Cisco IOS Software Modularity Installation and Configuration Guide, Cisco IOS Release 12.2SX

C O N T E N T S

Last Updated: July 19, 2011

and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Installing and Configuring Cisco IOS Software Modularity

Cisco IOS Software Modularity that runs on the renewed infrastructure microkernel and new Cisco IOS processes that are modified to make use of the new microkernel constitute enhancements to the Cisco IOS infrastructure. These enhancements increase system availability through fault containment, process restartability, event management, and modular software delivery. Cisco IOS Software Modularity is also referred to as Software Modularity, and the shorter form will be used, where appropriate, in this module.

This module describes the installation and basic configuration of Software Modularity images.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Installing and Configuring Cisco IOS Software Modularity

- Effective in releases following Cisco IOS 12.2(33)SXI3, Software Modularity Installer and patching are no longer supported. If the image is run in the installed mode, warning messages are displayed at startup immediately after the image is decompressed and at the very end of a **show version** command.

Effective in releases following Cisco IOS 12.2(33)SXI3, the **install** commands are no longer supported.

- Effective in releases following Cisco IOS 12.2(33)SXI3, run the image using the normal Cisco IOS load and boot process.
- The Software Modularity Installer manages all file copying, moving, and deletion in the system directory. Do not manipulate any files in the installed software directory that is specified when you install the software. You may manipulate files in other directories.
- If you are running an installed image, you must leave the flash card in the router. Do not remove the flash card while the router is running.
- When adding patches or maintenance packs, be aware that the patch functionality is available only when the router is running installed code where you have performed the install process and the bind process and you have reloaded a base image.
- In Modular IOS, you cannot restart a process on the standby router. The standby router console is disabled by default. If you enable the standby router console and then enter the **process restart** command to restart a process, the standby console will reload and display one of the following error messages:

```
Standby process exited, rebooting.
```

or

```
This process is not known to sysmgr.
```

- Your system must be running a noninstalled Software Modularity image to install a base Software Modularity image. The **install file** command is available only in Software Modularity images. For initial installation, the Software Modularity image is typically booted and run in a noninstalled mode, just as any other Cisco IOS image would be loaded and booted. After this has been done, the **install file** command can be used to fully install the image on the file system.
- Prior to Cisco IOS Release 12.2(33)SXH, Software Modularity supported directory operations, such as the creation and deletion of directories, on bootflash: and slot0: flash file systems. This was done initially to allow the installer in Software Modularity to use the flash file systems if needed. But, because Software Modularity images are too large for linear flash and the Software Modularity Installer works on compact flash, these directory commands are no longer supported. This change takes effect in Cisco IOS Release 12.2(33)SXH.

Information About Installing and Configuring Cisco IOS Software Modularity

Cisco IOS Software Modularity Processes

Software Modularity images contain different types of processes: Posix processes and Cisco IOS style processes.

A Posix process is a collection of code and data that resides in a single address space. Posix processes contain one or more threads of execution named Posix pthreads. A Posix pthread cannot access data outside the address space of the process (except when shared-memory application programming interfaces [APIs] are used). Residing in an individual address space, a Posix process cannot corrupt the data of another Posix process. Each Posix pthread has its own stack but shares all the process code and data.

Cisco IOS style processes contain code and data with one sequence of execution (thread) and one stack. The thread and the stack of a Cisco IOS style process are contained within one address space. The entity commonly known in Cisco IOS software as a process has been renamed as a task in Cisco IOS Software Modularity. Related tasks have been grouped in separate Cisco IOS style processes to achieve modularity.

Cisco IOS Software Modularity Installer

Software Modularity introduces the concept of installed software, which is different from just booting an image on the networking device. Software Modularity images can be saved into the flash file system and booted like a Cisco IOS image, but this is referred to as uninstalled software. To gain the benefits of the Software Modularity Installer and permit patch files to be installed, use the **install file** command to write the software to flash. Installation and activation are now separate processes. The **install bind** command is used to bind Software Modularity base images system-wide. The **install activate** command must be entered to activate a patch. Some patches require a reload to be performed, and a message appears on the console after the **install activate** command has been entered to note the current state of the patch. The figure below shows a flowchart of the install activation and rollback processes.

Figure 1
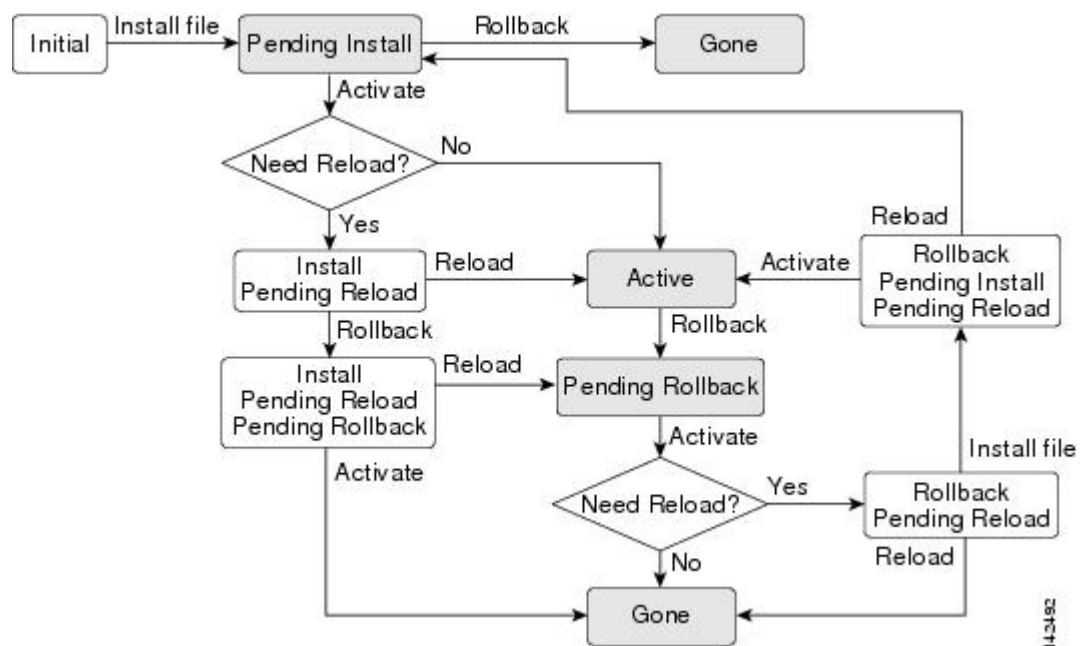
The table below shows whether the patch code is running in the various patch states. The table starts from the Pending Install state as shown at the top of the figure above.

Table 1   Patch State Descriptions

State	State Description	Is Patch Code Running?
PendInst	Pending installation activation	No processes are running the patch code.
InstPRel	Installation activation pending reload	No processes are running the patch code until a card reload is performed.
IPRPndRo	Installation activation pending reload pending rollback	No processes are running the patch code until a card reload is performed.
PendRoll	Pending rollback	Some processes are running the patch code.
RollPRel	Rollback pending reload	Some processes are running the patch code.
RPRPndIn	Rollback pending reload pending installation activation	Some processes are running the patch code.
Active	Patch is active	Some processes are running the patch code.
Pruned	Patch is removed	No processes are running the patch code.

The Software Modularity Installer provides the ability to install, track, and manage system software. Cisco IOS Software Modularity system software includes executables, patches, shared objects, data files, and scripts. Installation of patch files--created to fix bugs or security issues--does not always require the system to be rebooted. Installable entities are checked by the Software Modularity Installer for compatibility with the currently installed system before being installed.

Note   The Software Modularity Installer manages all file copying, moving, and deletion in the system directory. Do not manipulate any files in the installed software directory that is specified when you install the software. You may manipulate files in other directories.

Cisco IOS Software Modularity Rollback Using Tags

Similar to the idea of a database rollback, Software Modularity can roll back to a set of installed files defined by a tag. The installed system is captured at a point in time by defining a tag. If a subsequent installation of a patch file adversely affects the installed system, a rollback can be performed using the defined tag. All installation actions performed since the tag was defined are deleted, and the processes affected by the rollback of installed software are restarted. After the restart, these processes use the software that was present at the time the tag was created. Tags can be deleted, and the system will remove all installation files, which will now never be used because the tag has been removed.

Cisco IOS Software Modularity Patching

When an installed Software Modularity image is running, you can add to or update portions of the software by installing a patch file. When adding patch images, be aware that the patch functionality is available only when the router is running installed code where you have performed the install process, the bind process, and reloaded a base image. Patching involves the replacement of one or more Software Modularity subsystems with an updated or corrected version. Adding a patch can usually be done with minimal impact on the operation of the system. Patching allows the delivery of specific bug fixes instead of an entire new image with many bug fixes. Fixing only specific issues allows faster deployment and minimizes the chance of unrelated bug fixes affecting other features. The Software Modularity rollback facility ensures that a patch can be removed and the system restored to a known state. When some of the Embedded Event Manager features are used, the rollback can occur without manual intervention. For more details about using Embedded Event Manager, see the "Embedded Event Manager Overview" module.

Patches are bundled into maintenance packs that may contain a collection of patches, including a specific fix plus any other dependent patches. The Software Modularity Installer verifies that the patch is compatible with the currently installed software before installing the patch. During the installation of a patch, Software Modularity can determine which subsystems are affected by the patch. Depending on the state of the patch and the relevant conditions when it is installed and activated, some processes that use the subsystems may be restarted.

Information about patches is maintained in the Patch Navigator system, which performs a similar function as the Download Software Area tool. To access Patch Navigator, go to http://www.cisco.com/go/pn . You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

There are three ways to search for patches using the Patch Navigator tool:

- Search by Software Modularity image or DDTS ID
- Search by platform and base release
- Search by patch ID

Each search displays a list of patches with a headline and DDTS ID for each patch. To search for more information on the specified patch, click the patch ID link. If you are searching by patch ID, detailed patch information is displayed. To download a patch, click the download patch link displayed on the listing or detail screens. The download link uses the CCO Software Center, where your username is authenticated and you are asked to provide CCO account information before being allowed to download the patch.

Cisco IOS Software Modularity Installation Repackage Creation

To allow for easier deployment of a base image and several patches to multiple routers, an installable bundled image, referred to as a repackage, can be created. While the image repackage is being created, the Software Modularity Installer saves everything in the installed state including rollback tags. An initial boot must be performed on the device on which the repackage image is to be installed. The ability to create a repackage allows standard installations to be performed across the network and saves installation time.

Cisco IOS Software Modularity Restartability

In the images that do not contain Software Modularity, if a Cisco IOS process fails, the entire system fails. Cisco IOS Software Modularity provides fault containment by isolating groups of functionality into processes. Each process runs in its own address space. A fault that causes a crash in one process will not have an adverse effect on other processes. A process can crash without causing the networking device to crash. The process will be restarted after it has crashed, and the process will return to performing its

intended function. The particular services offered by the process that crashes may be interrupted during the process restart, but other services in the networking device should not be affected.

How to Install and Configure Cisco IOS Software Modularity

Installing Cisco IOS Software Modularity Base Images on a Single RP

Perform this task to install a Software Modularity base image and save the configuration to the running configuration file. Step 5 is an optional step included to allow you to remove all existing software bindings before you bind the software. Remember that installing a base Software Modularity image is different from copying the base image to the disk and performing a reload. The install process allows access to the patching functionality.

The base image may be a repackaged image (see the Creating a Repackage of a Cisco IOS Software Modularity Installation, page 44 task).

Depending on the feature set that you want to run, you need a minimum of 256 MB of compact flash memory and we recommend that you have 512 MB of compact flash memory. If you are installing the 512 MB compact flash memory, you must reformat the flash disk before starting a Software Modularity base image installation.

Your system must be running a noninstalled Software Modularity image to perform this task because the **install file** command is available only in Software Modularity images. For initial installation, the Software Modularity image is typically booted and run in a noninstalled mode, just as any other Cisco IOS image would be loaded and booted. After this has been done, the **install file** command can be used to fully install the image on the file system.

To boot a Software Modularity image, follow the same procedure as when booting a Cisco IOS image.

For more information about booting Cisco IOS images, see the " Loading and Managing System Images " section of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Note In this task you remove all the existing **boot system** commands before entering a new **boot system**command for the new installed image. We recommend that you run the **show startup-config** command and note all your existing **boot system** commands to determine which of them must be reentered and in which order.

SUMMARY STEPS

1. **enable**
2. **install file** *source-file-url destination-directory [second-destination-directory]* [**interactive**]
3. **show install** *search-root-directory* [**detailed**| **pending**]
4. **configure terminal**
5. **no boot system** [*file-url* | *filename*]
6. **install bind** *search-root-directory* [**prepend**]
7. **boot system** {*file-url* | *filename*}
8. Repeat Step 7 for each **boot system** command to be added to the configuration file.
9. **exit**
10. **copy running-config startup-config**
11. **reload**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**install file** *source-file-url destination-directory [second-destination-directory]* [**interactive**] **Example:** Router# install file rcp://s72033/base/ s72033-adventerprisek9_wan_dbg-vz disk0:/sys	Installs a Software Modularity base image file from the specified path for the source into the specified local directory for the destination. • The optional *second-destination-directory* argument is the path of a secondary destination directory in which the installable file is to be installed. • The optional **interactive** keyword displays more detailed output. **Note** This example shows how to use the remote copy protocol (rcp) to source the file, but any URL that can be used as the source of the **copy** command can be used as the source of the **install file** command.
Step 3	**show install** *search-root-directory* [**detailed**| **pending**] **Example:** Router# show install disk0:/sys	(Optional) Displays information about the installed software. • The *search-root-directory* argument displays information about the software installed at the specified location. • The optional **detailed** keyword displays more detailed information about the installed software. • The optional **pending keyword** displays patch upgrade summary information.

	Command or Action	Purpose
Step 4	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 5	**no boot system** [*file-url* | *filename*] **Example:** Router(config)# no boot system	(Optional) Removes **boot system** commands from the startup configuration file. • Use the **no boot system** command without any arguments to remove all **boot system** commands from the startup configuration file.
Step 6	**install bind** *search-root-directory* [**prepend**] **Example:** Router(config)# install bind disk0:/sys	Binds the software by generating a **boot system** command in the configuration file that defines a location from which the software will run. • Remember that **boot system** commands in the startup configuration file are executed in the order in which they were configured. • Use the optional **prepend** keyword to move the latest **boot system** statement to the top of the boot variable, which makes that statement the primary image to boot. • If you know the complete directory path and image name, you can use the **boot system** command instead of the **install bind** command. For more details, see the Binding Cisco IOS Software Modularity, page 37 task.
Step 7	**boot system** {*file-url* | *filename*} **Example:** Router(config)# boot system disk0:/sys/ s72033-entservicesk9_wan-mz	Adds a **boot system** command to the configuration file. • Use the *file-url* or *filename* argument to specify the directory path and image name. • Use this command to provide a **boot system** command for a backup image. **Note** Only one form of the **boot system** command syntax is shown. For more details, see the Cisco IOS Configuration Fundamentals Command Reference.
Step 8	Repeat Step 7 for each **boot system** command to be added to the configuration file.	--
Step 9	**exit** **Example:** Router(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 10	**copy running-config startup-config** **Example:** Router# copy running-config startup-config	Copies the running configuration file to the startup configuration file.
Step 11	**reload** **Example:** Router# reload	(Optional) Reloads the operating system. • Perform this step when you are ready to run the base image that was installed in this task. After the reload, the base image becomes an installed image on which patch files can be activated.

Examples

The following partial sample output from the **show install** privileged EXEC command shows the output for the base file s72033-adventerprisek9_wan_dbg-vm after the **install file** command has been performed. The state of PendInst means that the file is set to be made available to run on the system after the next activation.

```
Router# show install disk0:/sys
B/P C State     Filename
--- - -------- --------
 B    PendInst  disk0:/sys/s72033/base/s72033-adventerprisek9_wan_dbg-vm(12.2(99)SX1010)
.
.
.
LEGEND:
-------:
B/P/MP - (B)ase image, (P)atch, or (M)aintenance (P)ack
'C' - (C)ommitted
Pruned - This file has been pruned from the system
Active - This file is active in the system
PendInst - This file is set to be made available to run on the
   system after next activation.
PendRoll - This file is set to be rolled back after next activation.
InstPRel - This file will run on the system after next reload
RollPRel - This file will be removed from the system after next reload
RPRPndIn - This file is both rolled back pending a reload, and pending
   installation.  On reload, this file will not run and will move to
   PendInst state.  If 'install activate' is done before reload, pending
   removal and install cancel each other and file simply remains active
IPRPndRo - This file is both installed pending a reload, and pending rollback.
   If the card reloads, it will be active on the system pending a rollback
   If 'install activate' is done before a reload, the pending install and
   removal with cancel each other and the file will simply be removed.
```

Installing Cisco IOS Software Modularity Patch Files on a Single RP

Perform this task to install one or more Software Modularity patches on a device that is running a single Route Processor (RP). After the initial install step, there are additional steps required to activate the patch file to implement the changes in the software.

Before attempting this task, you must perform the Installing Cisco IOS Software Modularity Base Images on a Single RP, page 6 task.

Note If the patch does not require a reload, this task ends at Step 6.

SUMMARY STEPS

1. **enable**
2. **install file** *source-file-url destination-directory [second-destination-directory]* [**interactive**]
3. **show install** *search-root-directory* [**detailed**| **pending**]
4. Repeat Step 2 and Step 3 for each patch file to be installed.
5. **install activate** *search-root-directory* [**reload**]
6. **show install running** [**detailed**| **pending**]
7. **reload**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**install file** *source-file-url destination-directory [second-destination-directory]* [**interactive**]
	
	Example:
	
	`Router# install file rcp://`
	`s72033_rp/patch/`
	`s72033-AMA0001-patch disk0:/sys`
		Installs a Software Modularity patch file from the specified path for the source into the specified local directory for the destination.
		
		• The optional *second-destination-directory argument is the path of a secondary destination directory in which the installable file is to be installed.*
		• The optional **interactive** keyword displays more detailed output.
		
		Note This example shows how to use the remote copy protocol (rcp) to source the file, but any URL that can be used as the source of the **copy** command can be used as the source of the **install file** command.

Step 3	**show install** *search-root-directory* [**detailed**| **pending**] **Example:** `Router# show install disk0:/sys`	(Optional) Displays information about the installed software. • The *search-root-directory*argument displays information about the software installed at the specified location. • The optional **detailed** keyword displays more detailed information about the installed software. • The optional **pending keyword**displays patch upgrade summary information.
Step 4	Repeat Step 2 and Step 3 for each patch file to be installed.	--

	Command or Action	Purpose

Step 5	**install activate** *search-root-directory* [**reload**]
	
	Example:
	
	`Router# install activate`
	`disk0:/sys`
		Activates the current pending change set.
		
		• Enter **Y** when prompted for confirmation.
		• The optional **reload** keyword treats the patch to be activated as a reload patch, thereby bypassing a time-consuming process restart.

Step 6	**show install running** [**detailed**| **pending**] **Example:** `Router# show install running`	(Optional) Displays information about the software that is currently running on each location in the system. • The optional **detailed** keyword displays more detailed information about the installed software. • The optional **pending**keyword displays patch upgrade summary information.
Step 7	**reload** **Example:** `Router# reload`	(Optional) Reloads the operating system. • Use this command only if a reload is required after the **install activate** command in Step 5.

Examples

The following is sample output from the **show install running**command when the **install file** and **install activate** commands have been entered on a single RP device but a reload has not been performed. In this example, the latest change state, InstPRel, is displayed. This change state means that the software is installed and pending a reload.

```
Router# show install running
Software running on card installed at location s72033 - Slot 5 :
B/P C State      Filename
--- - -------- --------
 B     InstPRel disk0:/sys/s72033/base/s72033-adventerprisek9_wan_dbg-vm(12.2(99)SX1010)
Software running on card installed at location s72033_rp - Slot 5 :
B/P C State      Filename
--- - -------- --------
 B     InstPRel disk0:/sys/s72033_rp/base/DRACO2_MP
LEGEND:
-------:
B/P/MP - (B)ase image, (P)atch, or (M)aintenance (P)ack
'C' - (C)ommitted
Pruned - This file has been pruned from the system
Active - This file is active in the system
PendInst - This file is set to be made available to run on the
  system after next activation.
PendRoll - This file is set to be rolled back after next activation.
InstPRel - This file will run on the system after next reload
RollPRel - This file will be removed from the system after next reload
RPRPndIn - This file is both rolled back pending a reload, and pending
    installation.  On reload, this file will not run and will move to
    PendInst state.  If 'install activate' is done before reload, pending
    removal and install cancel each other and file simply remains active
IPRPndRo - This file is both installed pending a reload, and pending rollback.
    If the card reloads, it will be active on the system pending a rollback
    If 'install activate' is done before a reload, the pending install and
    removal with cancel each other and the file will simply be removed.
```

The following is sample output from the **show install running** command after a reload has been performed. This command displays the latest change state to be active (Active).

```
Router# show install running
Software running on card installed at location s72033 - Slot 5 :
B/P C State     Filename
--- - -------- --------
 B    Active   disk0:/sys/s72033/base/s72033-adventerprisek9_wan_dbg-vm(12.2(99)SX1010)
MP             MAINTENANCE PACK MA0005
 P    Active   disk0:/sys/s72033_rp/patch/s72033-AMA0001-patch
Software running on card installed at location s72033_rp - Slot 5 :
B/P C State     Filename
--- - -------- --------
 B    Active   disk0:/sys/s72033_rp/base/DRACO2_MP
MP             MAINTENANCE PACK MA0005
 P    Active   disk0:/sys/s72033_rp/patch/s72033-AMA0001-patch
LEGEND:
-------:
B/P/MP - (B)ase image, (P)atch, or (M)aintenance (P)ack
'C' - (C)ommitted
Pruned - This file has been pruned from the system
Active - This file is active in the system
PendInst - This file is set to be made available to run on the
   system after next activation.
PendRoll - This file is set to be rolled back after next activation.
InstPRel - This file will run on the system after next reload
RollPRel - This file will be removed from the system after next reload
RPRPndIn - This file is both rolled back pending a reload, and pending
   installation.  On reload, this file will not run and will move to
   PendInst state.  If 'install activate' is done before reload, pending
   removal and install cancel each other and file simply remains active
IPRPndRo - This file is both installed pending a reload, and pending rollback.
   If the card reloads, it will be active on the system pending a rollback
   If 'install activate' is done before a reload, the pending install and
   removal with cancel each other and the file will simply be removed.
```

Installing Cisco IOS Software Modularity Base Images on a Dual RP

Perform this task to install a Software Modularity base image on a dual RP device and save the configuration to the running configuration file. Step 6 is an optional step included to allow you to remove all existing software binds before you bind the software. Remember that installing a base Software Modularity image is different from copying the base image to the disk and performing a reload. The install process allows access to the patching functionality.

The base image may be a repackaged image (see the Creating a Repackage of a Cisco IOS Software Modularity Installation, page 44 task).

Depending on the feature set that you want to run, you need a minimum of 256 MB of compact flash memory and we recommend that you have 512 MB of compact flash memory. If you are installing the 512 MB compact flash memory, you must reformat the flash disk before starting a Software Modularity base image installation.

Your system must be running a noninstalled Software Modularity image to perform this task because the **install file** command is available only in Software Modularity images. For initial installation, the Software Modularity image is typically booted and run in a noninstalled mode, just as any other Cisco IOS image would be loaded and booted. After this has been done, the **install file** command can be used to fully install the image on the file system.

To boot a Software Modularity image, follow the same procedure as when booting a Cisco IOS image.

For more information about booting Cisco IOS images, see the " Loading and Managing System Images " section of the *Cisco IOS Configuration Fundamentals Configuration Guide* at the following URL:

http://www.cisco.com/en/US/docs/ios/fundamentals/configuration/guide/cf_system_images.html

✎

Note In this task you remove all the existing **boot system** commands before entering a new **boot system**command for the new installed image. We recommend that you run the **show startup-config** command and note all your existing **boot system** commands to determine which of them must be reentered, and in which order.

SUMMARY STEPS

1. **enable**
2. **install file** *source-file-url destination-directory [second-destination-directory]* [**interactive**]
3. **install file** *source-file-url destination-directory [second-destination-directory]* [**interactive**]
4. **show install** *search-root-directory* [**detailed | pending**]
5. **configure terminal**
6. **no boot system** [*file-url | filename*]
7. **install bind** *search-root-directory* [**prepend**]
8. **boot system** {*file-url | filename*}
9. Repeat Step 8 foreach **boot system** command to be added to the configuration file.
10. **exit**
11. **copy running-config startup-config**
12. **hw-module module** *module-number* **reset**
13. **redundancy force-switchover**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**install file** *source-file-url destination-directory [second-destination-directory]* [**interactive**]
	
	Example:
	
	`Router# install file rcp://s72033/`
	`base/`
	`s72033-adventerprisek9_wan_dbg-vz`
	`disk0:/sys`
		Installs a Software Modularity base image file from the specified path for the source into the specified local directory for the destination on the active RP.
		
		• The optional *second-destination-directory argument is the path of a secondary destination directory in which the installable file is to be installed.*
		• The optional **interactive** keyword displays more detailed output.
		
		Note This example shows how to use the remote copy protocol (rcp) to source the file, but any URL that can be used as the source of the **copy** command can be used as the source of the **install file** command.

	Command or Action	Purpose

Step 3	**install file** *source-file-url destination-directory [second-destination-directory]* [**interactive**]
	
	Example:
	
	`Router# install file rcp://s72033/`
	`base/`
	`s72033-adventerprisek9_wan_dbg-vz`
	`slavedisk0:/sys`
		Installs a Software Modularity base image file from the specified path for the source into the specified local directory for the destination on the standby RP.
		
		• This step is performed to install a file in the standby RP where the destination is a slave device that exists on the standby RP.
		• Use the *destination-directory* argument to specify the slave destination equivalent to the destination on the active RP in Step 2. In this example, the destination is disk0: for the active RP and slavedisk0: for the standby RP.
		• The optional **interactive** keyword displays more detailed output.
		
		Note This example shows how to use the remote copy protocol (rcp) to source the file, but any URL that can be used as the source of the **copy** command can be used as the source of the **install file** command.

Step 4	**show install** *search-root-directory* [**detailed | pending**] **Example:** `Router# show install disk0:/sys`	(Optional) Displays information about the installed software. • The *search-root-directory* argument displays information about the software installed at the specified location. • The optional **detailed** keyword displays more detailed information about the installed software. • The optional **pending keyword** displays patch upgrade summary information.
Step 5	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 6	**no boot system** [*file-url | filename*] **Example:** `Router(config)# no boot system`	(Optional) Removes **boot system** commands from the startup configuration file. • Use the **no boot system** command without any arguments to remove all **boot system** commands from the startup configuration file.

	Command or Action	Purpose
Step 7	**install bind** *search-root-directory* [**prepend**] **Example:** Router(config)# install bind disk0:/sys	Binds the software by generating a **boot system** command in the configuration file that defines a location from which the software will run. • Remember that **boot system** commands in the startup configuration file are executed in the order in which they were configured. • Use the optional **prepend** keyword to move the latest **boot system** statement to the top of the boot variable, which makes that statement the primary image to boot. • If you know the complete directory path and image name, you can use the **boot system** command instead of the **install bind** command. For more details, see the Binding Cisco IOS Software Modularity, page 37 task.
Step 8	**boot system** {*file-url* | *filename*} **Example:** Router(config)# boot system disk0:/sys/s72033-entservicesk9_wan-mz	Adds a **boot system** command to the configuration file. • Use the *file-url* or *filename* argument to specify the directory path and image name. • Use this command to provide a **boot system** command for a backup image. **Note** Only one form of the **boot system** command syntax is shown. For more details, see the Cisco IOS Configuration Fundamentals Command Reference.
Step 9	Repeat Step 8 for each **boot system** command to be added to the configuration file.	--
Step 10	**exit** **Example:** Router(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.
Step 11	**copy running-config startup-config** **Example:** Router# copy running-config startup-config	Copies the running configuration file to the startup configuration file.
Step 12	**hw-module module** *module-number* **reset** **Example:** Router# hw-module module 6 reset	Resets the standby RP, which will reboot and start running the installed code. • Use the *module-number* argument to specify the module number of the standby RP. • After entering this command, wait until the standby RP has rebooted fully before performing the next step.

	Command or Action	Purpose
Step 13	**redundancy force-switchover** **Example:** Router# redundancy force-switchover	Conducts a manual switchover to the redundant supervisor engine for a dual processor redundant system. • The redundant supervisor engine becomes the new active supervisor engine running the new Software Modularity image. The modules are reloaded, and the module software is downloaded from the new active supervisor engine. • The old active supervisor engine reboots with the new image and becomes the redundant supervisor engine.

Installing Cisco IOS Software Modularity Patch Files on a Dual RP

Perform this task to install one or more Software Modularity patch files on a device running dual RPs. Although this task is similar to the task for a single RP, there are additional steps to install and activate the patch file on a dual RP device.

The steps required to activate a Software Modularity patch file are more complex on a dual RP device than on a single RP device. After a patch file is installed on both active and standby RPs, a process restart may be performed if the patch file does not require a reload. The first instance of the **install activate** command causes the standby RP to reset and renegotiate the high availability (HA) mode for the activated patch files. When the standby RP comes back up, if the set of patches that are in an active state is different from the set of patches currently running on the active RP, the standby RP comes up in route processor redundancy (RPR) mode. If a reload is required to activate the patch file, a message is displayed, but no reset is performed.

The second instance of the **install activate** command causes a process restart on the active RP followed by another reset of the standby RP and a renegotiation of the high availability (HA) mode for the activated patch files. At this point both the active and standby RPs should have the same set of patch files in the active state causing the standby RP to come up in the highest HA mode that you have configured. Only the standby RP is being reset so no outage should occur. If a reload is required, the software does not perform a reset.

If a reload is not required for the patch files, the **show install running** command will display the patches in an active state and the task is complete. If a reload is required by the patch files, the display will show the patches in an installed and pending a reload (InstPRel) change state. Use the **hw-module module reset** command for the standby RP module to reset the standby RP and activate the patches on the standby RP. In a similar process to the **install activate** command, the standby RP may come up in RPR mode if the patches are different between the active and standby RP. The **redundancy force-switchover** command is then entered, and the previous active RP resets while the previous standby RP becomes the active RP. If the system is in RPR mode, the switchover causes an outage. After the switchover is complete and the set of patches in an active state is the same on both the active and standby RPs, the software will come up in the highest HA mode that you have configured. Use the **show install running** command to view the state of the patches after the patch file activation is complete.

Before attempting this task, you must perform the Installing Cisco IOS Software Modularity Base Images on a Dual RP, page 12 task.

![Note icon]

Note   If the patch does not require a reload, the task ends at Step 8.

SUMMARY STEPS

1. **enable**

2. **install file** *source-file-url destination-directory [second-destination-directory]* [**interactive**]

3. **install file** *source-file-url destination-directory [second-destination-directory]* [**interactive**]

4. Repeat Step 2 and Step 3 to install more patch files.

5. **show install** *search-root-directory* [**detailed**| **pending**]

6. **install activate** *search-root-directory* [**reload**

7. **install activate** *search-root-directory* [**reload**]

8. **show install running** [**detailed**| **pending**]

9. **hw-module module** *module-number* **reset**

10. **redundancy force-switchover**

11. **show install running** [**detailed**| **pending**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**install file** *source-file-url destination-directory [second-destination-directory]* [**interactive**]
	
	Example:
	
	`Router# install file rcp://`
	`s72033_rp/patch/`
	`s72033-AMA0001-patch`
	`disk0:/sys`
		Installs a Software Modularity patch file from the specified path for the source into the specified local directory for the destination on the active RP.
		
		• The optional *second-destination-directory argument is the path of a secondary destination directory in which the installable file is to be installed.*
		• The optional **interactive** keyword displays more detailed output.
		
		Note This example shows how to use the remote copy protocol (rcp) to source the file, but any URL that can be used as the source of the **copy** command can be used as the source of the **install file** command.

	Command or Action	Purpose

Step 3	**install file** *source-file-url destination-directory [second-destination-directory]* [**interactive**]
	
	Example:
	
	`Router# install file rcp://`
	`s72033_rp/patch/`
	`s72033-AMA0001-patch`
	`slavedisk0:/sys`
		Installs a Software Modularity patch file from the specified path for the source into the specified local directory for the destination on the standby RP.
		
		• This step is performed to install a patch file in the standby RP where the destination is a slave device that exists on the standby RP.
		• The optional **interactive** keyword displays more detailed output.
		
		Note This example shows how to use the remote copy protocol (rcp) to source the file, but any URL that can be used as the source of the **copy** command can be used as the source of the **install file** command.

Step 4	Repeat Step 2 and Step 3 to install more patch files.	--

Step 5	**show install** *search-root-directory* [**detailed**| **pending**]
	
	Example:
	
	`Router# show install`
	`disk0:/sys`
		(Optional) Displays information about the installed software.
		
		• The *search-root-directory* argument displays information about the software installed at the specified location.
		• The optional **detailed** keyword displays more detailed information about the installed software.
		• The optional **pending keyword** displays patch upgrade summary information.

Step 6	**install activate** *search-root-directory* [**reload**
	
	Example:
	
	`Router(config)# install`
	`activate slavedisk0:/sys`
		Activates the current pending change set on the standby RP.
		
		• The search-root-directory here is for the standby RP.
		• Enter **Y** when prompted for confirmation.
		
		Note If the patch file does not require a reload, a process restart may be performed. The standby RP is reset to renegotiate the high availability (HA) mode for the activated patch files. When the standby RP comes back up, if the set of patches that are in an active state is different from the set of patches currently on the active RP, the standby RP comes up in RPR mode. If a reload is required, a message is displayed, but no reset is performed.
		
		• The optional **reload** keyword treats the patch to be activated as a reload patch, thereby bypassing a time-consuming process restart.

	Command or Action	Purpose
Step 7	**install activate** *search-root-directory* [**reload**] **Example:** `Router(config)# install activate disk0:/sys`	Activates the current pending change set on the active RP. • The search-root-directory here is for the active RP. • Enter **Y** when prompted for confirmation. **Note** This second instance of the **install activate** command causes a process restart on the active RP, followed by another reset of the standby RP and a renegotiation of the high availability (HA) mode for the activated patch files. At this point both the active and standby RPs should have the same set of patch files in the active state, causing the standby RP to come up in the highest HA mode that you have configured. Only the standby RP is being reset, so no outage should occur. If a reload is required, the software does not perform a reset.
Step 8	**show install running** [**detailed**|**pending**] **Example:** `Router# show install running`	(Optional) Displays information about the software that is currently running on each location in the system. • The optional **detailed** keyword displays more detailed information about the installed software. • The optional **pending keyword** displays patch upgrade summary information. • If a reload is not required for the patch files, the display shows the patches in an active state and the task is complete. • If a reload is required by the patch files, the display shows the patches in an InstPRel change state. This change state means that the software is installed and pending a reload. **Note** If the patch does not require a reload, the task is complete and Step 9 through Step 11 are not required.
Step 9	**hw-module module** *module-number* **reset** **Example:** `Router# hw-module module 6 reset`	(Optional) Resets the standby RP, which will reboot and start running the installed code. • Use this command only if a reload is required after the **install activate** command in Step 7. • Use the *module-number* argument to specify the module number of the standby RP. • After entering this command, wait until the standby RP has rebooted fully before performing the next step. **Note** This command resets the standby RP and activates the patches on the standby RP. In a similar process to the **install activate** command, the standby RP may come up in RPR mode if the patches are different between the active and standby RP.

	Command or Action	Purpose
Step 10	**redundancy force-switchover** **Example:** Router# redundancy force-switchover	(Optional) Conducts a manual switchover to the redundant supervisor engine for a dual processor redundant system. • Use this command only if a reload is required after the **install activate** command in Step 7. • The redundant supervisor engine becomes the new active supervisor engine running the new Software Modularity image. The modules are reloaded, and the module software is downloaded from the new active supervisor engine. • The old active supervisor engine reboots with the new image and becomes the redundant supervisor engine. **Note** This command resets the previous active RP while the previous standby RP becomes the active RP. If the system is in RPR mode, the switchover causes an outage. After the switchover is complete and the set of patches in an active state is the same on both the active and standby RPs, the software will come up in the highest HA mode that you have configured.
Step 11	**show install running** [**detailed**|**pending**] **Example:** Router# show install running	(Optional) Displays information about the software that is currently running on each location in the system. • The optional **detailed** keyword displays more detailed information about the installed software. • The optional **pending keyword**displays patch upgrade summary information. • This command displays the latest change state to be active (Active).

Examples

The following is sample output from the **show install running** command when the **install file** and **install activate** commands have been entered on a dual RP device, but a reload has not been performed:

```
Router# show install running
Software running on card installed at location s72033 - Slot 5 :
B/P C State      Filename
--- - -------- --------
 B    Active    disk0:/sys/s72033/base/s72033-adventerprisek9_wan_dbg-vm(12.2(99)SX1010)
Software running on card installed at location s72033_rp - Slot 5 :
B/P C State      Filename
--- - -------- --------
 B    Active    disk0:/sys/s72033_rp/base/DRACO2_MP
Software running on card installed at location s72033 - Slot 6 :
B/P C State      Filename
--- - -------- --------
 B    Active    slavedisk0:/sys/s72033/base/s72033-adventerprisek9_wan_dbg-
vm(12.2(99)SX1010)
Software running on card installed at location s72033_rp - Slot 6 :
B/P C State      Filename
--- - -------- --------
 B    Active    slavedisk0:/sys/s72033_rp/base/DRACO2_MP
LEGEND:
-------:
B/P/MP - (B)ase image, (P)atch, or (M)aintenance (P)ack
'C' - (C)ommitted
Pruned - This file has been pruned from the system
Active - This file is active in the system
PendInst - This file is set to be made available to run on the
   system after next activation.
PendRoll - This file is set to be rolled back after next activation.
InstPRel - This file will run on the system after next reload
```

```
RollPRel - This file will be removed from the system after next reload
RPRPndIn - This file is both rolled back pending a reload, and pending
    installation.  On reload, this file will not run and will move to
    PendInst state.  If 'install activate' is done before reload, pending
    removal and install cancel each other and file simply remains active
IPRPndRo - This file is both installed pending a reload, and pending rollback.
    If the card reloads, it will be active on the system pending a rollback
    If 'install activate' is done before a reload, the pending install and
    removal with cancel each other and the file will simply be removed.
```

Upgrading a Cisco IOS Software Image to a Cisco IOS Software Modularity Image

Cisco IOS Software Modularity images cannot be installed directly on a device that is currently running a Cisco IOS software image. Cisco IOS Software Modularity uses the **install file** privileged EXEC command to install new images. Cisco IOS images do not recognize commands that are specific to Software Modularity, so the **install file** command is not present in Cisco IOS images. Therefore, you must perform the following task to install the first Software Modularity image on a device that is currently running a Cisco IOS image.

You can use a TAR file created from a system that is running a Cisco IOS Software Modularity image to extract that configuration onto a system that is running a Cisco IOS software image. This enables you to upgrade a system that is running a Cisco IOS image to a Software Modularity image without the need for multiple system reloads.

Before you can create a TAR file, you must install a Software Modularity image on a system that is running a Software Modularity image. See the Installing Cisco IOS Software Modularity Base Images on a Single RP, page 6 and Installing Cisco IOS Software Modularity Base Images on a Dual RP, page 12.

The device that is running a Software Modularity image must be identical to the system running the Cisco IOS software image that is being upgraded to run a Software Modularity image.

SUMMARY STEPS

1. **enable**
2. **dir** [**/all**] [*filesystem* **:**][*file-url*]
3. **archive tar /create** *destination-url* **flash:/** *file-url*
4. Do one of the following:
    - **copy system:running-config tftp:** [[[*//location*]*/directory*]*/filename*]
    -
    -
    - **copy nvram:startup-config tftp:** [[[*/location*]*/directory/*]*filename*
5. **copy tftp:** [[[*// location*]*/ directory*]*/ filename*] *flash-filesystem* **:**[*filename*]
6. **archive tar /xtract** *source-url* **flash:/** *file-url* [*dir/file*...]
7. **configure terminal**
8. **boot system flash** [*flash-fs* **:**] [*partition-number* **:**] [*filename*]
9. **exit**
10. **config-register** *value*
11. **copy running-config startup-config**
12. **reload**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Switch> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**dir** [**/all**] [*filesystem* **:**][*file-url*] **Example:** Switch# dir disk0: **Example:** **Example:** Directory of disk0:/ **Example:** 1 -rw- 21954344 Feb 20 2006 09:46:22 +00:00 c10k2-p11-mz.122-27.SBB3 . **Example:** . **Example:** . **Example:** 128614400 bytes total (61366272 bytes free)	Displays a list of files on a file system. • Display the list of files on the system to ensure that there is enough space in the boot disk to create a TAR file of the configuration.

Command or Action		Purpose
Step 3	**archive tar /create** *destination-url* **flash:/** *file-url* **Example:** `Switch# archive tar /create disk0:ion_tar flash:/sys`	Creates a TAR file. • For *destination-url*, specify the destination URL alias for the local or network file system and the name of the TAR file to create. • For **flash:/** *file-url*, specify the location on the local flash file system from which the new TAR file is created. You can also specify an optional list of files or directories within the source directory to write to the new TAR file. If none are specified, all files and directories at this level are written to the newly created TAR file.

	Command or Action	Purpose

Step 4	Do one of the following:
	
	  • **copy system:running-config tftp:** [[[*//location*]/*directory*]/*filename*]
	  •
	  •
	  • **copy nvram:startup-config tftp:** [[[*/location*]/*directory/*]*filename*
	
	Example:
	
	`Switch# copy system:running-config tftp://172.16.2.155/ion_tar`
	
	Example:
	
	.
	
	Example:
	
	.
	
	Example:
	
	.
	
	Example:
	
	`Write file ion_tar on host 172.16.2.155? [confirm] y`
		Uploads the file to the TFTP server.
		
		  • Specify the IP address or hostname of the TFTP server and the destination filename.

	Command or Action	Purpose
Step 5	**copy tftp:** [[[// *location*]/ *directory*]/ *filename*] *flash-filesystem* **:**[*filename*]	Copies a Cisco IOS Modular TAR file from TFTP to the flash disk on the system running a Cisco IOS image.
	Example: Switch# copy tftp disk0: [enter]	
	Example: Address or name of remote host []? 172.16.2.155[enter] Source filename []?/ion_tar [enter] Destination filename [ion_tar]? [enter]	
	Example:	
	Example: (device will start displaying progress messages here)	
	Example:	
	Example: !! !! . . . !! [OK 22729716 bytes]	
Step 6	**archive tar /xtract** *source-url* **flash:/** *file-url* [*dir/file*...] **Example:** Switch# archive tar /xtract disk0:ion_tar flash:/sys	Extracts the TAR file containing the Software Modularity configuration.
Step 7	**configure terminal** **Example:** Switch# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 8	**boot system flash** [*flash-fs* **:**] [*partition-number* **:**] [*filename*] **Example:** Switch(config)# boot sys flash disk0:/sys/s72033/base/s72033-adventerprisek9_wan_dbg-vm	Specifies the system image that the devices loads at startup. Configures the system to boot from a system image loaded in internal Flash.
Step 9	**exit** **Example:** Switch(config)# exit	Exits global configuration mode.
Step 10	**config-register** *value* **Example:** Switch# config-register 0x2102	(Optional) Changes the configuration register boot field settings. The configuration register should be configured to autoboot.
Step 11	**copy running-config startup-config** **Example:** Switch# copy running-config startup-config **Example:** Destination filename [startup-config]? **Example:** **Example:** 20338 bytes copied in 10.284 secs (1978 bytes/sec)	Copies the running configuration file to the startup configuration file.

Command or Action	Purpose
Step 12 **reload** **Example:** Switch# reload **Example:** Proceed with reload? [confirm] **Example:** . **Example:** . **Example:** . **Example:** Autoboot executing command: "boot disk0:/sys/s72033/base/s72033-adventerprisek9_wan_dbg-vm	Reloads the operating system to load the Software Modularity image that has been extracted from the TAR file.

Upgrading a Cisco IOS Software Modularity Image

Perform this task to upgrade between Software Modularity images, to upgrade to an installed Software Modularity image, and to save the configuration to the running configuration file. Your system must already be running a Software Modularity image to perform this task.

Installing a base Software Modularity image can be achieved by copying the image onto the flash disk and performing a system reload. This brings the system up in non-installed or single binary file mode. Patching is not available when the system is brought up in single binary file mode. To install a base Software Modularity image so that the system is in installed mode and can perform patching, install the Software Modularity image on the flash disk and reload the system.

The base image may be a repackaged image (see the Creating a Repackage of a Cisco IOS Software Modularity Installation, page 44 task).

To successfully boot an image from ROMMON, the boot device (for example, disk0:) must have a MONLIB file present. The MONLIB file is the ROMMON library used by ROMMON to access files in the flash file system. To verify that a MONLIB file is present, use the **show** *file-system* **system** command. If no MONLIB file is present, you must format the disk before the installation can be performed. To format the disk, use the **format** command in privileged EXEC mode.

✎

Note   In this task you remove all the existing **boot system** commands before entering a new **boot system** command for the upgraded image. We recommend that you run the **show startup-config** command and note all your existing **boot system** commands to determine which of them must be reentered and in which order.

SUMMARY STEPS

1. **enable**
2. **install file** *source-file-url destination-directory [second-destination-directory]* [**interactive**]
3. **show install** *search-root-directory* [**detailed**| **pending**]
4. **configure terminal**
5. **no boot system** [*file-url | filename*]
6. **install bind** *search-root-directory* [**prepend**]
7. **boot system** {*file-url | filename*}
8. Repeat Step 6 and/or Step 7 for each **boot system** command to be added to the configuration file.
9. **exit**
10. **copy running-config startup-config**
11. **show bootvar**
12. **reload**
13. show version
14. **install clear** *search-root-directory*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose

Step 2	**install file** *source-file-url destination-directory [second-destination-directory]* [**interactive**]
	
	Example:
	
	`Router# install file tftp://172.16.1.1/s72033-ipservicesk9-vz.`
	`122-18.SXF12.bin disk0:/newsys`
	
	Example:
	
	`Address or name of remote host [172.16.1.1]?`
	
	Example:
	
	`Source filename [s72033-ipservicesk9-vz.122-18.SXF12.bin]?`
	
	Example:
	
	`!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!`
		Installs a Software Modularity base image file from the specified path for the source into the specified local directory for the destination.
		
		Note The new software modularity image must be installed in a different location than that of the existing image.
		
		• The optional **interactive** keyword displays more detailed output.
		
		Note This example shows how to use the remote copy protocol (rcp) to source the file, but any URL that can be used as the source of the **copy** command can be used as the source of the **install file** command.

Step 3	**show install** *search-root-directory* [**detailed**| **pending**] **Example:** `Router# show install disk0:/newsys`	(Optional) Displays information about the installed software. • The *search-root-directory* argument displays information about the software installed at the specified location. • The optional **detailed** keyword displays more detailed information about the installed software. • The optional **pending keyword** displays patch upgrade summary information.

	Command or Action	Purpose
Step 4	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 5	**no boot system** [*file-url* | *filename*] **Example:** Router(config)# no boot system	Removes **boot system** commands from the startup configuration file. • Use the **no boot system** command without any arguments to remove all **boot system** commands from the startup configuration file. **Note** The **no install bind** command removes only installed Software Modularity images. It does not remove Cisco IOS boot strings. Use the **no boot system** command to remove all boot strings.

	Command or Action	Purpose
Step 6	**install bind** *search-root-directory* [**prepend**] **Example:** `Router(config)# install bind disk0:/newsys`	Binds the software by generating a **boot system** command in the configuration file that defines a new location from which the software will run. • Remember that **boot system** commands in the startup configuration file are executed in the order in which they were configured. • Use the optional **prepend** keyword to move the latest **boot system** statement to the top of the boot variable, which makes that statement the primary image to boot. • If you know the complete directory path and image name, you can use the **boot system** command instead of the **install bind** command. For more details, see the Binding Cisco IOS Software Modularity, page 37 task.

	Command or Action	Purpose
Step 7	**boot system** {*file-url*	*filename*} **Example:** Router(config)# boot system disk0:/newsys/s72033/base/s72033-ipservicesk9-vm	(Optional) Adds a Cisco IOS or noninstalled Software Modulariy boot image to the configuration file. • Use the *file-url*or *filename* argument to specify the directory path and image name. • Use this command to provide a **boot system** command for a backup image. **Note** Only one form of the **boot system** command syntax is shown. For more details, see the *Cisco IOS Configuration Fundamentals Command Reference*.
Step 8	Repeat Step 6 and/or Step 7 for each **boot system** command to be added to the configuration file.	--
Step 9	**exit** **Example:** Router(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.
Step 10	**copy running-config startup-config** **Example:** Router# copy running-config startup-config **Example:** Destination filename [startup-config]? **Example:** 20338 bytes copied in 10.284 secs (1978 bytes/sec)	Copies the running configuration file to the startup configuration file.

Command or Action	Purpose
Step 11 **show bootvar**	(Optional) Displays information about the BOOT environment variable.
Example:	
Router# show bootvar	
Example:	
BOOT variable = disk0:/newsys/s72033/base/s72033-ipservicesk9-vm,12;	
Example:	
CONFIG_FILE variable =	
Example:	
BOOTLDR variable =	
Example:	
Configuration register is 0x2102	

Command or Action	Purpose

Step 12 **reload**
 	
Example:
 	
`Router# reload`
 	
Example:
 	
`Proceed with reload? [confirm]`
 	
Example:
 	
.
 	
Example:
 	
.
 	
Example:
 	
.
 	
Example:
 	
`Autoboot executing command: "boot disk0:/newsys/s72033/base/s72033-`
`ipservicesk9-vm"`
	(Optional) Reloads the operating system to load the upgraded Software Modularity image.
	
	• Perform this step when you are ready to run the base image that was installed in this task. After the reload, the base image becomes an installed image on which patch files can be activated.

	Command or Action	Purpose
Step 13	show version	(Optional) Verifies device is running from the upgraded Software Modularity image.
	Example:	
	Router# show version	
	Example:	
	.	
	Example:	
	.	
	Example:	
	.	
	Example:	
	ROM: System Bootstrap, Version 12.2(17r)S4, RELEASE SOFTWARE (fc1)	
	Example:	
	BOOTLDR:	
	Example:	
	C6509E uptime is 4 minutes	
	Example:	
	Time since C6509E switched to active is 4 minutes	
	Example:	
	System returned to ROM by reload at 19:53:17 UTC Wed Nov 21 2007 (SP by reload)	
	Example:	
	System image file is "disk0:/newsys/s72033/base/s72033-ipservicesk9-vm"	

Command or Action	Purpose

Step 14   **install clear** *search-root-directory*
 	
Example:
 	
`Router# install clear disk0:/sys`
 	
Example:
 	
`Deleting the contents of disk0:/sys`
	(Optional) Removes an entire installed software system.
	
	Note Use the **install clear** command with caution because the command cannot be reversed. After an installation is cleared, it cannot be undone. Software that is currently running or that has been bound to run cannot be cleared. For bound software, you must remove the binding with the **no install bind**command before using the **install clear** command.

Examples

The following sample output from the **show install running** privileged EXEC command shows the output for the base file s72033-lpservicesk9-vm after the **install clear** command has been performed. The Active state means that the file is active in the system.

```
Router# show install running
B/P C State      Filename
--- - -----      --------
Software running on card installed at location c2_lc - Slot 1 :
 B    Active    disk0:/sys/c2_lc/base/C2LC
Software running on card installed at location s72033_rp - Slot 6 :
 B    Active    disk0:/sys/s72033_rp/base/DRACO2_MP
Software running on card installed at location s72033 - Slot 6 :
 B    Active    disk0:/sys/s72033/base/s72033-lpservicesk9-vm - Version 12.2(33)SXH
Software running on card installed at location c2_lc - Slot 8:
 B    Active    disk0:/sys/c2_lc/base/C2LC
LEGEND:
-------:
B/P/MP - (B)ase image, (P)atch, or (M)aintenance (P)ack
'C' - (C)ommitted
Pruned - This file has been pruned from the system
Active - This file is active in the system
PendInst - This file is set to be made available to run on the
   system after next activation.
PendRoll - This file is set to be rolled back after next activation.
InstPRel - This file will run on the system after next reload
RollPRel - This file will be removed from the system after next reload
RPRPndIn - This file is both rolled back pending a reload, and pending
   installation. On the reload, this file will not run and will move to
   PendInst state. If 'install activate' is done before reload, pending
   removal and install cancel each other and file simply remains active
IPRPndRo - This file is both installed pending a reload, and pending rollback.
   If the card reloads, it will be active on the system pending a rollback
   If 'install activate' is done before a reload, the pending install and
   removal will cancel each other and the file will simply be removed
Occluded - This file has been occluded from the system,
   a newer version of itself has superseded it.
```

Binding Cisco IOS Software Modularity

Perform this task to bind the Software Modularity image system-wide or on just one specified node. This task can be useful if you want to change the software to run when you have several installed systems. The **install bind** command generates a **boot system** command, but the **install bind** command is not inserted into the configuration. The benefit of using the **install bind** command is that you just specify the search root directory, which is the destination directory used in the **install file** command, and the software will determine the directory structure and image file. If you use the **boot system** command, you must enter the complete directory path and image name.

Each instance of the **boot system** command generated by an **install bind** command is saved in the configuration file in the order in which it was configured; the normal behavior for **boot system** commands. To configure a system to have the newly installed Software Modularity image as the primary image to boot, you must remove all previous **boot system** commands in the configuration and enter them in the order in which you want them to run. Alternatively, you can download the startup configuration to a text file, insert the new **install bind** or **boot system** command, and copy the changes back into the startup configuration.

To remove all **boot system** commands from the configuration file, use the **no** form of the command without any arguments. Using the **no** form of the **install bind** command will remove only the **boot system** commands for installed software, and leaving other **boot system** commands.

Note Performing this task to bind one or more Software Modularity images changes the startup configuration file, but an image reload or switchover must be performed before the installed and bound image is actually running on the device.

Note In this task you remove all the existing **boot system** commands before entering a new **boot system** command for the new installed image. We recommend that you run the **show startup-config** command and note all your existing **boot system** commands to determine which of them must be reentered, and in which order.

SUMMARY STEPS

1. **enable**
2. **show startup-config**
3. **configure terminal**
4. **no boot system** [*file-url* | *filename*]
5. **install bind** *search-root-directory* [**prepend**
6. Repeat Step 5 , if required, to bind each system in order of priority.
7. **boot system** {*file-url* | *filename*}
8. Repeat Step 7 for each boot system command to be added to the configuration file.
9. **exit**
10. **copy running-config startup-config**
11. **show startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show startup-config** **Example:** Router# show startup-config	(Optional) Displays the contents of the startup configuration file.
Step 3	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 4	**no boot system** [*file-url*	*filename*] **Example:** Router(config)# no boot system	(Optional) Removes **boot system** commands from the startup configuration file. • Use the **no boot system** command without any arguments to remove all **boot system** commands from the startup configuration file.
Step 5	**install bind** *search-root-directory* [**prepend** **Example:** Router(config)# install bind disk0:/sys	Binds the software by generating a **boot system** command in the configuration file that defines a location from which the software will run. • Remember that **boot system** commands in the startup configuration file are executed in the order in which they were configured. • If you know the complete directory path and image name, you can use the **boot system** command instead of the **install bind** command.
Step 6	Repeat Step 5 , if required, to bind each system in order of priority.	--

Command or Action	Purpose

Step 7   **boot system** {*file-url* | *filename*}
 	
Example:
 	
`Router(config)# boot system disk0:/sys/`
`s72033-entservicesk9_wan-mz`
	Adds a **boot system** command to the configuration file.
	
	• Use the *file-url* or *filename* argument to specify the directory path and image name.
	• Use this command to provide a **boot system** command for a backup image.
	
	Note Only one form of the **boot system** command syntax is shown. For more details, see the *Cisco IOS Configuration Fundamentals Command Reference*.

Step 8 Repeat Step 7 for each boot system command to be added to the configuration file.	--
Step 9 **exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 10 **copy running-config startup-config** **Example:** `Router# copy running-config startup-config`	Copies the running configuration file to the startup configuration file.
Step 11 **show startup-config** **Example:** `Router# show startup-config	boot system`	(Optional) Displays the startup configuration file. • In this example, an output modifier is used to show only the boot system commands.

Examples

The following sample output from the **show startup-config** command with the optional modifier shows all the entries for **boot system** commands in the order in which they were configured:

```
Router# show startup-config | boot system
boot system disk0:/sys/s72033/base/s72033-adventerprisek9_wan_dbg-vm(12.2(99)SX1010)
boot system disk0:/sys/s72033/patch/patch-AAA1258-patch-0-n.so
```

Defining Tags to Roll Back the Cisco IOS Software Modularity Installation

Perform this task to create a tag to define an installation that was set at a point in time. If a subsequent installation of a patch file adversely affects the installed system, a rollback can be performed using the defined tag.

There are three Cisco-defined rollback tags:

• CISCO_BASE--This tag is defined as the base image with no patches or other tags. Using this tag with the **install rollback** command takes you back to the installed base image.

- CISCO_LATEST--This tag is defined as removing one level of install file. Using this tag with the **install rollback** command removes the set of files that were added with the last **install file** command entry. Effectively, the software rolls back the most recently installed patch whether active or not. If the patch is in an active state, it will set the patch to a PendRoll state, meaning that the changes will not take place until the **install activate** command is entered. If the patch has been installed but is not activated, the **install rollback** command removes the installed patch.
- CISCO_LATEST_ACTIVATE--This tag is defined as removing one level of install activation. Using this tag with the **install rollback** command removes the set of files that were most recently activated by the **install activate** command.

Do not use these tag names for your tags. If you do not create any tags, these tags are defined by default and can be used with the **install rollback** command.

SUMMARY STEPS

1. **enable**
2. **install commit** *search-root-directory tag-name*
3. **install prune** *search-root-directory tag-name* [**files**]
4. **show install** [**tags**] {**running**| *search-root-directory*} [**tagname** *tag-name*][**detailed** | **pending**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**install commit** *search-root-directory tag-name* **Example:** Router# install commit disk0:/sys MA0005	Defines a tag name for a set of software files installed at the root directory specified in the **install file** command.
Step 3	**install prune** *search-root-directory tag-name* [**files**] **Example:** Router# install prune disk0:/sys tag1	(Optional) Removes a previously defined tag or unused files from the installed software. • All files no longer required by the system as a result of the tag removal are deleted. • The optional **files** keyword removes all of the tags from the base image to the tag specified except for the specified tag. After this command is entered with the optional **files** keyword, rollback cannot be done to any tag beyond the specified tag; rollback can be performed to the base image only.

	Command or Action	Purpose
Step 4	**show install** [**tags**] {**running**| *search-root-directory*} [**tagname** *tag-name*][**detailed** | **pending**]	(Optional) Displays information about the installed software.
		• The optional **tags** keyword displays the tag information defined for this installed software.
		• The **running** keyword displays information about the software currently running on each location in the system.
	Example:	• The *search-root-directory*argument displays information about the software installed at the specified directory.
	Router# show install tags running	• The **tagname** *tag-name keyword/argument pair*displays information for a particular tag.
		• The optional **detailed** keyword displays more detailed information about the installed software.
		• The optional **pending keyword**displays patch upgrade summary information.

Examples

The following sample output from the **show install** privileged EXEC command displays information about the tags, base and MA0005, that have been defined for this system:

```
Router# show install tags running
Tags defined over software running on location s72033 - Slot 5 :
Tagname               # of Files         Date Committed
-------------------- ----------      -----------------------
base                         1           20:08:51 UTC Sep 9 2005
MA0005                       1           20:34:16 UTC Sep 9 2005
Tags defined over software running on location s72033_rp - Slot 5 :
Tagname               # of Files         Date Committed
-------------------- ----------      -----------------------
base                         1           20:08:51 UTC Sep 9 2005
MA0005                       1           20:34:16 UTC Sep 9 2005
Tags defined over software running on location s72033 - Slot 6 :
Tagname               # of Files         Date Committed
-------------------- ----------      -----------------------
base                         1           20:28:54 UTC Sep 9 2005
Tags defined over software running on location s72033_rp - Slot 6 :
Tagname               # of Files         Date Committed
-------------------- ----------      -----------------------
base                         1           20:28:54 UTC Sep 9 2005
```

Using Tags to Roll Back the Cisco IOS Software Modularity Installation

Perform this task to roll back the Software Modularity installation using tags that define an installation that was set at a point in time. All installation actions performed since the tag was defined are deleted, and the processes affected by the rollback of installed software are restarted. After the restart, these processes use the software that was present at the time at which the tag was created.

Unless you are using the Cisco-defined tags, you must define tags before you perform this task. For more details, see the Defining Tags to Roll Back the Cisco IOS Software Modularity Installation, task.

SUMMARY STEPS

1. **enable**
2. **install rollback** *search-root-directory tag-name*
3. Repeat Step 2 if additional levels of rollback are required.
4. **show install** [**tags**] {**running**| *search-root-directory*} [**tagname** *tag-name*][**detailed | pending**]
5. **install activate** *search-root-directory* [**reload**]
6. **show install running** [**detailed**| **pending**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**install rollback** *search-root-directory tag-name*
	
	Example:
	
	`Router# install rollback`
	`disk0:/sys base`
		Performs a rollback of the installed software to the point at which the specified tag was defined.
		
		• Files and tags defined after the specified tag are removed.
		• All processes affected by the removed software are restarted.

Step 3	Repeat Step 2 if additional levels of rollback are required.	--
Step 4	**show install** [**tags**] {**running**| *search-root-directory*} [**tagname** *tag-name*][**detailed	pending**] **Example:** `Router# show install running`	(Optional) Displays information about the installed software. • The optional **tags** keyword displays the tag information defined for this installed software. • The **running** keyword displays information about the software currently running on each location in the system. • The *search-root-directory*argument displays information about the software installed at the specified directory. • The **tagname** *tag-name keyword/argument pair*displays information for a particular tag. • The optional **detailed** keyword displays more detailed information about the installed software. • The optional **pending keyword**displays patch upgrade summary information.

	Command or Action	Purpose
Step 5	**install activate** *search-root-directory* [**reload**] **Example:** `Router# install activate disk0:/sys`	(Optional) Activates the current pending change set. • Enter **Y** when prompted for confirmation. **Note** You may want to save all your state changes and run this command once. • The optional **reload**keyword treats the patch to be activated as a reload patch, thereby bypassing a time-consuming process restart.
Step 6	**show install running** [**detailed**| **pending**] **Example:** `Router# show install running`	(Optional) Displays information about the software that is currently running on each location in the system. • This command displays the latest change state.

Examples

The following is sample output from the **show install tags running**command after the **install rollback** command using the tag base has been entered, but before activation:

```
Router# show install tags running
Tags defined over software running on location s72033 - Slot 5 :
Tagname             # of Files         Date Committed
------------------- ----------     -----------------------
base                     1              20:08:51 UTC Sep 9 2005
Tags defined over software running on location s72033_rp - Slot 5 :
Tagname             # of Files         Date Committed
------------------- ----------     -----------------------
base                     1              20:08:51 UTC Sep 9 2005
Tags defined over software running on location s72033 - Slot 6 :
Tagname             # of Files         Date Committed
------------------- ----------     -----------------------
base                     1              20:28:54 UTC Sep 9 2005
Tags defined over software running on location s72033_rp - Slot 6 :
Tagname             # of Files         Date Committed
------------------- ----------     -----------------------
base                     1              20:28:54 UTC Sep 9 2005
```

The following is sample output from the **show install running detailed**command after the **install rollback** command has been entered to roll back the software from the MA0005 tag back to the base tag and after the **install activate** command has been entered:

```
Router# show install running detailed
Software running on card installed at location s72033 - Slot 5 :
Base image : disk0:/sys/s72033/base/s72033-adventerprisek9_wan_dbg-vm
Version : 12.2(99)SX1010
File state: Active    File Checksum : 8BB2F966EA945E8E25010A1BAC7205C3DFBCA197
Date Installed : 19:51:22 UTC Sep 8 2005 Commit Tags :  base
Software running on card installed at location s72033_rp - Slot 5 :
Base image : disk0:/sys/s72033_rp/base/DRACO2_MP
File state: Active    File Checksum : 48849DBB2E47A8C55AC68CF3F6EE747B054CD392
Date Installed : 19:49:06 UTC Sep 8 2005 Commit Tags :  base
Software running on card installed at location s72033 - Slot 6 :
Base image : slavedisk0:/sys/s72033/base/s72033-adventerprisek9_wan_dbg-vm
Version : 12.2(99)SX1010
File state: Active    File Checksum : 8BB2F966EA945E8E25010A1BAC7205C3DFBCA197
Date Installed : 19:32:21 UTC Sep 8 2005 Commit Tags :  base Patch : slavedisk0:/sys/
s72033/patch/patch-AAA1258-patch-0-n.so
File state: PendInst  File Checksum : A129339A6A3ED1F8B92D6992AD1BE67C716E4430
```

```
Date Installed : 20:31:01 UTC Sep 9 2005 Commit Tags : NONE  Maintenance Pack : MA0005
Software running on card installed at location s72033_rp - Slot 6 :
Base image : slavedisk0:/sys/s72033_rp/base/DRACO2_MP
File state: Active    File Checksum : 48849DBB2E47A8C55AC68CF3F6EE747B054CD392
Date Installed : 19:31:19 UTC Sep 8 2005 Commit Tags :  base Patch : slavedisk0:/sys/
s72033_rp/patch/patch-AAA1258-patch-0-n.so
File state: PendInst  File Checksum : A129339A6A3ED1F8B92D6992AD1BE67C716E4430
Date Installed : 20:30:55 UTC Sep 9 2005 Commit Tags : NONE Maintenance Pack : MA0005
```

Creating a Repackage of a Cisco IOS Software Modularity Installation

Perform this task to create a repackage (replication) of a Software Modularity image and related patches for installation on multiple routers. While the image repackage is being created, the Software Modularity Installer saves everything in the installed state, including rollback tags. An initial boot must be performed on the device on which the repackaged image is to be installed.

To install the repackaged image as a base image, use the Installing Cisco IOS Software Modularity Base Images on a Single RP, page 6 or the Installing Cisco IOS Software Modularity Base Images on a Dual RP, page 12 task.

SUMMARY STEPS

1. **enable**
2. **show install** [**tags**] {**running**| *search-root-directory*} [**tagname** *tag-name*][**detailed | pending**]
3. **install repackage** *source-root-directory destination-file-url* [**compress**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**show install** [**tags**] {**running**| *search-root-directory*} [**tagname** *tag-name*][**detailed	pending**]
		
		Example:
		
		`Router# show install disk0:/`
		`sys detailed`
			(Optional) Displays information about the installed software.
			
			• The optional **tags** keyword displays the tag information defined for this installed software.
			• The **running** keyword displays information about the software currently running on each location in the system.
			• The *search-root-directory*argument displays information about the software installed at the specified directory.
			• The **tagname** *tag-name keyword/argument pair*displays information for a particular tag.
			• The optional **detailed** keyword displays more detailed information about the installed software.
			• The optional **pending keyword**displays patch upgrade summary information.

Command or Action	Purpose
Step 3 **install repackage** *source-root-directory destination-file-url* [**compress**] **Example:** `Router# install repackage disk0:/sys disk0:/s72033-enterprise-vz`	Generates an installation file by replicating an installed system. • The repackaged file can be used as a backup file or for installation on multiple devices. • The optional **compress** keyword compresses the generated file.

Examples

The following sample output from the **show install** command shows detailed information about the software currently running in the system:

```
Router# show install disk0:/sys detailed
Software running on card installed at location s72033 - Slot 5 :
Base image : disk0:/sys/s72033/base/s72033-adventerprisek9_wan_dbg-vm
Version : 12.2(99)SX1010
File state: Active    File Checksum : 8BB2F966EA945E8E25010A1BAC7205C3DFBCA197
Date Installed : 19:51:22 UTC Sep 8 2005 Commit Tags :  base
Software running on card installed at location s72033_rp - Slot 5 :
Base image : disk0:/sys/s72033_rp/base/DRACO2_MP
File state: Active    File Checksum : 48849DBB2E47A8C55AC68CF3F6EE747B054CD392
Date Installed : 19:49:06 UTC Sep 8 2005 Commit Tags :  base
```

The following sample output from the **show install**command shows information about the software currently running in the system:

```
Router# show install disk0:/sys
B/P C State     Filename
--- - -------- --------
 B    Active   disk0:/sys/s72033/base/s72033-adventerprisek9_wan_dbg-vm(12.2(99)SX1010)
MP             MAINTENANCE PACK MA0005
 P    Active   disk0:/sys/s72033_rp/patch/s72033-AMA0001-patch
B/P C State     Filename
--- - -------- --------
 B    Active   disk0:/sys/aboot/base/LCP_ABOOT
B/P C State     Filename
--- - -------- --------
 B    Active   disk0:/sys/ax1000/base/LCP_AX1000
B/P C State     Filename
--- - -------- --------
 B    Active   disk0:/sys/ax10100/base/LCP_AX10100
B/P C State     Filename
--- - -------- --------
 B    Active   disk0:/sys/boot/base/LCP_BOOT
B/P C State     Filename
--- - -------- --------
 B    Active   disk0:/sys/c2_lc/base/C2LC
MP             MAINTENANCE PACK MA0005
 P    Active   disk0:/sys/s72033_rp/patch/s72033-AMA0099-patch
B/P C State     Filename
--- - -------- --------
 B    Active   disk0:/sys/chevyslc/base/CHEVYS-LC
B/P C State     Filename
--- - -------- --------
 B    Active   disk0:/sys/cp10g/base/LCP_CP10G
B/P C State     Filename
--- - -------- --------
 B    Active   disk0:/sys/cpfab/base/LCP_CPFAB
B/P C State     Filename
```

```
     --- - -------- --------
      B    Active    disk0:/sys/cpgbit/base/LCP_CPGBIT
     B/P C State     Filename
     --- - -------- --------
      B    Active    disk0:/sys/cpmbit/base/LCP_CPMBIT
     B/P C State     Filename
     --- - -------- --------
      B    Active    disk0:/sys/cpmbit2/base/LCP_CPMBIT2
     B/P C State     Filename
     --- - -------- --------
      B    Active    disk0:/sys/cpxbit/base/LCP_CPXBIT
     B/P C State     Filename
     --- - -------- --------
      B    Active    disk0:/sys/cwpa/base/CWPA_version_10.10
     B/P C State     Filename
     --- - -------- --------
      B    Active    disk0:/sys/cwpa2/base/CWPA2_version_10.10
     B/P C State     Filename
     --- - -------- --------
      B    Active    disk0:/sys/cwpa2_fpd/base/CWPA2_FPD_version_10.10
     B/P C State     Filename
     --- - -------- --------
      B    Active    disk0:/sys/cwtlc/base/CWTLC_version_10.10
     B/P C State     Filename
     --- - -------- --------
      B    Active    disk0:/sys/lx1000/base/LCP_LX1000
     B/P C State     Filename
     --- - -------- --------
      B    Active    disk0:/sys/lx10100/base/LCP_LX10100
     B/P C State     Filename
     --- - -------- --------
      B    Active    disk0:/sys/macedon_lc/base/MACEDON
     B/P C State     Filename
     --- - -------- --------
      B    Active    disk0:/sys/s72033_rp/base/DRACO2_MP
     MP             MAINTENANCE PACK MA0005
      P    Active    disk0:/sys/s72033_rp/patch/s72033-AMA0200-patch
     B/P C State     Filename
     --- - -------- --------
      B    Active    disk0:/sys/sip1/base/SIP1_version_10.10
     B/P C State     Filename
     --- - -------- --------
      B    Active    disk0:/sys/sip10g/base/SIP10G_version_10.10
     B/P C State     Filename
     --- - -------- --------
      B    Active    disk0:/sys/sip2/base/SIP2_version_10.10
     B/P C State     Filename
     --- - -------- --------
      B    Active    disk0:/sys/smsc/base/SMSC_version_10.10
     LEGEND:
     -------:
     B/P/MP - (B)ase image, (P)atch, or (M)aintenance (P)ack
     'C' - (C)ommitted
     Pruned - This file has been pruned from the system
     Active - This file is active in the system
     PendInst - This file is set to be made available to run on the
        system after next activation.
     PendRoll - This file is set to be rolled back after next activation.
     InstPRel - This file will run on the system after next reload
     RollPRel - This file will be removed from the system after next reload
     RPRPndIn - This file is both rolled back pending a reload, and pending
        installation.  On reload, this file will not run and will move to
        PendInst state.  If 'install activate' is done before reload, pending
        removal and install cancel each other and file simply remains active
     IPRPndRo - This file is both installed pending a reload, and pending rollback.
        If the card reloads, it will be active on the system pending a rollback
        If 'install activate' is done before a reload, the pending install and
        removal with cancel each other and the file will simply be removed.
```

Configuration Examples for Installing and Configuring Cisco IOS Software Modularity

Example Installing Cisco IOS Software Modularity Base Images and Patch Files on a Single RP

In the following example, the Software Modularity Installer is used to install a Software Modularity image and then bind the image directory. A tag is created and the installation is replicated to create a repackage file. After a patch file is installed and the pending change state is activated, a decision is made to perform a roll back of the software to the point in time when tag1 was created. The processes affected by the roll back are then restarted and tag1 is deleted.

```
Router# install file rcp://s72033/base/s72033-adventerprisek9_wan_dbg-vz disk0:/sys
Router# configure terminal
Router(config)# no boot system
Router(config)# install bind disk0:/sys
Router(config)# exit
Router# install commit disk0:/sys tag1
Router# install repackage disk0:/sys disk0:/s72033-finance-vz.repackage
!
Router# install file rcp://username@hostname//s72033_rp/patch/s72033-AMA0001-patch

Router# disk0:/sys
Router# install activate disk0:/sys
Router# install rollback disk0:/sys tag1
Router# install activate disk0:/sys
Router# install prune disk0:/sys tag1
Router# copy running-config startup-config
Router# reload
```

Example Installing Cisco IOS Software Modularity Base Images and Patch Files on a Dual RP

In the following example, the Software Modularity Installer is used to install a base Software Modularity image and a patch file on a dual RP device. The **boot system** commands are removed, a software bind is entered and followed by another **boot system** command for a backup image. A patch file is installed and the standby RP is activated. When the standby RP comes up, the active RP is activated. The configuration file is copied to the startup configuration file, and a switchover is performed.

```
Router# install file rcp://s72033/base/s72033-adventerprisek9_wan_dbg-vz disk0:/sys
Router# install file rcp://s72033/base/s72033-adventerprisek9_wan_dbg-vz slavedisk0:/sys
Router# configure terminal
Router(config)# no boot system
Rotuer(config)# install bind disk0:/sys
Router(config)# boot system disk0:/sys/s72033-entservicesk9_wan-mz
Router(config)# exit
Router# install file rcp://s72033_rp/patch/s72033-AMA0001-patch disk0:/sys
Router# install file rcp://s72033_rp/patch/s72033-AMA0001-patch slavedisk0:/sys
```

```
Router# install activate slavedisk0:/sys
Router# install activate disk0:/sys
Router# copy running-config startup-config
Router# hw-module module 5 reset
Router# redundancy force-switchover
```

Example Upgrading a Cisco IOS Software Modularity Image

In the following example, the Software Modularity Installer is used to upgrade a Software Modularity image and then bind the image to a new directory. The **install clear**command is then used to remove the older Software Modularity image from its original directory.

```
Router# show startup-config
.
.
.
C6509E uptime is 1 hour, 2 minutes
Uptime for this control processor is 1 hour, 1 minute
Time since C6509E switched to active is 1 hour, 1 minute
System returned to ROM by reload at 17:07:40 UTC Wed Nov 21 2007 (SP by reload)
System image file is "disk0:/sys/s72033/base/s72033-ipservicesk9-vm"
.
.
.
Router# dir disk0:
Directory of disk0:/
    1  drwx          0  Nov 21 2007 16:39:56 +00:00  sys
   75  drwx          0   Sep 7 2006 20:47:02 +00:00  USER_TCL
   81  -rwx      20497  Nov 21 2007 17:06:08 +00:00  c6509e.cfg
Router# install file tftp://172.16.1.1/s72033-ipservicesk9-vz.122-18.SXF12.bin
Router# disk0:/newsys
Address or name of remote host [172.16.1.1]?
Source filename [s72033-ipservicesk9-vz.122-18.SXF12.bin]?
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! ....
.
.
.
Router# dir disk0:
Directory of disk0:/
    1  drwx          0  Nov 21 2007 16:39:56 +00:00  sys
   82  drwx          0  Nov 21 2007 19:01:30 +00:00  newsys
   75  drwx          0   Sep 7 2006 20:47:02 +00:00  USER_TCL
   81  -rwx      20353  Nov 21 2007 18:30:04 +00:00  c6509e.cfg
Router# configure terminal

Router(config)# no boot system
Router(config)# install bind disk0:/newsys

Router(config)# boot system disk0:/newsys/s72033/base/s72033-ipservicesk9-vm
Router(config)# exit

Router# copy running-config startup-config
Destination filename [startup-config]?
.
.
.
20338 bytes copied in 10.284 secs (1978 bytes/sec)
Router# show bootvar
BOOT variable = disk0:/newsys/s72033/base/s72033-ipservicesk9-vm,12;
CONFIG_FILE variable =
BOOTLDR variable =
Configuration register is 0x2102
Router# reload
Proceed with reload? [confirm]
.
.
.
Autoboot executing command: "boot disk0:/newsys/s72033/base/s72033-ipservicesk9-vm"
.
.
```

```
.
Router# show version
ROM: System Bootstrap, Version 12.2(17r)S4, RELEASE SOFTWARE (fc1)
BOOTLDR:
C6509E uptime is 4 minutes
Time since C6509E switched to active is 4 minutes
System returned to ROM by reload at 19:53:17 UTC Wed Nov 21 2007 (SP by reload)
System image file is "disk0:/newsys/s72033/base/s72033-ipservicesk9-vm"
Router# install clear disk0:/sys
Deleting the contents of disk0:/sys
```

Where to Go Next

If you want to implement event management capability using Software Modularity images, see the "Embedded Event Manager Overview, " "Writing Embedded Event Manager Policies Using the Cisco IOS CLI, " and "Writing Embedded Event Manager Policies Using Tcl" modules.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Commands specific to Cisco IOS Software Modularity: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	Cisco IOS Software Modularity Command Reference
Commands specific to the Catalyst 6500 series: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	Catalyst 6500 Series MSFC Cisco IOS Command Reference , Release 12.2SX
Embedded Event Manager	*Embedded Event Manager Overview*
Embedded Event Manager policy writing	• *Writing Embedded Event Manager Policies Using the Cisco IOS CLI* • *Writing Embedded Event Manager Policies Using Tcl*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
CISCO-PROCESS-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL:
	http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Installing and Configuring Cisco IOS Software Modularity

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2   *Feature Information for Cisco IOS Software Modularity Installation and Configuration*

Feature Name	Releases	Feature Information
Most Cisco IOS features are supported in Cisco IOS Release 12.2(18)SXF4 Software Modularity images.	12.2(18)SXF4	--
Modular IOS Usability - Software Modularity Installer over Multiple Paths	12.2(33)SXI1	The **install file**, **install bind**, **show install**, and **install activate** commands have been enhanced to support the Software Modularity Installer, which is used to manage the installation of base images and patches on an Software Modularity system.
Process MIB Enhancements for Software Modularity	12.2(33)SXH	The CISCO-PROCESS-MIB has been enhanced to support Portable Operating System Interface (POSIX) operating systems such as Cisco Software Modularity. Cisco Software Modularity images have been updated to implement the enhanced CISCO-PROCESS-MIB. To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs