Description
bootflash:	The boot flash memory file system on the active RP.
cns:	The Cisco Networking Services file directory.
harddisk:	The hard disk file system on the active RP. The harddisk: file system is not available on the Cisco ASR 1002 Routers.
nvram:	Router NVRAM. You can copy the startup configuration to NVRAM or from NVRAM.
obfl:	The file system for Onboard Failure Logging files.
stby-bootflash:	The boot flash memory file system on the standby RP.
stby-harddisk:	The hard disk file system on the standby RP. The harddisk: file system is not available on the Cisco ASR 1002 Routers.
stby-usb[0-1]:	The Universal Serial Bus (USB) flash drive file systems on the standby RP. The stby-usb: file system is not available on the Cisco ASR 1002 Routers.
system:	The system memory file system, which includes the running configuration.
tar:	The archive file system.
tmpsys:	The temporary system files file system.

File System	Description
usb[0-1]:	The Universal Serial Bus (USB) flash drive file systems on the active RP. Only usb0: is available on the Cisco ASR 1002 Router.

If you run into a file system not listed in the above table, enter the**?** help option or see the **copy** command reference for additional information on that file system.

Autogenerated File Directories and Files

This section discusses the autogenerated files and directories that might appear on your Cisco ASR 1000 Series Routers, and how the files in these directories can be managed.

The following table provides a list and descriptions of autogenerated files on the Cisco ASR 1000 Series Routers.

Table 4: Autogenerated Files

File or Directory	Description
crashinfo files	A crashinfo file may appear in the bootflash: or harddisk: file system. These files provide descriptive information of a crash and may be useful for tuning or troubleshooting purposes, but the files are not part of router operations and can be erased without impacting the functioning of the router.
core directory	The storage area for.core files. If this directory is erased, it will automatically regenerate itself at bootup. The .core files in this directory can be erased without impacting any router functionality, but the directory itself should not be erased.
lost+found directory	This directory is created on bootup if a system check is performed. Its appearance is completely normal and does not indicate any issues with the router.
tracelogs directory	The storage area for trace files. Trace files are useful for troubleshooting. Trace files, however, are not part of router operations and can be erased without impacting the router’s performance.

Important Notes About Autogenerated Directories

The important information about autogenerated directories include:

- Any autogenerated file on the bootflash: directory should not be deleted, renamed, moved, or altered in any way unless directed by customer support. Altering autogenerating files on the bootflash: can have unpredictable consequences for system performance.
- Crashinfo, core, and trace files can be deleted, but the core and tracelog directories that are automatically part of the harddisk: file system should not be deleted.