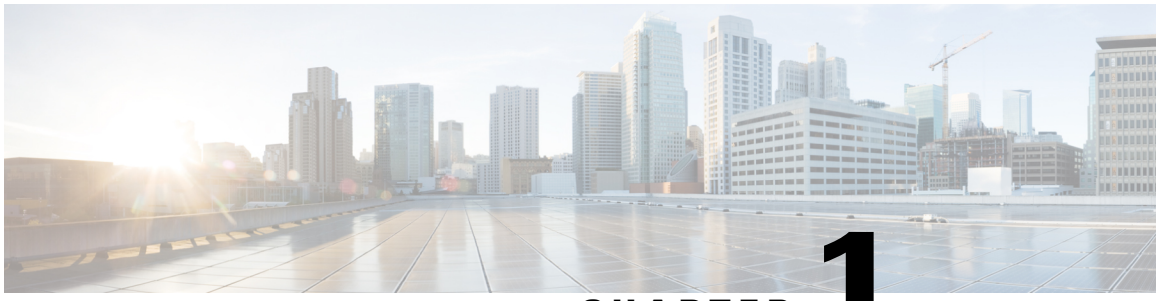
The Integrated File System Configuration Guide, Cisco IOS Release 15M&T

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Using the Cisco IOS Integrated File System

The Cisco IOS File System (IFS) feature provides a single interface to all the file systems available on your routing device, including the following:

- Flash memory file systems
- Network file systems (TFTP, rcp, and FTP)
- Any other endpoint for reading or writing data (such as NVRAM, the running configuration, ROM, raw system memory, system bundled microcode, Xmodem, Flash load helper log, modems, and BRI multiplexing device [mux] interfaces)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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.

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.

Prerequisites for Cisco IOS Integrated File System

- You should have at least a basic familiarity with the Cisco IOS environment and the command-line interface.

- You should have at least a minimal configuration running on your system.

Restrictions for Cisco IOS Integrated File System

- You must have your network up and running, with Cisco IOS Release 12.2 or a later release installed.
- Some of the Cisco IOS configuration commands are only available on certain router platforms, and the command syntax may vary on different platforms.

Information About Cisco IOS Integrated File System

Overview of the IFS

Display and Classify Files

With IFS, all files can be viewed and classified (image, text file, and so on), including files on remote servers. For example, you may want to determine the size and type of an image on a remote server before you copy it to ensure that it is a valid image. You can also display a configuration file on a remote server to verify that it is the correct configuration file before you load the file on the router.

Platform-Independent Commands

With IFS, the file system user interface is no longer platform-specific. Commands have the same syntax, regardless of which platform is used. Thus, you can use the same commands for all of your routers.

However, not all commands are supported on all platforms and file systems. Because different types of file systems support different operations, certain commands are not available for all file systems. Platforms will support commands for the file systems they use.

Minimal Prompting for Commands

IFS minimizes the required prompting for many commands, such as the **copy** EXEC command. You can enter all of the required information in the command line, rather than needing to provide information when the system prompts you for it. For example, if you want to copy a file to an FTP server, on a single line you can specify the specific location on the router of the source file, the specific location of the destination file on the FTP server, and the username and password to use when connecting to the FTP server. However, to have the router prompt you for the needed information, you can still enter the minimal form of the command.

Depending on the current configuration of the **fileprompt** global configuration command and the type of command you entered, the router may prompt you for confirmation, even if you have provided all the information in the command. In these cases, the default value will be the value entered in the command. Press Return to confirm the values.

Create and Navigate Directories

With IFS, you can navigate to different directories and list the files in a directory. On newer platforms, you can create subdirectories in Flash memory or on a disk.

URL Specification for Locating Files

The new file system interface uses Uniform Resource Locators (URLs) to specify the location of a file. URLs are commonly used to specify files or locations on the World Wide Web. However, on Cisco routers, they can now be used to specify the location of files on the router or remote file servers.

On Cisco routers, use URLs in commands to specify the location of the file or directory. For example, if you want to copy a file from one location to another, use the **copy***source-urldestination-url* EXEC command.

The format of URLs used by the routers can vary from the format you may be used to using. There are also a variety of formats that can be used, based on the location of the file.

Files on a Network Server

To specify a file on a network server, use one of the following forms:

- **ftp:** [[// [*username*[:*password@location*] /*directory*] /*filename*
- **rcp:** [[// [*username@location*] /*directory*] /*filename*
- **tftp:** [[//*location*] /*directory*] /*filename*

The *location* can be an IP address or a host name. The *username* variable, if specified, overrides the username specified by the **iprcmdremote-username** or **ipftpusername** global configuration command. The *password* overrides the password specified by the **ipftppassword** global configuration command.

The file path (directory and filename) is specified relative to the directory used for file transfers. For example, on UNIX file servers, TFTP pathnames start in the /tftpboot directory, and rcp and FTP paths start in the home directory associated with the username.

The following example specifies the file named c7200-j-mz.112-current on the TFTP server named myserver.cisco.com. The file is located in the directory named /tftpboot/master.

```
tftp://myserver.cisco.com/master/c7200-j-mz.112-current
```

The following example specifies the file named mill-config on the server named enterprise.cisco.com. The router uses the username liberty and the password secret to access this server via FTP.

```
ftp://liberty:secret@enterprise.cisco.com/mill-config
```

Local Files

Use the *prefix***:***directory*/*filename*syntax to specify a file located on the router. You can use this form to specify a file in Flash memory or NVRAM.

For example, nvram:startup-config specifies the startup configuration in NVRAM, and flash:configs/backup-config specifies the file named backup-config in the configs directory of Flash memory.

When referring to a file system instead of a file, use the *prefix***:** form. This form specifies the file system itself, rather than a file in the file system. Use this form to issue commands on file systems themselves, such as commands to list the files in a file system or to format the file system.

For example, slot0: can indicate the first Personal Computer Memory Card Industry Association (PCMCIA) Flash memory card in slot 0.

URL Prefixes

The URL prefix specifies the file system. The list of available file systems differs by platform and operation. Refer to your product documentation or use the **showfilesystems** EXEC command to determine which prefixes are available on your platform. File system prefixes are listed in the table below.

Table 1: File System Prefixes

Prefix	File System
bootflash:	Boot Flash memory.
disk0:	Rotating media.
flash:	Flash memory. This prefix is available on all platforms. For platforms that do not have a device named flash:, the prefix flash: is aliased to slot0:. Therefore, you can use the prefix flash: to refer to the main Flash memory storage area on all platforms.
flh:	Flash load helper log files.
ftp:	FTP network server.
null:	Null destination for copies. You can copy a remote file to null to determine its size.
nvram:	NVRAM.
rcp:	Remote copy protocol network server.
slavebootflash:	Internal Flash memory on a slave RSP card of a router configured for high system availability (HSA).
slavenvram:	NVRAM on a slave Route/Switch Processor (RSP) card of a router configured for HSA.
slaveslot0:	First PCMCIA card on a slave RSP card of a router configured for HSA.
slaveslot1:	Second PCMCIA card on a slave RSP card of a router configured for HSA.
slot0:	First PCMCIA Flash memory card.

Prefix	File System
slot1:	Second PCMCIA Flash memory card.
system:	Contains the system memory, including the running configuration.
tftp:	TFTP network server.
xmodem:	Obtain the file from a network machine using the Xmodem protocol.
ymodem:	Obtain the file from a network machine using the Ymodem protocol.

Note Maintenance Operation Protocol (MOP) servers are no longer supported as file systems.

In all commands, the colon is required after the file system name. However, commands that did not require the colon previously will continue to be supported, although they will not be available in the context-sensitive help.

URL Prefix for Partitioned Devices

For partitioned devices, the URL prefix includes the partition number. The syntax is *device***:***partition-number***:**for the prefix on a partitioned device.

For example, flash:2:refers to the second partition in Flash memory.

URL Component Lengths

The table below lists the maximum lengths in characters of the different URL components.

Table 2: URL Component Lengths

Component	Length (Number of Characters)
Prefix	31
Username	15
Password	15
Hostname	31
Directory	63
Filename	63

URLs in Commands

Depending on which command you are using, different file systems are available . Some file systems can only serve as a source for files, not a destination. For example, you cannot copy to another machine using Xmodem. Other operations, such as **format** and **erase**, are only supported by certain file systems on certain platforms.

The following sections describe the use of for using URLs in commands:

File Systems Supporting a Command

Use the context-sensitive help to determine which file systems can be used for a particular command. In the following example, the context-sensitive help displays which file systems can be used as sources for the **copy** EXEC command. The output will vary based on the platform.

```
Router# copy ?
  /erase      Erase destination file system.
  bootflash:  Copy from bootflash: file system
  flash:      Copy from flash: file system
  ftp:        Copy from ftp: file system
  null:       Copy from null: file system
  nvram:      Copy from nvram: file system
  rcp:        Copy from rcp: file system
  system:     Copy from system: file system
  tftp:       Copy from tftp: file system
```

Default File System

For most commands, if no file system is specified, the file is assumed to be in the default directory, as specified by the **cd** command.

```
Router# pwd
slot0:
Router# dir
Directory of slot0:/

  1  -rw-     4720148   Aug 29 1997 17:49:36  hampton/nitro/c7200-j-mz
  2  -rw-     4767328   Oct 01 1997 18:42:53  c7200-js-mz
  5  -rw-         639   Oct 02 1997 12:09:32  foo
  7  -rw-         639   Oct 02 1997 12:37:13  the_time

20578304 bytes total (3104544 bytes free)
Router# cd nvram:
Router# dir
Directory of nvram:/

  1  -rw-        2725              <no date>  startup-config
  2  ----           0              <no date>  private-config
  3  -rw-        2725              <no date>  underlying-config

129016 bytes total (126291 bytes free)
```

Tab Completion

You can use tab completion to reduce the number of characters you need to type for a command. Type the first few characters of the filename, and press the Tab key. If the characters are unique to a filename, the router will complete the filename for you. Continue entering the command as normal and press Return to execute the command.

In the following example, the router completes the filename startup-config because it is the only file in the nvram: file system that starts with "s":

```
Router# show file info nvram:s<tab>
Router# show file info nvram:startup-config<Enter>
```

If you use tab completion without specifying any characters, the router uses the first file in the file system.

```
Router# show file info nvram:<tab>
Router# show file info nvram:private-config<Enter>
```

List of Files in a File System

For many commands, you can get a listing of the files in a file system on the router by using the context-sensitive help. In the following example, the router lists the files in NVRAM:

```
Router# show file info nvram:?
nvram:private-config  nvram:startup-config  nvram:underlying-config
```

Remote File System Management

On remote file systems (file systems on FTP, rcp, or TFTP servers) you can perform the following tasks:

- View the contents of a file with the **more** EXEC command.
- Copy files to or from the router using the **copy** EXEC command.
- Display information about a file using the **showfileinformation** EXEC command.

Note You cannot delete files on remote systems.

NVRAM File System Management

On most platforms, NVRAM contains the startup configuration. On Class A Flash file system platforms, the CONFIG_FILE environment variable specifies the location of the startup configuration. However, the file URL nvram:startup-config always specifies the startup configuration, regardless of the CONFIG_FILE environment variable.

You can display the startup-config (with the **morenvram:startup-config** EXEC command), replace the startup config with a new configuration file (with the **copy***source-url***nvram:startup-config** EXEC command), save the startup configuration to another location (with the **copynvram:startup-config***destination-url* EXEC

command), and erase the contents of NVRAM (with the **erasenvram:**EXECcommand). The **erasenvram:** command also deletes the startup configuration if another location is specified by the CONFIG_FILE variable.

System File System Management

The "system" file system contains the system memory and the current running configuration. You can display the current configuration (with the **showrunning-config**or **moresystem:running-config** EXEC command), save the current configuration to another location (with the **copysystem:running-config***destination-url* EXEC command), and add configuration commands to the current configuration (with the **copy***source-url***system:running-config** EXEC command).

Flash Memory File System Types

Cisco platforms use one of the following three different Flash memory file system types:

- Class A Flash File Systems
- Managing Files on Class B Flash File Systems
- Managing Files on Class C Flash File Systems

The methods used for erasing, deleting, and recovering files depend on the class of the Flash file system. Some commands are supported on only one or two file system types. The command reference documentation notes commands that are not supported on all file system types.

See the table below to determine which Flash memory file system type your platform uses.

Table 3: Flash Memory File System Types

Type	Platforms
Class A	Cisco 7000 series (including the Cisco 7500 series), Cisco 12000 Gigabit Switch Router (GSR), LS1010
Class B	Cisco 1003, Cisco 1004, Cisco 1005, Cisco 2500 series, Cisco 3600 series, Cisco 4000 series, Cisco AS5200
Class C	Cisco MC3810, disk0 of SC3640

Class A Flash File Systems

On Class A Flash file systems, you can delete individual files using the **delete** EXEC command and later recover these files with the **undelete** EXEC command. The **delete** command marks the files as "deleted," but the files still take up space in Flash memory. To permanently delete the files, use the **squeeze** EXEC command. The **squeeze** command removes all of the files marked "deleted" from the specified Flash memory device. These files can no longer be recovered. To erase all of the files on a Flash device, use the **format** EXEC command.

Class B Flash File Systems

On Class B Flash file systems, you can delete individual files with the **delete** EXEC command. The **delete** command marks the file as "deleted." The file is still present in Flash memory and takes up space. To recover the file, use the **undelete** EXEC command. To reclaim any space in Flash memory, you must erase the entire Flash file system with the **erase** EXEC command.

Class C Flash File Systems

On Class C Flash memory file systems, you can delete individual files with the **delete** EXEC command. Files cannot be reclaimed once they have been deleted. Instead, the Flash file system space is reclaimed dynamically. To erase all of the files in Flash, use the **format** EXEC command.

How to Manage Cisco IOS Integrated File Systems

Listing Available File Systems

Not all file systems are supported on every platform. To list the file systems available on your platform, complete the task in this section:

Command	Purpose
show file systems `Router> show file systems`	Lists the file systems available on your platform. This command also displays information about each file system.

Setting the Default File System

To set a default file system, complete the task in this section:

Command	Purpose
cd *filesystem* **:** `Router> cd slot0:`	Sets a default Flash memory device. **Note** You can specify the file system or directory that the system uses as the default file system. Setting the default file system allows you to omit an optional *filesystem***:** argument from related commands. For all EXEC commands that have an optional *filesystem***:** argument, the system uses the file system specified by the **cd** EXEC command when you omit the optional *filesystem***:** argument. For example, the **dir**EXECcommand contains an optional *filesystem***:** argument and displays a list of files on the file system.

Examples

The following example sets the default file system to the Flash memory card inserted in slot 0:

```
cd slot0:
```

Displaying the Current Default File System

To display the current default file system, as specified by the **cd** EXEC command, complete the task in this section:

Command	Purpose
pwd `Router> pwd`	Displays the current file system.

Examples

The following example shows that the default file system is slot 0:

```
Router> pwd
slot0:
```

The following example uses the **cd** command to change the default file system to system and then uses the **pwd** command to verify that the default file system was changed:

```
Router> cd system:

Router> pwd
system:
```

Displaying Information About Files on a File System

To display information about files on a file system, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **dir** [**/all**] [*filesystem***:**][*filename*
3. **show file systems**
4. **show file information** *file-url*
5. **show file descriptors**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**dir** [**/all**] [*filesystem***:**][*filename* **Example:** `Router# dir /all`	Displays a list of files on a file system.
Step 3	**show file systems** **Example:** `Router# show file system`	Displays detailed information about each of the files on a file system.
Step 4	**show file information** *file-url* **Example:** `Router# show file system 10.1.1.1`	Displays information about a specific file.

	Command or Action	Purpose
Step 5	**show file descriptors** **Example:** `Router# show file descriptors`	Displays a list of open file descriptors.

Displaying a File

Examples

The following example compares the different commands used to display information about files for the PCMCIA card in the first slot. Notice that deleted files appear in the **dir/all** command output but not in the **dir** command output.

```
Router# dir slot0:
Directory of slot0:/

  1  -rw-     4720148   Aug 29 1997 17:49:36  hampton/nitro/c7200-j-mz
  2  -rw-     4767328   Oct 01 1997 18:42:53  c7200-js-mz
  5  -rw-         639   Oct 02 1997 12:09:32  foo
  7  -rw-         639   Oct 02 1997 12:37:13  the_time

20578304 bytes total (3104544 bytes free)
Router# dir /all slot0:
Directory of slot0:/

  1  -rw-     4720148   Aug 29 1997 17:49:36  hampton/nitro/c7200-j-mz
  2  -rw-     4767328   Oct 01 1997 18:42:53  c7200-js-mz
  3  -rw-     7982828   Oct 01 1997 18:48:14  [rsp-jsv-mz]
  4  -rw-         639   Oct 02 1997 12:09:17  [the_time]
  5  -rw-         639   Oct 02 1997 12:09:32  foo
  6  -rw-         639   Oct 02 1997 12:37:01  [the_time]
  7  -rw-         639   Oct 02 1997 12:37:13  the_time

20578304 bytes total (3104544 bytes free)
Router# show slot0:
-#- ED --type-- --crc--- -seek-- nlen -length- -----date/time------ name
1   .. unknown  317FBA1B  4A0694   24  4720148 Aug 29 1997 17:49:36 hampton/nitz
2   .. unknown  9237F3FF  92C574   11  4767328 Oct 01 1997 18:42:53 c7200-js-mz
3   .D unknown  71AB01F1 10C94E0   10  7982828 Oct 01 1997 18:48:14 rsp-jsv-mz
4   .D unknown  96DACD45 10C97E0    8      639 Oct 02 1997 12:09:17 the_time
5   .. unknown  96DACD45 10C9AE0    3      639 Oct 02 1997 12:09:32 foo
6   .D unknown  96DACD45 10C9DE0    8      639 Oct 02 1997 12:37:01 the_time
7   .. unknown  96DACD45 10CA0E0    8      639 Oct 02 1997 12:37:13 the_time

3104544 bytes available (17473760 bytes used)
```

Troubleshooting Tips

You can display a list of the contents of a file system before manipulating its contents. For example, before copying a new configuration file to Flash memory, you may want to verify that the file system does not already contain a configuration file with the same name. Similarly, before copying a Flash configuration file to another location, you may want to verify its filename for use in another command.

Displaying a File

To display the contents of any readable file, including a file on a remote file system, complete the task in this section:

SUMMARY STEPS

1. **enable**
2. **more** [**ascii** | **binary** | **ebcdic**| **tftp**] *file-location*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**more** [**ascii**	**binary**	**ebcdic**| **tftp**] *file-location* **Example:** `Router# more tftp://serverA/hampton/savedconfig`	Displays the specified file.

Examples

The following example displays the contents of a configuration file on a TFTP server:

```
Router# more tftp://serverA/hampton/savedconfig

!
! Saved configuration on server
!
version 11.3
service timestamps log datetime localtime
service linenumber
service udp-small-servers
service pt-vty-logging
!
end
```

Managing Files on a Class A Flash File Systems

Deleting Files on a Flash Memory Device

To delete a file from a specified Flash memory device, complete the task in this section:

SUMMARY STEPS

1. **enable**
2. **delete** [*device***:**]*filename*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**delete** [*device***:**]*filename* **Example:** `Router# delete slot0:myconfig`	Deletes a file from a Flash memory device. **Note** When you no longer need a file on a Flash memory device, you can delete it. When you delete a file, the router simply marks the file as deleted, but it does not erase the file. This feature allows you to recover a deleted file, as discussed in the following section. You may want to recover a "deleted" image or configuration file if the new image or configuration file becomes corrupted.

Recovering Deleted Files on a Flash Memory Device

Examples

The following example deletes the file named myconfig from a Flash memory card inserted in slot 0:

```
delete slot0:myconfig
```

Troubleshooting Tips

If you omit the device, the router uses the default device specified by the **cd** EXEC command.

If you attempt to delete the file specified by the CONFIG_FILE or BOOTLDR environment variable, the system prompts you to confirm the deletion. Also, if you attempt to delete the last valid system image specified in the BOOT environment variable, the system prompts you to confirm the deletion.

Recovering Deleted Files on a Flash Memory Device

You can undelete a deleted file. For example, you may want to revert to a previous configuration file because the current one is corrupt.To undelete a deleted file on a Flash memory device, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **dir /all** [*filesystem***:**]
3. **undelete** *index* [*filesystem***:**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**dir /all** [*filesystem***:**] **Example:** `Router# dir /all`	Determines the index of the deleted file.
Step 3	**undelete** *index* [*filesystem***:**] **Example:** `Router# undelete 1 slot 0:`	Restores a deleted file on a Flash memory device.

Examples

The following example recovers the deleted file whose index number is 1 to the Flash memory card inserted in slot 0:

```
undelete 1 slot0:
```

Troubleshooting Tips

You must undelete a file by its index because you can have multiple deleted files with the same name. For example, the "deleted" list could contain multiple configuration files with the name router-config. You undelete by index to indicate which of the many router-config files from the list to undelete. Use the **dir** command with the **/all** option to learn the index number of the file you want to undelete.

You cannot undelete a file if a valid file with the same name exists. Instead, first delete the existing file and then undelete the file you want. For example, if you had a file with the name router-config and you wanted to use a file with the same name that you had previously deleted, you cannot simply undelete the previous version by index. You must first delete the existing router-config file and then undelete the previous router-config file by index. You can undelete a file as long as the file has not been permanently erased with the **squeeze** EXEC command. You can delete and undelete a file up to 15 times.

Permanently Deleting Files on a Flash Memory Device

When a Flash memory device is full, you may need to rearrange the files so that the space used by the deleted files can be reclaimed. To determine whether a Flash memory device is full, use the **dir**EXECcommand. To permanently delete files on a Flash memory device, complete the task in this section:

SUMMARY STEPS

1. **enable**
2. **squeeze** *filesystem* **:**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**squeeze** *filesystem* **:** **Example:** `Router# squeeze slot1:`	Permanently deletes all files marked "deleted" on a Flash memory device. **Note** On Cisco 2600 and 3600 series routers,the entire flash file system needs to be erased once before the **squeeze** command can be used. After being erased once, the **squeeze** command should operate properly on the flash file system for the rest of the flash file system's history.

Troubleshooting Tips

When you issue the **squeeze** command, the router copies all valid files to the beginning of Flash memory and erases all files marked "deleted." At this point, you cannot recover deleted files, and you can now write to the reclaimed Flash memory space.

Note The squeeze operation can take as long as several minutes because it can involve erasing and rewriting almost an entire Flash memory space.

Permanently Deleting Files on a Cisco 2600 or 3600 Router

To erase an entire flash file system on a Cisco 2600 or 3600 series router, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **no partition** *flash-filesystem:*
3. **erase** *filesystem* **:**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**no partition** *flash-filesystem:* **Example:** `Router# no partition flash-filesystem:`	Removes all partitions on the specified flash file system. **Note** The reason for removing partitions is to ensure that the entire flash file system is erased. The **squeeze** command can be used in a flash file system with partitions after the flash file system is erased once.
Step 3	**erase** *filesystem* **:** **Example:** `Router# erase slot1:`	Erases all of the file on the specified flash file system.

Examples

In the following example, the image named c7200-js-mz is deleted and undeleted. Note that the deleted file does not appear in the output for the first **dir** EXEC command, but it appears in the output for the **dir/all** EXEC command.

```
Router# delete slot1:
Delete filename []? c7200-js-mz
Delete slot1:c7200-js-mz? [confirm]
Router# dir slot1:
Directory of slot1:/

No such file

20578304 bytes total (15754684 bytes free)
Router# dir /all slot1:
Directory of slot1:/

  1  -rw-     4823492   Dec 17 1997 13:21:53  [c7200-js-mz]

20578304 bytes total (15754684 bytes free)
Router# undelete 1 slot1:
Router# dir slot1:
Directory of slot1:/

  1  -rw-     4823492   Dec 17 1997 13:21:53 c7200-js-mz
```

```
20578304 bytes total (15754684 bytes free)
```

In the following example, the image is deleted. In order to reclaim the space taken up by the deleted file, the **squeeze** EXEC command is issued.

```
Router# delete slot1:c7200-js-mz
Delete filename [c7200-js-mz]?
Delete slot1:c7200-js-mz? [confirm]
Router# squeeze slot1:
All deleted files will be removed. Continue? [confirm]
Squeeze operation may take a while. Continue? [confirm]
Erasing squeeze log
Squeeze of slot1: complete
Router# dir /all slot1:
Directory of slot1:/
No such file
20578304 bytes total (20578304 bytes free)
```

Troubleshooting Tips

To recompute and verify the checksum of a file in Flash memory on a Class A Flash file system, use the **verify** EXEC command.

Managing Files on Class B Flash File Systems

Deleting Files on a Flash Memory Device

When you no longer need a file on a Flash memory device, you can delete it. When you delete a file, the router simply marks the file as deleted, but it does not erase the file. This feature allows you to recover a deleted file, as discussed in the following section. You may want to recover a "deleted" image or configuration file if the new image or configuration file becomes corrupted. To delete a file from a specified Flash memory device, complete the task in this section:

SUMMARY STEPS

1. **enable**
2. **delete** [*device***:**]*filename*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**delete** [*device***:**]*filename*	Deletes a file from a Flash memory device.

	Command or Action	Purpose
	Example: `Router# delete slot0:myconfig`	**Note** If you omit the device, the router uses the default device specified by the **cd** EXEC command. The following example deletes the file named myconfig from a Flash memory card inserted in slot 0: **deleteslot0:myconfig**

Recovering Deleted Files on a Flash Memory Device

You can undelete a deleted file. For example, you may want to revert to a previous configuration file because the current one is corrupt. To undelete a deleted file on a Flash memory device, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **dir /all** [*filesystem***:**]
3. **undelete** *index* [*filesystem***:**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**dir /all** [*filesystem***:**] **Example:** `Router# dir /all`	Determines the index of the deleted file.
Step 3	**undelete** *index* [*filesystem***:**] **Example:** `Router# undelete 1 slot 0:`	Undeletes a deleted file on a Flash memory device.

Examples

The following example recovers the deleted file whose index number is 1 to the Flash memory card inserted in slot 0:

```
undelete 1 slot0:
```

Troubleshooting Tips

You must undelete a file by its index because you can have multiple deleted files with the same name. For example, the "deleted" list could contain multiple configuration files with the name router-config. You undelete by index to indicate which of the many router-config files from the list to undelete. Use the **dir** command with the **/all** option to learn the index number of the file you want to undelete.

You cannot undelete a file if a valid (undeleted) one with the same name exists. Instead, first delete the existing file and then undelete the file you want. For example, if you had an undeleted version of the router-config file and you wanted to use a previous, deleted version instead, you cannot simply undelete the previous version by index. You must first delete the existing router-config file and then undelete the previous router-config file by index. You can undelete a file as long as the file system has not been permanently erased with the **erase** EXEC command. You can delete and undelete a file up to 15 times.

Erasing Flash Memory

To erase a Flash memory device, use the following command in EXEC mode:

SUMMARY STEPS

1. **enable**
2. **erase** *filesystem* **:**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**erase** *filesystem* **:** **Example:** `Router# erase flash:2`	Erases the Flash file system. **Note** In order to reclaim any space taken up by files in Flash memory, you must erase the entire file system using the **eraseflash:** or **erasebootflash:** EXEC command. These commands reclaim all of the space in Flash memory, erasing all files, deleted or not, in the process. Once erased, these files cannot be recovered. Before erasing Flash memory, save any files you want to keep in another location (an FTP server, for example). Copy the files back to Flash memory after you have erased the device.

Examples

The following example erases all files in the second partition in Flash memory:

```
Router# erase flash:2
System flash directory, partition 2:
File  Length   Name/status
  1   1711088  dirt/gate/c1600-i-mz
[1711152 bytes used, 15066064 available, 16777216 total]
Erase flash device, partition 2? [confirm]
Are you sure? [yes/no]: yes
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
```

Troubleshooting Tips

To recompute and verify the checksum of a file in Flash memory on a Class B Flash file system, use the **verify** EXEC command.

Managing Files on Class C Flash File Systems

Deleting Files on a Flash Memory Device

To delete a file from a specified Flash device, complete the task in this section:

SUMMARY STEPS

1. **enable**
2. **delete** [*device*:]*filename*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**delete** [*device*:]*filename* **Example:** `Router# delete slot0:myconfig`	Deletes a file from a Flash memory device. **Note** When you no longer need a file on a Flash memory device, you can delete it. When you delete a file on a Class C file system, the file is deleted permanently. The router reclaims the space dynamically.

Examples

The following example permanently deletes the file named myconfig from a Flash memory card inserted in slot 0:

```
delete slot0:myconfig
```

Troubleshooting Tips

If you omit the device, the router uses the default device specified by the **cd** EXEC command.

If you attempt to delete the file specified by the CONFIG_FILE or BOOTLDR environment variable, the system prompts you to confirm the deletion. Also, if you attempt to delete the last valid system image specified in the BOOT environment variable, the system prompts you to confirm the deletion.

Formatting Flash

To format a Class C Flash file system, complete the task in this section:

SUMMARY STEPS

1. **enable**
2. **format** *filesystem*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**format** *filesystem* **Example:** `Router# format flash:1`	Formats a Flash file system. **Note** If you format a Flash device, all of the files are erased and cannot be recovered.

Troubleshooting Tips

On Class C Flash file systems, you can create a new directory with the **mkdir** EXEC command. To remove a directory from a Flash file system, use the **rmdir** EXEC command.

On Class C Flash file systems, you can rename a file using the **rename**EXECcommand.

On Class C Flash file systems, you can check a file system for damage and repair any problems using the **fsck** EXEC command.

Configuration Examples for Cisco IOS Integrated File System

Example startup and NVRAM configuration

The following example displays the startup configuration:

```
nnm3640-2# more nvram:startup-config
Using 2279 out of 129016 bytes
!
! Last configuration change at 10:57:25 PST Wed Apr 22 1998
! NVRAM config last updated at 10:57:27 PST Wed Apr 22 1998
!
version 11.3
service timestamps log datetime localtime
service linenumber
service udp-small-servers
service pt-vty-logging
...
end
```

The following example displays the contents of the NVRAM file system on a Class A Flash file system platform. The file named startup-config is the current startup configuration file, in physical NVRAM or in Flash memory. If the file is located in a Flash memory file system, this entry is a symbolic link to the actual file. The file named underlying-config is always the NVRAM version of the configuration.

```
Router# dir nvram:
  Directory of nvram:/

    1  -rw-        2703              <no date>  startup-config
    2  ----           5              <no date>  private-config
    3  -rw-        2703              <no date>  underlying-config

  129016 bytes total (126313 bytes free)
```

Example System File System

The following example changes to the “system” file system, displays the contents of the file system, and displays the running configuration:

```
Router# cd ?
  bootflash:  Directory name
  flash:      Directory name
  lex:        Directory name
  modem:      Directory name
  null:       Directory name
  nvram:      Directory name
  system:     Directory name
  vfc:        Directory name
  <cr>
Router# cd system:?
system:memory  system:running-config  system:ucode  system:vfiles
Router# cd system:
Router# dir
Directory of system:/
    6  dr-x           0              <no date>  memory
    1  -rw-        7786   Apr 22 2001 03:41:39  running-config
No space information available
nnm3640-2# more system:running-config
!
! No configuration change since last restart
```

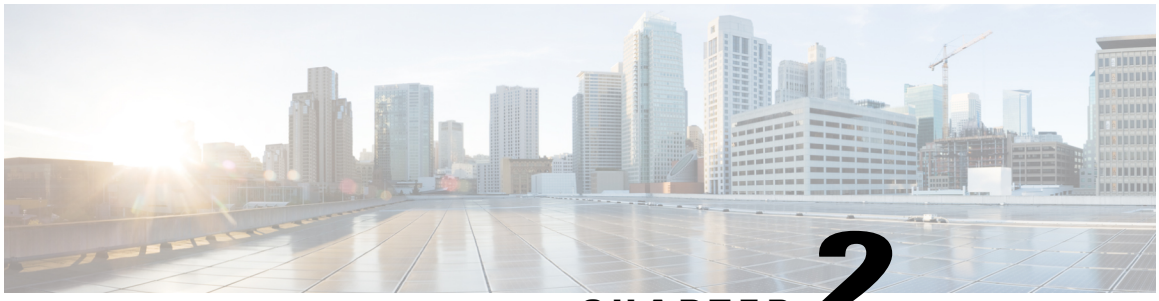
```
!
version 12.2
service timestamps log datetime localtime
service linenumber
service udp-small-servers
service pt-vty-logging
!
.
.
.
end
```

On some platforms, the system file system contains microcode in its ucode directory, as follows:

```
Router# dir system:/ucode
Directory of system:/ucode/

 21  -r--       22900             <no date>  aip20-13
 18  -r--       32724             <no date>  eip20-3
 25  -r--      123130             <no date>  feip20-6
 19  -r--       25610             <no date>  fip20-1
 22  -r--        7742             <no date>  fsip20-7
 23  -r--       17130             <no date>  hip20-1
 24  -r--       36450             <no date>  mip22-2
 29  -r--      154752             <no date>  posip20-0
 28  -r--      704688             <no date>  rsp220-0
 20  -r--       33529             <no date>  trip20-1
 26  -r--      939130             <no date>  vip22-20
 27  -r--     1107862             <no date>  vip222-20

No space information available
```



CHAPTER 2

File System Check and Repair for PCMCIA ATA Disks

The File System Check and Repair for PCMCIA ATA Disks feature introduces a File-System-Check (fsck) utility in Cisco IOS software for File Allocation Table (FAT) filesystems on (Personal Computer Memory Card International Association) PCMCIA disks. The utility performs functions such as checking the boot sector and partition table, checking the file and directory structure, reclaiming unused disk space, and updating the FAT file structure.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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.

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.

Information About File System Check and Repair for PCMCIA ATA Disks

File System Check and Repair for PCMCIA ATA Disks Overview

Prior to the introduction of the file system check (fsck) utility in Cisco IOS Release 12.2(13)T, corrupt files could not be removed from Advanced Technology Attachment (ATA) disks using the Cisco IOS command-line interface (CLI).

Files (or file metadata) in an ATA disk can be corrupted by a variety of events, from power failures or system crashes to simple TFTP copy failures. Prior to the introduction of the file system check (fsck) utility, corrupted files could not be deleted from a usable ATA disk without removing, reformatting, and reinstalling the disk.

The**fsck** privileged EXEC command allows you to conveniently recover wasted disk space directly from the CLI.

Note A FAT16 formatted disk can have only 512 root directory entries. This limits the maximum number of files stored under the root directory. The number of root directory entries stored by a file is in proportion to the filename length. A FAT32 formatted disk does not have this root directory entry limitation. A subdirectory of a FAT16 or FAT32 formatted disk also does not have any limitation on the maximum number of files stored in it.

How to Use the File System Check and Repair for PCMCIA ATA Disks

The fsck utility is enabled by default. No configuration is necessary. For more information, see the **fsck** command page.

Additional References

The following sections provide references related to the File System Check and Repair for PCMCIA ATA Disks feature.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Configuration fundamental commands	*Cisco IOS Configuration Fundamentals Command Reference*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

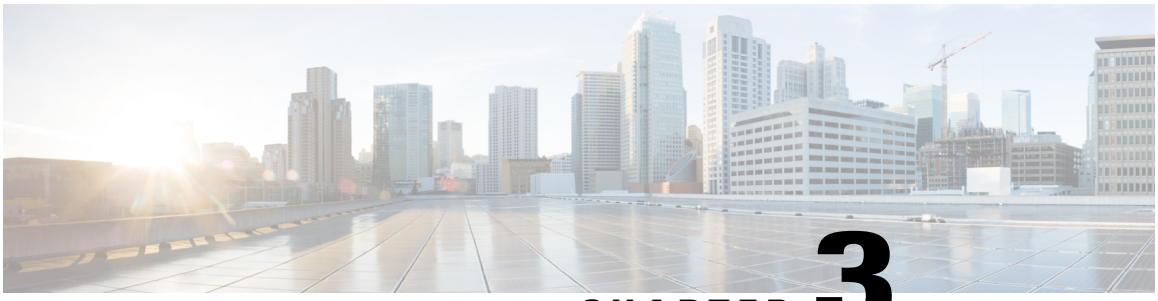
Feature Information for File System Check and Repair for PCMCIA ATA Disks

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 . An account on Cisco.com is not required.

Table 4: Feature Information for File System Check and Repair for PCMCIA ATA Disks

Feature Name	Releases	Feature Information
File System Check and Repair for PCMCIA ATA Disks	12.0(22)S 12.2(13)T	This feature introduces a File-System-Check (fsck) utility in Cisco IOS software for FAT filesystems on PCMCIA disks. The utility performs functions such as checking the boot sector and partition table, checking the file and directory structure, reclaiming unused disk space, and updating the FAT file structure. The following command was introduced or modified: **fsck**.



CHAPTER 3

Storing Data In USB

The Universal Serial Bus (USB) Storage feature enables certain models of Cisco routers to support USB flash modules and with SmartCard technology (which is owned by Aladdin Knowledge Systems) in a USB key form factor (also referred to as a USB eToken) to provide secure access to a router.

USB eTokens provides secure configuration distribution and allows users to store Virtual Private Network (VPN) credentials for deployment. USB flash drives allow users to store images and configurations external to the router.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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.

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.

Prerequisites for Storing Data In USB

Before you can use a USB Flash module or an eToken, you should have the following system requirements:

- A Cisco 871 router, Cisco 1800 series, Cisco 2800 series, or a Cisco 3800 series router.
- At least a Cisco IOS Release 12.3(14)T image running on any of the supported platforms.
- A Cisco supported USB flash or USB eToken.
- A k9 image is required for USB eToken support. (However, USB flash support is available in all images.)

Restrictions for Storing Data In USB

- USB eToken support requires a 3DES (k9) Cisco IOS software image, which provides secure file storage.
- USB hubs are currently not supported. Thus, the number of supported devices is limited to the number of available USB ports on the router chassis.
- You cannot boot an image from an eToken or a USB flash. (However, you can boot a configuration from both an eToken and flash.)

Information About Storing Data In USB

To use a USB flash module and a secure eToken on your router, you should understand the following concepts:

Roles of the USB eToken and the USB Flash

Both USB eTokens and USB flash modules can be used to store files (such as router configurations). The following sections discuss how each device functions and describe the differences between each device:

How a USB eToken Works

A SmartCard is a small plastic card, containing a microprocessor and memory that allows you to store and process data. A SmartCard eToken is a SmartCard with a USB interface. The eToken can securely store any type of file within its available storage space (32KB). Configuration files that are stored on the eToken can be encrypted and accessed only via a user PIN. The router will not load the configuration file unless the proper PIN has been configured for secure deployment of router configuration files.

After you plug the eToken into the router, you must log into the eToken; thereafter, you can change default settings, such as the user PIN (default: 1234567890) and the allowed number of failed login attempts before future logins are refused (default: 15 attempts). For more information on accessing and configuring the eToken, see the section “Accessing and Setting Up the eToken.”

After you have successfully logged into the eToken, you can copy files from the router on to the eToken via the **copy** command. By default, after the eToken is removed from the router, all associated RSA keys are removed; IPSec tunnels are not torn down until the next Internet Key Exchange (IKE) negotiation period. (To change the default behavior and configure a specified length of time before the IPSec tunnels are torn down, issue the **cryptopkitokenremovaltimeout** command.)

How a USB Flash Works

A Cisco USB flash module allows you to store and deploy router configurations and Cisco IOS software images. Cisco USB flash modules are available in 64MB, 128 MB, and 256MB versions.

Note The USB flash is not a replacement for the router compact flash, which must be present for the router to boot.

After you plug the USB flash module into the router, the router will automatically begin to boot the configuration file if the start-up configuration contains the **bootconfig** command to specify the new configuration located on the USB flash device; for example **bootconfigusbflash0:new-config**

Functionality Differences Between an eToken and a USB Flash

Both eTokens and USB flash provide users with secondary storage; however, each device has its own benefits and limitations. To help determine which device better suits your needs, the table below highlights the functionality differences between the eToken and the USB flash.

Table 5: Functionality Differences Between an eToken and a USB Flash

Function	USB eToken	USB Flash
Accessibility	Used to securely store and transfer digital certificates, preshared keys, and router configurations from the eToken to the router.	Used to store and deploy router configurations and images from the USB Flash to the router.
Storage Size	32KB	• 64MB • 128MB • 256MB
File Types	• Typically used to store digital certificates, preshared keys, and router configurations for IPSec VPNs. • eTokens cannot store Cisco IOS images.	Stores a file type that might be stored on a compact flash.
Security	• Files can be encrypted and accessed only with a user PIN. • Files can also be stored in a nonsecure format.	Files can be stored only in a nonsecure format.

Function	USB eToken	USB Flash
Boot Configurations	• The router can use the configuration stored in the eToken during boot time • The router can use the secondary configuration stored in the eToken during boot time. (A secondary configuration allows users to load their IPSec configuration.)	• Configuration file can be automatically transferred from the USB Flash to the router if the **bootconfig** command is issued (for example, **bootconfigusbflash0:new-config**).

USB Storage Filesystem Support

Since USB storage device capacities are increasing, it is necessary that the DOSFS and the usbflash components are modified so that large capacity USB storage devices can be used. The USB Storage Filesystem Support feature extends DOSFS support for USB flash devices. With this feature you can use large capacity USB storage devices for data storage.

Benefits of Storing Data In USB

USB flash drive and USB eToken support on a Cisco router provides the following application benefits:

Removable Credentials: Provide or Store VPN Credentials on an External Device for Deployment

An Aladdin eToken can use SmartCard technology to store a digital certificate and configuration for IPSec VPN deployment. This ability enhances the capability of the router to generate RSA public keys to authenticate at least one IPSec tunnel. (Because a router can initiate multiple IPSec tunnels, the eToken can contain several certificates, as appropriate.)

Storing VPN credentials on an external device reduces the threat of compromising secure data.

PIN Configuration for Secure File Deployment

An Aladdin eToken can store a configuration file that can be used for enabling encryption on the router via a user-configured PIN. (That is, no digital certificates, preshared keys, or VPNs are used.)

Touchless or Low Touch Configuration

Both the eToken and USB Flash can provide remote software configuration and provisioning with little or no human interaction. Configuration is set up as an automated process. That is, both devices can store a bootstrap configuration that the router can use to boot from after the eToken or USB Flash has been inserted into the router. The bootstrap configuration connects the router to a TFTP server, which contains a configuration that completely configures the router.

Login Methods for the eToken

AutomaticLogin

Automatic login allows the router to completely come back up without any user or operator intervention. The PIN is stored in the private configuration, so it is not visible in the startup or running configuration.

Note A hand-generated startup configuration can contain the automatic login command for deployment purposes, but the **copysystem:running-configngvram:startup-config** command must be issued to put the hand-generated configuration in the private configuration.

Manual Login

Manual login can be used when storing a PIN on the router is not desirable. Manual login can be executed with or without privileges, and it will make files and RSA keys on the eToken available to the Cisco IOS software. If a secondary configuration file is configured, it will only be executed with the privileges of the user who is performing the login. Thus, if you want to use manual login and set up the secondary configuration on the eToken to perform anything useful, you need to enable privileges.

Manual login can also be used in recovery scenarios for which the router configuration has been lost. If the scenario contains a remote site that normally connects to the core network with a VPN, the loss of the configuration and RSA keys requires out-of-band services that the eToken can provide. The eToken can contain a boot configuration, a secondary configuration, or both, and RSA keys to authenticate the connection.

Manual login may also be suitable for some initial deployment or hardware replacement scenarios for which the router is obtained from the local supplier or drop-shipped to the remote site.

Unlike automatic login, manual login requires the user to know the actual token PIN. The Aladdin's Windows-based utilities can be used to copy the RSA keys and secondary config files from the eToken if the user has physical access to the eToken.

How to Set Up and Use USB Modules on Cisco Routers

This section contains the following procedures that allow you to configure a router to support USB modules:

Storing the Configuration on an External USB Flash Drive or eToken

To store the configuration file in the USB flash drive module or in an eToken, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **boot config** *file-system-prefix* **:** [*directory*/]*filename* [**nvbypass**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**boot config** *file-system-prefix* **:** [*directory*/]*filename* [**nvbypass**] **Example:** `Router(config)# boot config usbflash0:`	Specifies that the startup configuration file is stored in a USB Flash drive or secure eToken. **Note** If a USB flash drive is used, the router will boot a boot helper from **flash:**. The boot helper is a Cisco IOS image that resides in **flash:**. The Cisco IOS image that is used must be USB-aware.

Accessing and Setting Up the eToken

Logging Into the eToken

To log into an eToken manually or automatically, complete the tasks in this section.

SUMMARY STEPS

1. **enable**
2. Do one of the following:
   - crypto pki token token-name [admin] login [pin]
3. crypto pki token token-name user-pin [pin]
4. **exit**
5. **show usbtoken 0-9 :** *filename*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	Do one of the following: • crypto pki token token-name [admin] login [pin] **Example:** `Router# crypto pki token usbtoken0 admin login 5678` **Example:** **configure terminal** **Example:** `Router# configure terminal`	Manually logs into the eToken. You must specify the **admin** keyword if later you want to change the user PIN. or Puts the router in global configuration mode, which allows you to configure automatic eToken login.
Step 3	crypto pki token token-name user-pin [pin] **Example:** `Router(config)# crypto pki token usbtoken0 user-pin 1234`	(Optional) Creates a PIN that automatically allows the router to log into the USB eToken at router startup. **Note** Do not issue this command if you have already set up manual login.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	**show usbtoken 0-9 :** *filename* **Example:** `Router#`	(Optional) Verifies whether the USB eToken has been logged onto the router.

What to Do Next

- RSA keys are loaded after the eToken is successfully logged into the router.
- By default, newly generated RSA keys are stored on the most recently inserted eToken. Regenerated keys should be stored in the same location that the original RSA key was generated.

Setting Administrative Functions on the eToken

To change default settings, such as the user PIN and the maximum number of failed on the eToken, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **crypto pki token** *token-name* **admin**] **change-pin** [*pin*]
3. **configure terminal**
4. crypto pki token {token-name | default} removal timeout [seconds]
5. crypto pki token {token-name | default} max-retries [number]
6. **exit**
7. copy usbflash [**09:***filenamedestination-url*
8. **show usbtoken 0-9 :** *filename*
9. crypto pki token *token-name* logout

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto pki token** *token-name* **admin**] **change-pin** [*pin*] **Example:** `Router# crypto pki token usbtoken0 admin change-pin`	(Optional) Changes the user PIN number on the USB eToken. • If the PIN is not changed, the default PIN--1234567890--will be used. **Note** After the PIN has been changed, you must reset the login failure count to zero (via the **cryptopkitokenmax-retries** command). The maximum number of allowable login failures is set (by default) to 15.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	crypto pki token {token-name | default} removal timeout [seconds] **Example:** `Router(config)# crypto pki token usbtoken0 removal timeout 60`	(Optional) Sets the time interval, in seconds, that the router will wait before removing the RSA keys that are stored in the eToken after the eToken has been removed from the router. **Note** If this command is not issued, all RSA keys and IPSec tunnels associated with the eToken are torn down immediately after the eToken is removed from the router.

	Command or Action	Purpose
Step 5	crypto pki token {token-name | default} max-retries [number] **Example:** `Router(config)# crypto pki token usbtoken0 max-retries 20`	(Optional) Sets the maximum number of consecutive failed login attempts allowed before access to the eToken is denied. • By default, the value is set at 15.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 7	copy usbflash [**09:***filenamedestination-url* **Example:** `Router# copy usbflash0:`	Copies files from the router to the eToken. • *destination-url* --See the **copy** command page documentation for a list of supported options.
Step 8	**show usbtoken 0-9 :** *filename* **Example:** `Router#`	(Optional) Displays information about the USB eToken. You can use this command to verify whether the USB eToken has been logged onto the router.
Step 9	crypto pki token *token-name* logout **Example:** `Router# crypto pki toke usbtoken0 logout`	Logs the router out of the USB eToken. **Note** If you want to save any data to the USB eToken, you must log back into the eToken.

Troubleshooting USB Flash Drives and eTokens

This section contains descriptions of the following Cisco IOS commands that can be used to help troubleshoot possible problems that may arise while using a USB Flash or a USB eToken:

TheshowfilesystemsCommand

SUMMARY STEPS

1. Use the**showfilesystems**command to determine whether the router recognizes that there is a USB module plugged into a USB port. The USB module should appear on the list of file systems. If the module does not appear on the list, it can indicate any of the following problems:
2. Use the **showfilesystems** command to determine if a USB Flash module is formatted properly. To be compatible with a Cisco router, a USB Flash module must be formatted in a FAT16 format. If that is not the case, the **showfilesystems** command will display an error indicating an incompatible file system.

DETAILED STEPS

Step 1 Use the**showfilesystems**command to determine whether the router recognizes that there is a USB module plugged into a USB port. The USB module should appear on the list of file systems. If the module does not appear on the list, it can indicate any of the following problems:

- A connection problem with the USB module
- The Cisco IOS image running on the router does not support a USB module
- A hardware problem with the USB module itself

Step 2 Use the **showfilesystems** command to determine if a USB Flash module is formatted properly. To be compatible with a Cisco router, a USB Flash module must be formatted in a FAT16 format. If that is not the case, the **showfilesystems** command will display an error indicating an incompatible file system.
Sample output from the **showfilesystems** command showing a USB Flash module and a USB eToken appear below. The USB module listing appears in the last line of the examples.

Example:

```
Router# show file systems
File Systems:
     Size(b)     Free(b)      Type  Flags  Prefixes
           -           -    opaque     rw   archive:
           -           -    opaque     rw   system:
           -           -    opaque     rw   null:
           -           -   network     rw   tftp:
*  129880064    69414912      disk     rw   flash:#
      491512      486395     nvram     rw   nvram:
           -           -    opaque     wo   syslog:
           -           -    opaque     rw   xmodem:
           -           -    opaque     rw   ymodem:
           -           -   network     rw   rcp:
           -           -   network     rw   pram:
           -           -   network     rw   ftp:
           -           -   network     rw   http:
           -           -   network     rw   scp:
           -           -   network     rw   https:
           -           -    opaque     ro   cns:
    63158272    33037312  usbflash     rw   usbflash0:
       32768         858  usbtoken     rw   usbtoken1:
```

The show usb device Command

SUMMARY STEPS

1. Use the **showusbdevice** command to determine if a USB module is supported by Cisco. The sample output for both the USB Flash and the USB eToken that indicates whether or not the module is supported are highlighted in the sample outputs below.

DETAILED STEPS

Use the **showusbdevice** command to determine if a USB module is supported by Cisco. The sample output for both the USB Flash and the USB eToken that indicates whether or not the module is supported are highlighted in the sample outputs below.
The following sample output is for a USB Flash module:

Example:

```
Router# show usb device
Host Controller:1
Address:0x1
Device Configured:YES
Device Supported:YES
Description:DiskOnKey
Manufacturer:M-Sys
Version:2.0
Serial Number:0750D84030316868
Device Handle:0x1000000
USB Version Compliance:2.0
Class Code:0x0
Subclass Code:0x0
Protocol:0x0
Vendor ID:0x8EC
Product ID:0x15
Max. Packet Size of Endpoint Zero:64
Number of Configurations:1
Speed:Full
Selected Configuration:1
Selected Interface:0
Configuration:
    Number:1
    Number of Interfaces:1
    Description:
    Attributes:None
    Max Power:140 mA
    Interface:
        Number:0
        Description:
        Class Code:8
        Subclass:6
        Protocol:80
        Number of Endpoints:2
        Endpoint:
            Number:1
            Transfer Type:BULK
            Transfer Direction:Device to Host
            Max Packet:64
            Interval:0
        Endpoint:
            Number:2
            Transfer Type:BULK
            Transfer Direction:Host to Device
```

```
            Max Packet:64
            Interval:0
```

The following sample output is for a supported USB eToken:

Example:

```
Router# show usb device
Host Controller:1
Address:0x11
Device Configured:YES
Device Supported:YES
Description:eToken Pro 4254
Manufacturer:AKS
Version:1.0
Serial Number:
Device Handle:0x1010000
USB Version Compliance:1.0
Class Code:0xFF
Subclass Code:0x0
Protocol:0x0
Vendor ID:0x529
Product ID:0x514
Max. Packet Size of Endpoint Zero:8
Number of Configurations:1
Speed:Low
Selected Configuration:1
Selected Interface:0
Configuration:
    Number:1
    Number of Interfaces:1
    Description:
    Attributes:None
    Max Power:60 mA
    Interface:
        Number:0
        Description:
        Class Code:255
        Subclass:0
        Protocol:0
        Number of Endpoints:0
```

The show usb controllers Command

SUMMARY STEPS

1. Use the **showusbcontrollers** command to determine if there is a hardware problem with a USB Flash module. If the **showusbcontrollers** command displays an error, it indicates a hardware problem in the USB module.

DETAILED STEPS

Use the **showusbcontrollers** command to determine if there is a hardware problem with a USB Flash module. If the **showusbcontrollers** command displays an error, it indicates a hardware problem in the USB module.

You can also use the **showusbcontrollers** command to verify that copy operations onto a USB Flash module are occurring successfully. Issuing the **showusbcontrollers** command after performing a file copy should display successful data transfers.

Sample output for the**showusbcontrollers** command for a working USB Flash module appears below:

Example:

```
Router# show usb controllers
Name:1362HCD
Controller ID:1
Controller Specific Information:
    Revision:0x11
    Control:0x80
    Command Status:0x0
    Hardware Interrupt Status:0x24
    Hardware Interrupt Enable:0x80000040
    Hardware Interrupt Disable:0x80000040
    Frame Interval:0x27782EDF
    Frame Remaining:0x13C1
    Frame Number:0xDA4C
    LSThreshold:0x628
    RhDescriptorA:0x19000202
    RhDescriptorB:0x0
    RhStatus:0x0
    RhPort1Status:0x100103
    RhPort2Status:0x100303
    Hardware Configuration:0x3029
    DMA Configuration:0x0
    Transfer Counter:0x1
    Interrupt:0x9
    Interrupt Enable:0x196
    Chip ID:0x3630
    Buffer Status:0x0
    Direct Address Length:0x80A00
    ATL Buffer Size:0x600
    ATL Buffer Port:0x0
    ATL Block Size:0x100
    ATL PTD Skip Map:0xFFFFFFFF
    ATL PTD Last:0x20
    ATL Current Active PTD:0x0
    ATL Threshold Count:0x1
    ATL Threshold Timeout:0xFF
Int Level:1
Transfer Completion Codes:
         Success              :920             CRC             :0
         Bit Stuff            :0               Stall           :0
         No Response          :0               Overrun         :0
         Underrun             :0               Other           :0
         Buffer Overrun       :0               Buffer Underrun :0
Transfer Errors:
         Canceled Transfers   :2               Control Timeout :0
Transfer Failures:
         Interrupt Transfer   :0               Bulk Transfer   :0
         Isochronous Transfer :0               Control Transfer:0
Transfer Successes:
         Interrupt Transfer   :0               Bulk Transfer   :26
         Isochronous Transfer :0               Control Transfer:894
USBD Failures:
         Enumeration Failures :0               No Class Driver Found:0
         Power Budget Exceeded:0
USB MSCD SCSI Class Driver Counters:
         Good Status Failures :3               Command Fail    :0
         Good Status Timed out:0               Device not Found:0
         Device Never Opened  :0               Drive Init Fail :0
         Illegal App Handle   :0               Bad API Command :0
         Invalid Unit Number  :0               Invalid Argument:0
         Application Overflow :0               Device in use   :0
```

```
         Control Pipe Stall   :0                Malloc Error    :0
         Device Stalled       :0                Bad Command Code:0
         Device Detached      :0                Unknown Error   :0
         Invalid Logic Unit Num:0
USB Aladdin Token Driver Counters:
         Token Inserted       :1                Token Removed   :0
         Send Insert Msg Fail :0                Response Txns   :434
         Dev Entry Add Fail   :0                Request Txns    :434
         Dev Entry Remove Fail:0                Request Txn Fail:0
         Response Txn Fail    :0                Command Txn Fail:0
         Txn Invalid Dev Handle:0
USB Flash File System Counters:
         Flash Disconnected   :0                Flash Connected :1
         Flash Device Fail    :0                Flash Ok        :1
         Flash startstop Fail :0                Flash FS Fail   :0
USB Secure Token File System Counters:
         Token Inserted       :1                Token Detached  :0
         Token FS success     :1                Token FS Fail   :0
         Token Max Inserted   :0                Create Talker Failures:0
         Token Event          :0                Destroy Talker Failures:0
         Watched Boolean Create Failures:0
```

The dir Command

SUMMARY STEPS

1. Use the **dir** command with the **usbflash09:** or the **usbtoken09:** keyword to display all files, directories, and their permission strings on the USB Flash or USB eToken.

DETAILED STEPS

Use the **dir** command with the **usbflash09:** or the **usbtoken09:** keyword to display all files, directories, and their permission strings on the USB Flash or USB eToken.
The following sample output displays directory information for the USB Flash:

Example:

```
Router# dir usbflash0:
Directory of usbflash0:/
    1  -rw-    30125020  Dec 22 2032 05:31:32 +00:00  c3825-entservicesk9-mz.123-14.T
63158272 bytes total (33033216 bytes free)
```

The following sample output displays directory information for the USB eToken:

Example:

```
Router# dir usbtoken1:
Directory of usbtoken1:/
    2  d---          64  Dec 22 2032 05:23:40 +00:00  1000
    5  d---        4096  Dec 22 2032 05:23:40 +00:00  1001
    8  d---           0  Dec 22 2032 05:23:40 +00:00  1002
   10  d---         512  Dec 22 2032 05:23:42 +00:00  1003
   12  d---           0  Dec 22 2032 05:23:42 +00:00  5000
   13  d---           0  Dec 22 2032 05:23:42 +00:00  6000
```

```
   14  d---           0  Dec 22 2032 05:23:42 +00:00  7000
   15  ----         940  Jun 27 1992 12:50:42 +00:00  mystartup-config
   16  ----        1423  Jun 27 1992 12:51:14 +00:00  myrunning-config
32768 bytes total (858 bytes free)
```

The following sample output displays directory information for all devices the router is aware of:

Example:

```
Router# dir all-filesystems
Directory of archive:/
No files in directory
No space information available
Directory of system:/
    2  drwx           0                    <no date>  its
  115  dr-x           0                    <no date>  lib
  144  dr-x           0                    <no date>  memory
    1  -rw-        1906                    <no date>  running-config
  114  dr-x           0                    <no date>  vfiles
No space information available
Directory of flash:/
    1  -rw-    30125020  Dec 22 2032 03:06:04 +00:00  c3825-entservicesk9-mz.123-14.T
129880064 bytes total (99753984 bytes free)
Directory of nvram:/
  476  -rw-        1947                    <no date>  startup-config
  477  ----          46                    <no date>  private-config
  478  -rw-        1947                    <no date>  underlying-config
    1  -rw-           0                    <no date>  ifIndex-table
    2  ----           4                    <no date>  rf_cold_starts
    3  ----          14                    <no date>  persistent-data
491512 bytes total (486395 bytes free)
Directory of usbflash0:/
    1  -rw-    30125020  Dec 22 2032 05:31:32 +00:00  c3825-entservicesk9-mz.123-14.T
63158272 bytes total (33033216 bytes free)
Directory of usbtoken1:/
    2  d---          64  Dec 22 2032 05:23:40 +00:00  1000
    5  d---        4096  Dec 22 2032 05:23:40 +00:00  1001
    8  d---           0  Dec 22 2032 05:23:40 +00:00  1002
   10  d---         512  Dec 22 2032 05:23:42 +00:00  1003
   12  d---           0  Dec 22 2032 05:23:42 +00:00  5000
   13  d---           0  Dec 22 2032 05:23:42 +00:00  6000
   14  d---           0  Dec 22 2032 05:23:42 +00:00  7000
   15  ----         940  Jun 27 1992 12:50:42 +00:00  mystartup-config
   16  ----        1423  Jun 27 1992 12:51:14 +00:00  myrunning-config
32768 bytes total (858 bytes free)
```

Configuration Examples for Secure Token Support

Example Logging Into and Saving RSA Keys to eToken

The following configuration example shows to how log into the eToken, generate RSA keys, and store the RSA keys onto the eToken:

```
! Configure the router to automatically log into the eToken
configure terminal
 crypto pki token default user-pin 0 1234567890
! Generate RSA keys and enroll certificates with the CA.
crypto pki trustpoint IOSCA
```

```
 enrollment url http://10.23.2.2
 exit
crypto ca authenticate IOSCA
Certificate has the following attributes:
       Fingerprint MD5:23272BD4 37E3D9A4 236F7E1A F534444E
      Fingerprint SHA1:D1B4D9F8 D603249A 793B3CAF 8342E1FE 3934EB7A
% Do you accept this certificate? [yes/no]:yes
Trustpoint CA certificate accepted.
crypto pki enroll
crypto pki enroll IOSCA
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.
Password:
Re-enter password:
% The subject name in the certificate will include:c2851-27.cisco.com
% Include the router serial number in the subject name? [yes/no]:no
% Include an IP address in the subject name? [no]:no
Request certificate from CA? [yes/no]:yes
% Certificate request sent to Certificate Authority
% The 'show crypto ca certificate IOSCA verbose' command will show the fingerprint.
*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint MD5:E6DDAB1B
 0E30EFE6 54529D8A DA787DBA
*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint SHA1:3B0F33B
7 57C02A10 3935042B C4B6CD3D 61039251
*Jan 13 06:47:21.021:%PKI-6-CERTRET:Certificate received from Certificate Authority
! Issue the write memory command, which will automatically save the RSA keys to the eToken
 ! instead of private NVRAM.
Router# write memory
Building configuration...
[OK]
*Jan 13 06:47:29.481:%CRYPTO-6-TOKENSTOREKEY:Key c2851-27.cisco.com stored on
Cryptographic Token eToken Successfully
```

The following sample output from the **showcryptokeymypubkeyrsa** command displays stored credentials after they are successfully load from the eToken. Credentials that are stored on the eToken are in the protected area. When storing the credentials on the eToken, the files are stored in a directory called /keystore. However, the key files are hidden from the CLI.

```
Router#
show crypto key mypubkey rsa
% Key pair was generated at:06:37:26 UTC Jan 13 2005
Key name:c2851-27.cisco.com
 Usage:General Purpose Key
 Key is not exportable.
 Key Data:
  305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00E3C644 43AA7DDD
  732E0F4E 3CA0CDAB 387ABF05 EB8F22F2 2431F1AE 5D51FEE3 FCDEA934 7FBD3603
  7C977854 B8E999BF 7FC93021 7F46ABF8 A4BA2ED6 172D3D09 B5020301 0001
% Key pair was generated at:06:37:27 UTC Jan 13 2005
Key name:c2851-27.cisco.com.server
 Usage:Encryption Key
 Key is not exportable.
 Key Data:
  307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00DD96AE 4BF912EB
  2C261922 4784EF98 2E70E837 774B3778 7F7AEB2D 87F5669B BF5DDFBC F0D521A5
  56AB8FDC 9911968E DE347FB0 A514A856 B30EAFF4 D1F453E1 003CFE65 0CCC6DC7
  21FBE3AC 2F8DEA16 126754BC 1433DEF9 53266D33 E7338C95 BB020301 0001
```

Additional References

The following sections provide references related to the Storing data using USB feature.

Related Documents

Related Topic	Document Title
Connecting the USB modules to the router	*Cisco Access Router USB Flash Module and USB eToken Hardware Installation Guide*
eToken and USB Flash data sheet	USB eToken and USB Flash Features Support
File management (loading, copying, and rebooting files)	The section"File Management" in the *Cisco IOS Configuration Fundamentals and Network Management Configuration Guide*
Configuring digital certificate encryption	The chapter "Configuring Certification Authority Interoperability" in the *Cisco IOS Security Configuration Guide*

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	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

RFCs

RFCs	Title
None	--

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 Storing Data In USB

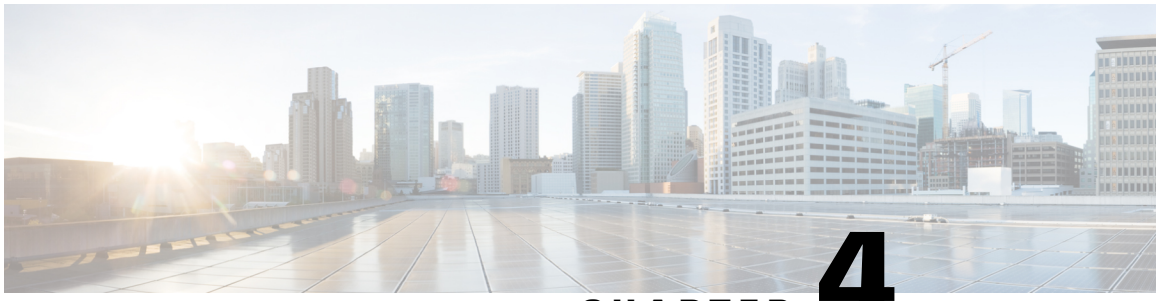
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train 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 . An account on Cisco.com is not required.

Table 6: Feature Information for Storing Data In USB

Feature Name	Releases	Feature Information
USB Storage	12.3(14)T	The USB Storage feature enables certain models of Cisco routers to support USB flash modules and with SmartCard technology (which is owned by Aladdin Knowledge Systems) in a USB key form factor (also referred to as a USB eToken) to provide secure access to a router. The following commands were introduced or modified: **crypto pki token change-pin**, **crypto pki token login**, **crypto pki token logout**, **crypto pki token max-retries**, **crypto pki token removal timeout**, **crypto pki token secondary config**, **crypto pki token user-pin**, **debug usb**, **driver**, **show usb driver**, **show usb controllers**, **show usb device**, **show usb driver**, **show usb port**, **show usbtoken**, **show usb tree**, **boot config**, **copy**, **delete**, **dir**, **format**.

Feature Name	Releases	Feature Information
USB Storage Filesystem Support	12.2(33)SRE	The USB Storage Filesystem Support feature extends DOSFS support for USB flash devices. With this feature you can use large capacity USB storage devices for data storage. In 12.2(33)SRE, this feature was introduced on the Cisco 7200-NPE-G2. The following commands were introduced or modified: **cd**, **verify**, **mkdir**, **fsck**.



CHAPTER 4

Configuring Basic File Transfer Services

Using basic file transfer services, you can configure a router as a Trivial File Transfer Protocol (TFTP) or Reverse Address Resolution Protocol (RARP) server, configure the router to forward extended BOOTP requests over asynchronous interfaces, and configure rcp, rsh, and FTP.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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.

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.

Prerequisites for Basic File Transfer Services

- You should have at least a basic familiarity with the Cisco IOS environment and the command-line interface.
- You should have at least a minimal configuration running on your system.

Restrictions for Basic File Transfer Services

- You must have your network up and running, with Cisco IOS Release 12.2 or a later release installed.

- Some of the Cisco IOS configuration commands are only available on certain router platforms, and the command syntax may vary on different platforms.

Information About Basic File Transfer Services

Use of a Router as a TFTP or RARP Server

It is too costly and inefficient to have a machine that acts only as server on every network segment. However, when you do not have a server on every segment, your network operations can incur substantial time delays across network segments. You can configure a router to serve as a RARP or TFTP server to reduce costs and time delays in your network while allowing you to use your router for its regular functions.

Typically, a router that is configured as a TFTP or RARP server provides other routers with system image or router configuration files from its Flash memory. You can also configure the router to respond to other types of service requests, such as requests.

Use of a Router as a TFTP Server

As a TFTP server host, the router responds to TFTP Read Request messages by sending a copy of the system image contained in ROM or one of the system images contained in Flash memory to the requesting host. The TFTP Read Request message must use one of the filenames that are specified in the configuration.

Note For the Cisco 7000 family, the filename used must represent a software image that is present in Flash memory. If no image resides in Flash memory, the client router will boot the server's ROM image as a default.

Flash memory can be used as a TFTP file server for other routers on the network. This feature allows you to boot a remote router with an image that resides in the Flash server memory.

Some Cisco devices allow you to specify one of the different Flash memory locations (**bootflash:**, **slot0:**, **slot1:**, **slavebootflash:**, **slaveslot0:**, or **slaveslot1:**) as the TFTP server.

Use of a Router as a RARP Server

Reverse Address Resolution Protocol (RARP) is a protocol in the TCP/IP stack that provides a method for finding IP addresses based on MAC (physical) addresses. This functionality is the reverse of broadcasting Address Resolution Protocols (ARPs), through which a host can dynamically discover the MAC-layer address corresponding to a particular IP network-layer address. RARP makes diskless booting of various systems possible (for example, diskless workstations that do not know their IP addresses when they boot, such as Sun workstations or PCs on networks where the client and server are on separate subnets). RARP relies on the presence of a RARP server with cached table entries of MAC-layer-to-IP address mappings.

You can configure a Cisco router as a RARP server. This feature enables the Cisco IOS software to answer RARP requests.

Use of a Router for rsh and rcp

Remote shell (rsh) gives users the ability to execute commands remotely. Remote copy (rcp) allows users to copy files to and from a file system residing on a remote host or server on the network. Cisco's implementation of rsh and rcp interoperates with the industry standard implementations. Cisco uses the abbreviation RCMD (Remote Command) to indicate both rsh and rcp.

Source Interface for Outgoing RCMD Communications

You can specify the source interface for RCMD (rsh and rcp) communications. For example, the router can be configured so that RCMD connections use the loopback interface as the source address of all packets leaving the router. Specifying the source-interface is most commonly used to specify a loopback interface. This allows you to associate a permanent IP address with RCMD communications. Having a permanent IP address is useful for session identification (remote device can consistently idendify the origin of packets for the session). A "well-known" IP address can also be used for security purposes, as you can then create access lists on remote devices which include the address.

About DNS Reverse Lookup for rcmd

As a basic security check, the Cisco IOS software does a reverse lookup of the client IP address using DNS for the remote command (rcmd) applications (rsh and rcp). This check is performed using a host authentication process.

When enabled, the system records the address of the requesting client. That address is mapped to a host name using DNS. Then a DNS request is made for the IP address for that host name. The IP address received is then checked against the original requesting address. If the address does not match with any of the addresses received from DNS, the rcmd request will not be serviced.

This reverse lookup is intended to help protect against "spoofing." However, please note that the process only confirms that the IP address is a valid routable address; it is still possible for a hacker to spoof the valid IP address of a known host.

Implementation of rsh

You can use rsh (remote shell) to execute commands on remote systems to which you have access. When you issue the **rsh** command, a shell is started on the remote system. The shell allows you to execute commands on the remote system without having to log in to the target host.

You do not need to connect to the system, router, or access server and then disconnect after you execute a command if you use rsh. For example, you can use rsh to remotely look at the status of other devices *without* connecting to the target device, executing the command, and then disconnecting. This capability is useful for looking at statistics on many different routers. Configuration commands for enabling rsh use the acronym "rcmd", which is short for "remote command".

Maintaining rsh Security

To gain access to a remote system running rsh, such as a UNIX host, an entry must exist in the system's *.rhosts* file or its equivalent identifying you as a user who is authorized to execute commands remotely on the system. On UNIX systems, the *.rhosts* file identifies users who can remotely execute commands on the system.

You can enable rsh support on a router to allow users on remote systems to execute commands. However, our implementation of rsh does not support an *.rhosts* file. Instead, you must configure a local authentication database to control access to the router by users attempting to execute commands remotely using rsh. A local authentication database is similar to a UNIX *.rhosts* file. Each entry that you configure in the authentication database identifies the local user, the remote host, and the remote user.

Implementation of rcp

The remote copy (rcp) commands rely on the rsh server (or daemon) on the remote system. To copy files using rcp, you do not need to create a server for file distribution, as you do with TFTP. You need only to have access to a server that supports the remote shell (rsh). (Most UNIX systems support rsh.) Because you are copying a file from one place to another, you must have read permission on the source file and write permission in the destination directory. If the destination file does not exist, rcp creates it for you.

Although Cisco's rcp implementation emulates the functions of the UNIX rcp implementation--copying files among systems on the network--Cisco's command syntax differs from the UNIX rcp command syntax. The Cisco IOS software offers a set of copy commands that use rcp as the transport mechanism. These rcp copy commands are similar in style to the Cisco IOS TFTP copy commands, but they offer an alternative that provides faster performance and reliable delivery of data. These improvements are possible because the rcp transport mechanism is built on and uses the Transmission Control Protocol/Internet Protocol (TCP/IP) stack, which is connection-oriented. You can use rcp commands to copy system images and configuration files from the router to a network server and vice versa.

You can also enable rcp support to allow users on remote systems to copy files to and from the router.

If you do not specify the**/user** keyword and argument, the Cisco IOS software sends a default remote username. As the default value of the remote username, the software sends the remote username associated with the current tty process, if that name is valid. If the tty remote username is invalid, the software uses the router host name as the both the remote and local usernames.

Configure the Remote Client to Send rcp Requests

The rcp protocol requires a client to send a remote username on each rcp request to a server. When you copy a configuration file from a server to the router using rcp, the Cisco IOS software sends the first valid username in the following list:

1. The username set by the **iprcmdremote-username** command, if the command is configured.
2. The remote username associated with the current tty (terminal) process. For example, if the user is connected to the router through Telnet and was authenticated through the **username** command, the router software sends the Telnet username as the remote username.

Note In Cisco products, ttys are commonly used in access servers. The concept of tty originated with UNIX. For UNIX systems, each physical device is represented in the file system. Terminals are called *tty devices* , which stands for *teletype* , the original UNIX terminal.

1. The router host name.

For **boot**commands using rcp, the software sends the router host name; you cannot explicitly configure the remote username.

For the rcp copy request to execute successfully, an account must be defined on the network server for the remote username.

If you are writing to the server, the rcp server must be properly configured to accept the rcp write request from the user on the router. For UNIX systems, you must add an entry to the *.rhosts* file for the remote user on the rcp server. For example, if the router contains the following configuration lines.

```
hostname Rtr1
ip rcmd remote-username User0
```

and the router's IP address translates to Router1.company.com, then the *.rhosts* file for User0 on the rcp server should contain the following line:

```
Router1.company.com Rtr1
```

Refer to the documentation for your rcp server for more details.

If the server has a directory structure, the configuration file or image is written or copied relative to the directory associated with the remote username on the server. Use the **iprcmdremote-username** command to specify which directory on the server to use. For example, if the system image resides in the home directory of a user on the server, you can specify that user's name as the remote username.

If you copy the configuration file to a personalcomputer used as a file server, the computer must support rsh.

Use of a Router for FTP Connections

You can configure a router to transfer files between systems on the network using the File Transfer Protocol (FTP). With the Cisco IOS implementation of FTP, you can set the following FTP characteristics:

- Passive-mode FTP
- User name
- Password
- IP address

How to Configure Basic File Transfer Services

Configuring the Router for Use as a TFTP Server

To configure your router for use as a TFTP server, complete the tasks in this section.

Before You Begin

The server and client router must be able to reach each other before the TFTP function can be implemented. Verify this connection by testing the connection between the server and client router (in either direction) using the **ping***a.b.c.d* command (where *a.b.c.d* is the address of the client device). After the **ping** command is issued, connectivity is indicated by a series of exclamation points (!), while a series of periods (.) plus [timed out] or [failed] indicates that the connection attempt failed. If the connection fails, reconfigure the interface, check the physical connection between the Flash server and client router, andping again.

After you verify the connection, ensure that a TFTP-bootable image is present on the server. This is the system software image the client router will boot. Note the name of this software image so you can verify it after the first client boot.

Caution For full functionality, the software image sent to the client must be the same type as the ROM software installed on the client router. For example, if the server has X.25 software, and the client does not have X.25 software in ROM, the client will not have X.25 capabilities after booting from the server's image in Flash memory.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **tftp-server flash** [*partition-number***:**]*filename1* [**alias***filename2*] [*access-list-number*]
   - **tftp-server flash** *device* **:** *filename* (Cisco 7000 family only)
   - **tftp-server flash** [*device***:**][*partition-number***:**]*filename* (Cisco 1600 series and Cisco 3600 series only)
   - **tftp-server rom alias** *filename1* [*access-list-number*]
4. **end**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Do one of the following: • **tftp-server flash** [*partition-number***:**]*filename1* [**alias***filename2*] [*access-list-number*] • **tftp-server flash** *device* **:** *filename* (Cisco 7000 family only)	Specifies the system image to send in response to Read Requests. You can enter multiple lines to specify multiple images.

	Command or Action	Purpose
	• **tftp-server flash** [*device*:][*partition-number*:]*filename* (Cisco 1600 series and Cisco 3600 series only) • **tftp-server rom alias** *filename1* [*access-list-number*] **Example:** `Device(config)# tftp-server flash version-10.3 22`	
Step 4	**end** **Example:** `Device(config)# end`	Ends the configuration session and returns you to privileged EXEC mode.
Step 5	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves the running configuration to the startup configuration file.

Examples

In the following example, the system can use TFTP to send copies of the Flash memory file *version-10.3* in response to a TFTP Read Request for that file. The requesting host is checked against access list 22.

```
tftp-server flash version-10.3 22
```

In the following example, the system can use TFTP to send a copy of the ROM image *gs3-k.101*in response to a TFTP Read Request for the *gs3-k.101* file:

```
tftp-server rom alias gs3-k.101
```

In the following example, a router sends a copy of the file *gs7-k.9.17* in Flash memory in response to a TFTP Read Request. The client router must reside on a network specified by access list 1. Thus, in the example, the any clients on network 172.16.101.0 are permitted access to the file.

```
Server# configure terminal

Enter configuration commands, one per line. End with CTRL/Z
Server(config)# tftp-server flash gs7-k.9.17 1

Server(config)# access-list 1 permit 172.16.101.0 0.0.0.255

Server(config)# end

Server# copy running-config startup-config

[ok]
Server#
```

Troubleshooting

The TFTP session can sometimes fail. TFTP generates the following special characters to help you determine why a TFTP session fails:

- An “E” character indicates that the TFTP server received an erroneous packet.
- An “O” character indicates that the TFTP server received an out-of-sequence packet.
- A period (.) indicates a timeout.

For diagnosing any undue delay in the transfer, the output is useful. For troubleshooting procedures, refer to the *Internetwork Troubleshooting Guide* publication.

Configuring the Client Router

To configure the client router to first load a system image from the server, and as a backup, to configure the client router to load its own ROM image if the load from a server fails, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no boot system**
4. **boot system** [**tftp**] *filename* [*ip-address*]
5. **boot system rom**
6. **config-register** *value*
7. **end**
8. **copy running-config startup-config**
9. **reload**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**no boot system** **Example:** `Device(config)# no boot system`	(Optional) Removes all previous **bootsystem** statements from the configuration file.
Step 4	**boot system** [**tftp**] *filename* [*ip-address*] **Example:** `Device(config)# boot system c5300-js-mz.121-5.T.bin 172.16.1.1`	Specifies that the client router load a system image from the server.
Step 5	**boot system rom** **Example:** `Device(config)# boot system rom`	Specifies that the client router loads its own ROM image if the load from a server fails.
Step 6	**config-register** *value* **Example:** `Device(config)# config-register 0x010F`	Sets the configuration register to enable the client router to load a system image from a network server.
Step 7	**end** **Example:** `Device(config)# end`	Exits global configuration mode.
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves the configuration file to your startup configuration.
Step 9	**reload** **Example:** `Device# reload`	(Optional) Reloads the router to make your changes take effect.

Examples

In the following example, the router is configured to boot from a specified TFTP server:

```
Client# configure terminal

Enter configuration commands, one per line. End with CTRL/Z
Client(config)# no boot system
```

```
Client(config)# boot system c5300-js-mz.121-5.T.bin 172.16.1.1

Client(config)# boot system rom

Client(config)# config-register 0x010F

Client(config)# end

Client# copy running-config startup-config

[ok]
Client# reload
```

In this example, the **nobootsystem** command invalidates all other**bootsystem** commands currently in the configuration memory, and any **bootsystem** commands entered after this command will be executed first. The second command, **bootsystem***filename address* , tells the client router to look for the file c5300-js-mz.121-5.T.bin on the TFTP server with an IP address of 172.16.111.111. Failing this, the client router will boot from its system ROM in response to the **bootsystemrom** command, which is included as a backup in case of a network problem. The **copyrunning-configstartup-config** command copies the configuration to the startup configuration, and the**reload**command boots the system.

Note The system software to be booted from the server must reside in Flash memory on the server. If it is not in Flash memory, the client router will boot the server's system ROM.

The following example shows sample output of the **showversion** command after the router has rebooted:

```
Device> show version
Cisco Internetwork Operating System Software
Cisco IOS (tm) 5300 Software (C5300-JS-M), Version 12.1(5)T,  RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2000 by Cisco Systems, Inc.
Compiled Sat 11-Nov-00 03:03 by joe
Image text-base: 0x60008958, data-base: 0x611C6000
ROM: System Bootstrap, Version 11.2(9)XA, RELEASE SOFTWARE (fc2)
BOOTFLASH: 5300 Software (C5300-BOOT-M), Version 12.0(7)T,  RELEASE SOFTWARE (f)
Router uptime is 8 weeks, 4 days, 22 hours, 36 minutes
System returned to ROM by power-on
System restarted at 00:37:38 UTC Thu Feb 22 2001
System image file is "flash:c5300-js-mz.121-5.T.bin"
.
.
.
Configuration register is 0x010F
```

The important information in this example is contained in the first line “Cisco IOS (tm)..” and in the line that begins “System image file....” The “Cisco IOS (tm)...” line shows the version of the operating system in NVRAM. The “System image file....” line show the filename of the system image loaded from the TFTP server.

What to Do Next

After the system reloads, you should use the **showversion** EXEC mode command to verify that the system booted the desired image.

Caution Using the **nobootsystem** command, as in the following example, will invalidate *all* other boot system commands currently in the client router system configuration. Before proceeding, determine whether the system configuration stored in the client router should first be saved (uploaded) to a TFTP file server so you have a backup copy.

Configuring the Router as a RARP Server

To configure the router as a RARP server, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type* [*slot*/]*port*
4. **ip rarp-server** *ip-address*

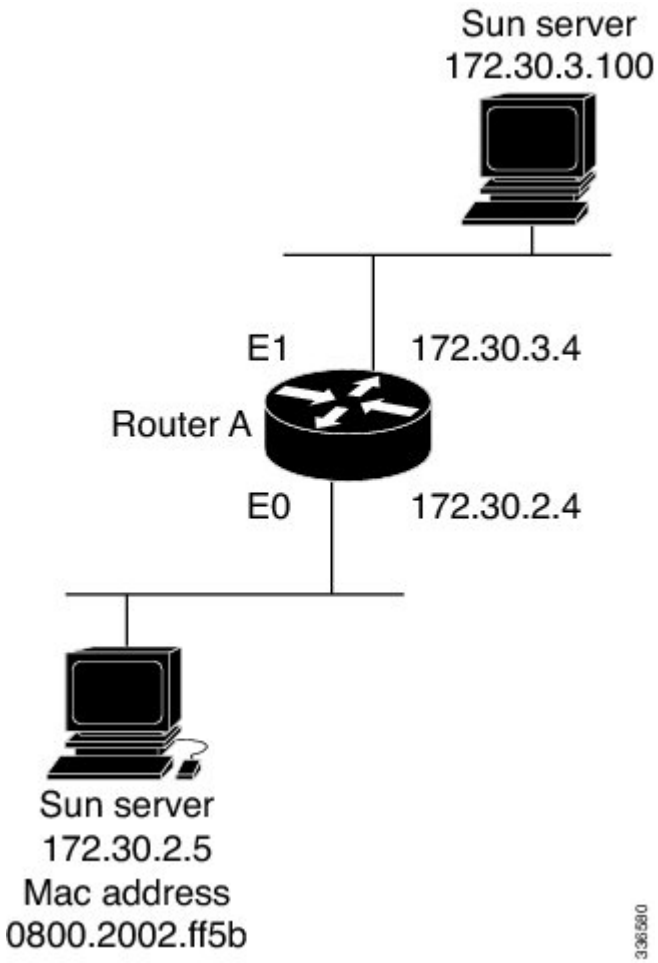
DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type* [*slot*/]*port* **Example:** `Device(config)# interface Gigabitethernet 0/0`	Specifies the interface that you will be configuring the RARP service on and enters interface configuration mode for the specified interface.
Step 4	**ip rarp-server** *ip-address* **Example:** `Device(config-if)# ip rarp-server 172.30.3.100`	Enables the RARP service on the router.

Examples

The figure below illustrates a network configuration in which a router is configured to act as a RARP server for a diskless workstation. In this example, the Sun workstation attempts to resolve its MAC (hardware) address to an IP address by sending a SLARP request, which is forwarded by the router to the Sun server.

Figure 1: Configuring a Router As a RARP Server



Router A has the following configuration:

```
! Allow the router to forward broadcast portmapper requests
ip forward-protocol udp 111
! Provide the router with the IP address of the diskless sun
arp 172.30.2.5 0800.2002.ff5b arpa
interface Gigabitethernet 0/0
! Configure the router to act as a RARP server, using the Sun Server's IP
! address in the RARP response packet.
ip rarp-server 172.30.3.100
! Portmapper broadcasts from this interface are sent to the Sun Server.
ip helper-address 172.30.3.100
```

The Sun client and server's IP addresses must use the same major network number because of a limitation with the current SunOS *rpc.bootparamd* daemon.

In the following example, an access server is configured to act as a RARP server.

```
! Allow the access server to forward broadcast portmapper requests
 ip forward-protocol udp 111
! Provide the access server with the IP address of the diskless sun
 arp 172.30.2.5 0800.2002.ff5b arpa
 interface Gigabitethernet 0/0
! Configure the access server to act as a RARP server, using the Sun Server's
! IP address in the RARP response packet.
 ip rarp-server 172.30.3.100
! Portmapper broadcasts from this interface are sent to the Sun Server.
 ip helper-address 172.30.3.100
```

Configuring System BOOTP Parameters

To configure extended BOOTP parameters for asynchronous interfaces, complete the task in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **async-bootp** *tag* [:*hostname*] *data*
4. **show async bootp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**async-bootp** *tag* [:*hostname*] *data* **Example:** `Device# async-bootp bootfile :172.30.1.1 "pcboot"`	Configures extended BOOTP requests for asynchronous interfaces. **Note** The Boot Protocol (BOOTP) server for asynchronous interfaces supports extended BOOTP requests (defined in RFC 1084). This command is useful in conjunction with using the auxiliary port as an asynchronous interface.
Step 4	**show async bootp** **Example:** `Device# show async bootp`	Displays the extended data that will be sent in BOOTP responses for BOOTP responses.

Examples

For example, if the DNS server address is specified as extended data for BOOTP responses, you will see output similar to the following:

```
Device# show async bootp

The following extended data will be sent in BOOTP responses:
dns-server 172.22.53.210
```

Configuring a Router to Use rsh and rcp

Specifying the Source Interface for Outgoing RCMD Communications

To configure the router so that RCMD connections use the loopback interface as the source address of all packets leaving the router, specify the interface associated with RCMD communications by completing the task in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rcmd source-interface** *interface-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rcmd source-interface** *interface-id* **Example:** `Device(config)# ip rcmd source-interface`	Specifies the interface address that will be used to label all outgoing rsh and rcp traffic.

Disabling DNS Reverse Lookup for rcmd

DNS Reverse Lookup for rcmd is enabled by default. You can disable the DNS check for RCMD (rsh and rcp) access by completing the task in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip rcmd domain-lookup**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no ip rcmd domain-lookup** **Example:** `Device(config)# no ip rcmd domain-lookup`	Disables the Domain Name Service (DNS) reverse lookup function for remote command (rcmp) applications (rsh and rcp).

Configuring the Router to Allow Remote Users to Execute Commands Using rsh

To configure the router to allow remote user to execute commands using rsh, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rcmd remote-host** *local-username* {*ip-address* | *host* } *remote-username* [**enable**[*level*]]
4. **ip rcmd rsh-enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rcmd remote-host** *local-username* {*ip-address* | *host* } *remote-username* [**enable**[*level*]] **Example:** `Device(config)# ip rcmd remote-host Router1 172.16.101.101 netadmin4 enable`	Creates an entry in the local authentication database for each remote user who is allowed to execute rsh commands.
Step 4	**ip rcmd rsh-enable** **Example:** `Device(config)# ip rcmd rsh-enable`	Enables the software to support incoming rsh commands. **Note** To disable the software from supporting incoming rsh commands , use the **noiprcmdrsh-enable** command. **Note** When support of incoming rsh commands is disabled, you can still issue an rsh command to be executed on other routers that support the remote shell protocol and on UNIX hosts on the network.

Examples

The following example shows how to add two entries for remote users to the authentication database, and enable a router to support rsh commands from remote users:

```
ip rcmd remote-host Router1 172.16.101.101 rmtnetad1
ip rcmd remote-host Router1 172.16.101.101 netadmin4 enable
ip rcmd rsh-enable
```

The users, named *rmtnetad1* and *netadmin4* , are both on the remote host at IP address 172.16.101.101. Although both users are on the same remote host, you must include a unique entry for each user. Both users are allowed to connect to the router and remotely execute rsh commands on it after the router is enabled for rsh. The user named *netadmin4* is allowed to execute privileged EXEC mode commands on the router. Both authentication database entries give the router's host name *Router1* as the local username. The last command enables the router for to support rsh commands issued by remote users.

Executing Commands Remotely Using rsh

To execute a command remotely on a network server using rsh, use the following commands in user EXEC mode:

SUMMARY STEPS

1. **enable**
2. **rsh** {*ip-address* | *host* } [/**user***username*] *remote-command*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**rsh** {*ip-address* | *host* } [/**user***username*] *remote-command* **Example:** `Device# rsh mysys.cisco.com /user sharon ls -a`	Executes a command remotely using rsh.

Examples

The following example executes the “ls -a” command in the home directory of the user sharon on mysys.cisco.com using rsh:

```
Router# enable
Router# rsh mysys.cisco.com /user sharon ls -a
.
..
.alias
.cshrc
.emacs
.exrc
.history
.login
.mailrc
.newsrc
.oldnewsrc
.rhosts
.twmrc
.xsession
jazz
Router#
```

Configuring the Router to Accept rcp Requests from Remote Users

To configure the Cisco IOS software to support incoming rcp requests, use the following commands in global configuration mode:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rcmd remote-host** *local-username* {*ip-address* | *host* } *remote-username* [**enable**[*level*]]
4. **ip rcmd rcp-enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rcmd remote-host** *local-username* {*ip-address* | *host* } *remote-username* [**enable**[*level*]] **Example:** `Device(config)# ip rcmd remote-host Router1 172.16.101.101 netadmin3`	Create an entry in the local authentication database for each remote user who is allowed to execute rcp commands. **Note** To disable the software from supporting incoming rcp requests, use the **noiprcmdrcp-enable** command. **Note** When support for incoming rcp requests is disabled, you can still use the rcp commands to copy images from remote servers. The support for incoming rcp requests is distinct from its ability to handle outgoing rcp requests.
Step 4	**ip rcmd rcp-enable** **Example:** `Device(config)# ip rcmd rcp-enable`	Enable the software to support incoming rcp requests.

Examples

The following example shows how to add two entries for remote users to the authentication database and then enable the software to support remote copy requests from remote users. The users, named *netadmin1* on the remote host at IP address 172.16.15.55 and *netadmin3* on the remote host at IP address 172.16.101.101, are

both allowed to connect to the router and remotely execute rcp commands on it after the router is enabled to support rcp. Both authentication database entries give the host name *Router1* as the local username. The last command enables the router to support for rcp requests from remote users.

```
ip rcmd remote-host Router1 172.16.15.55 netadmin1
ip rcmd remote-host Router1 172.16.101.101 netadmin3
ip rcmd rcp-enable
```

Configuring the Remote to Send rcp Requests

To override the default remote username sent on rcp requests, use the following command in global configuration mode:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rcmd remote-username** *username*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rcmd remote-username** *username* **Example:** `Device(config)# ip rcmd remote-username sharon`	Specifies the remote username. **Note** To remove the remote username and return to the default value, use the **noiprcmdremote-username** command.

Configuring a Router to Use FTP Connections

To configure a router to transfer files between systems on the network using the File Transfer Protocol (FTP), complete the tasks in this section to configure the FTP characteristics:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ftp username** *string*
4. **ip ftp password** [*type*] *password*
5. Do one of the following:
   - **ip ftp passive**
   -
   -
   - **no ip ftp passive**
6. **ip ftp source-interface** *interface*

DETAILED STEPS

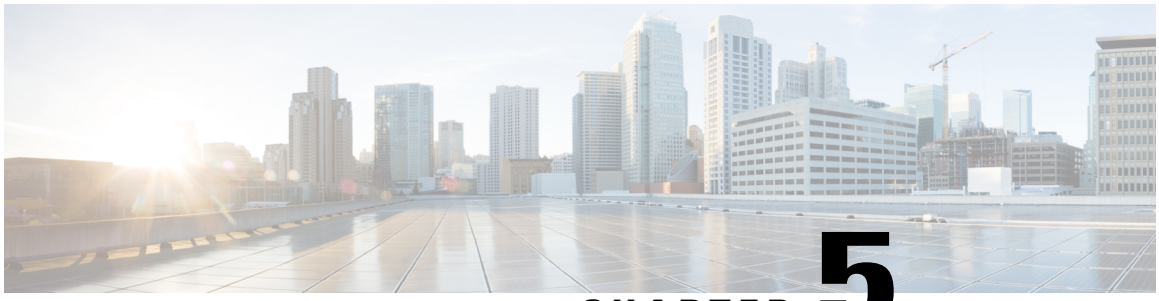
	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip ftp username** *string* **Example:** `Device(config)# ip ftp username zorro`	Specifies the user name to be used for the FTP connection.
Step 4	**ip ftp password** [*type*] *password* **Example:** `Device(config)# ip ftp password sword`	Specifies the password to be used for the FTP connection.
Step 5	Do one of the following: • **ip ftp passive** • • • **no ip ftp passive**	Configures the router to only use passive-mode FTP connections. or Allows all types of FTP connections (default).

	Command or Action	Purpose
	Example: `Device(config)# ip ftp passive`	
Step 6	**ip ftp source-interface** *interface* **Example:** `Device(config)# ip ftp source-interface to1`	Specifies the source IP address for FTP connections.

Examples

The following example demonstrates how to capture a core dump using the Cisco IOS FTP feature. The router accesses a server at IP address 192.168.10.3 with login name zorro and password sword. The default passive-mode FTP is used, and the server is accessed using Token Ring interface to1 on the router where the core dump will occur:

```
ip ftp username zorro
ip ftp password sword
ip ftp passive
ip ftp source-interface to1
! The following command allows the core-dump code to use FTP rather than TFTP or RCP
exception protocol ftp
! The following command identifies the FTP server
! 192.168.10.3 crashes
exception dump 192.168.10.3
```



CHAPTER 5

Transferring Files Using HTTP or HTTPS

Cisco IOS Release 12.4 provides the ability to transfer files between your Cisco IOS software-based device and a remote HTTP server using the HTTP or HTTP Secure (HTTPS) protocol. HTTP and HTTPS can now be specified as the targets and source locations in Cisco IOS command-line interface (CLI) commands that use file system prefixes such as the **copy** command.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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.

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.

Prerequisites for Transferring Files Using HTTP or HTTPs

To copy files to or from a remote HTTP server, your system must support the HTTP client feature, which is integrated in most Cisco IOS software images. The HTTP client is enabled by default. To determine if the HTTP client is supported on your system, issue the **show ip http client all**command. If you are able to execute the command, the HTTP client is supported.

Commands exist for the optional configuration of the embedded HTTP client and for the HTTPS client, but the default configuration is sufficient for using the File Transfer Using HTTP or HTTPS feature. For information on configuring optional HTTP or HTTPS client characteristics, see the “Related Documents” section.

Restrictions for Transferring Files Using HTTP or HTTPs

Existing limitations to the **copy** command, such as no network-to-network copies, are in effect for the File Transfer Using HTTP or HTTPS feature.

Note The **copy** command in Cisco IOS Release 12.4T does not work in conjunction with older versions of the Apache server software. The Apache server software must be upgraded to version 2.0.49 or later in order to use the copy command.

Information About File Transfers Using HTTP or HTTPs

To transfer files using HTTP or HTTPS, you should understand the following concept:

The File Transfer Using HTTP or HTTPs feature provides the capability to copy files, such as Cisco IOS image files, core files, configuration files, log files, scripts, and so on, to and from a remote server and your local routing device using the Cisco IOS **copy** command and command-line interface. The HTTP copy operation works in the same way as copying from other remote file systems, such as FTP or TFTP.

The HTTP copy operation can use the embedded HTTPS client for HTTP Secure transfers, providing secure and authenticated file transfers within the context of a public key infrastructure (PKI).

How to Transfer Files Using HTTP or HTTPs

This section contains the following procedures:

Note To use the File Transfer Using HTTP feature, you may need to specify a username and password for the HTTP connections for those servers that require a username and password to connect. Commands are also available to specify custom connection characteristics, although default settings can be used. The feature also offers commands to monitor and maintain connections and files.

Configuring HTTP Connection Characteristics for File Transfers

Default values are provided for HTTP File transfers. The following task is used to customize the connection characteristics for your network to specify a username and password, connection preferences, a remote proxy server, and the source interface to be used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http client connection** {**forceclose** | **idletimeout***seconds* | **timeout***seconds*}
4. **ip http client username** *username*
5. **ip http client password** *password*
6. **ip http client proxy-server** {*proxy-name* | *ip-address*} [**proxy-port***port-number*]
7. **ip http client source-interface** *interface-id*
8. **do copy running-config startup-config**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip http client connection** {**forceclose** | **idletimeout***seconds* | **timeout***seconds*} **Example:** `Router(config)# ip http client connection timeout 15`	Configures characteristics for HTTP client connections to a remote HTTP server for all file transfers: • **forceclose** --Disables the default persistent connection. • **idle timeout** *seconds* --Sets the period of time allowed for an idle connection, in a range from 1 to 60 seconds. Default timeout is 30 seconds. • **timeout** *seconds* --Sets the maximum time the HTTP client waits for a connection, in a range from 1 to 60 seconds. Default is 10 seconds.
Step 4	**ip http client username** *username* **Example:** `Router(config)# ip http client username user1`	Specifies the username to be used for HTTP client connections that require user authentication. **Note** You can also specify the username on the CLI when you issue the **copy** command, in which case the username entered overrides the username entered with this command. See the "Downloading a File from a Remote Server Using HTTP or HTTPs: Example" section for an example.

	Command or Action	Purpose
Step 5	**ip http client password** *password* **Example:** `Router(config)# ip http client password letmein`	Specifies the password to be used for HTTP client connections that require user authentication. **Note** You can also specify the password on the CLI when you issue the **copy** command, in which case the password entered overrides the password entered with this command. See the "Downloading a File from a Remote Server Using HTTP or HTTPs: Example" section for an example.
Step 6	**ip http client proxy-server** {*proxy-name* | *ip-address*} [**proxy-port***port-number*] **Example:** `Router(config)# ip http client proxy-server edge2 proxy-port 29`	Configures the HTTP client to connect to a remote proxy server for HTTP file system client connections. • The optional**proxy-port***port-number* keyword and argument specify the proxy port number on the remote proxy server.
Step 7	**ip http client source-interface** *interface-id* **Example:** `Router(config)# ip http client source-interface Ethernet 0/1`	Specifies the interface for the source address in all HTTP client connections.
Step 8	**do copy running-config startup-config** **Example:** `Router(config)# do copy running-config startup-config`	(Optional) Saves the running configuration as the startup configuration file. • The **do** command allows you to execute privileged EXEC mode commands from global configuration mode.
Step 9	**end** **Example:** `Router(config)# end` **Example:** `Router#`	Ends your configuration session and returns the CLI to user EXEC mode.

Downloading a File from a Remote Server Using HTTP or HTTPs

Perform this task to download a file from a remote HTTP server using HTTP or HTTPs. The **copy** command helps you to copy any file from a source to a destination.

SUMMARY STEPS

1. **enable**
2. Do one of the following:
   - **copy** [**/erase**] [**/noverify**] **http://***remote-source-urllocal-destination-url*
   - **copy https://** *remote-source-url local-destination-url*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable` **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	Do one of the following: • **copy** [**/erase**] [**/noverify**] **http://***remote-source-urllocal-destination-url* • **copy https://** *remote-source-url local-destination-url* **Example:** `Router# copy http://user1:mypassword@209.165.202.129:8080/image_files/c7200-i-mx flash:c7200-i-mx` **Example:** `Router# copy` **Example:** `copy https://user1:mypassword@209.165.202.129:8080/image_files/c7200-i-mx flash:c7200-i-mx`	Copies a file from a remote web server to a local file system using HTTP or HTTPS. • **/erase** --Erases the local destination file system before copying. This option is provided on Class B file system platforms with limited memory to allow an easy way to clear local flash memory space. • **/noverify** --If the file being copied is an image file, this keyword disables the automatic image verification that occurs after an image is copied. • The *remote-source-url* argument is the location URL (or alias) from which to get the file to be copied, in standard Cisco IOS file system HTTP syntax as follows: **http://** [[*username*:*password*]**@**] {*hostname* | *host-ip*}[*/filepath*]*/filename*

	Command or Action	Purpose
		Note The optional *username* and *password* arguments can be used to log in to an HTTP server that requires user authentication, in place of configuring the **iphttpclientusername** and **iphttpclientpassword** global configuration commands to specify these authentication strings. • The *local-destination-url* is the location URL (or alias) to put the copied file, in standard Cisco IOS file system syntax as follows: *filesystem* **:** [*/filepath*][*/filename*] **Note** For more information on URL syntax when you use the **copy** command, see the "Additional References" section.

Troubleshooting Tips

If file transfers from a remote web server fail, verify the following:

- Your router has an active connection to the Internet.
- The correct path and filename have been specified.
- The remote server requires a username and password.
- The remote server has a nonstandard communications port configured. (The default port for HTTP is 80; the default port for HTTPS is 443.)

The CLI returns error messages to help you determine the cause of a failed copy request. Additional information on the copy process can be displayed with the **debugiphttpclientall** command.

Uploading a File to a Remote Server Using HTTP or HTTPS

Perform this task to upload a file to a remote HTTP server using HTTP or HTTPS.

SUMMARY STEPS

1. **enable**
2. Do one of the following:
   - **copy** [**/erase**] [**/noverify**] *local-source-url***http://***remote-destination-url*
   - **copy** *local-source-url* **https://** *remote-destination-url*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	Do one of the following: • **copy** [**/erase**] [**/noverify**] *local-source-url***http://***remote-destination-url* • **copy** *local-source-url* **https://** *remote-destination-url* **Example:** `Router# http://user1:mypassword@209.165.202.129:8080/image_files/c7200-i-mx_backup` **Example:** `Router# copy flash:c7200-i-mx http://user1:mypassword@209.165.202.129:8080/image_files/c7200-i-mx_backup` **Example:**	Copies a file from a local file system to a remote web server using HTTP or HTTPS. • **/erase** --Erases the local destination file system before copying. This option is provided on Class B file system platforms with limited memory to allow an easy way to clear local flash memory space. • **/noverify** --If the file being copied is an image file, this keyword disables the automatic image verification that occurs after an image is copied. • The *local-source-url* argument is the location URL (or alias) from which to get the file to be copied, in standard Cisco IOS file system syntax as follows: **http://** [[*username*:*password*]**@**] {*hostname* | *host-ip*}[*/filepath*]*/filename* **Note** The optional *username* and *password* arguments can be used to log in to an HTTP server that requires user authentication, in place of configuring the **iphttpclientusername** and **iphttpclientpassword** global configuration commands to specify these authentication strings. • The *remote-destination-url* is the URL (or alias) to put the copied file, in standard Cisco IOS file system syntax, as follows: *filesystem* **:** [*/filepath*][*/filename*] **Note** For more information on URL syntax when you use the **copy** command, see the "Additional References" section.

Troubleshooting Tips

If file transfers from a remote web server fail, verify the following:

- Your router has an active connection to the Internet.
- The correct path and filename have been specified.
- The remote server requires a username and password.
- The remote server has a nonstandard communications port configured. (The default port for HTTP is 80; the default port for HTTPS is 443.)

The CLI returns error messages to help you determine the cause of a failed copy request. Additional information on the copy process can be displayed with the **debugiphttpclientall** command.

Maintaining and Monitoring File Transfers Using HTTP

Perform this task to maintain and monitor HTTP connections. Steps 2 through 4 can be performed in any order.

SUMMARY STEPS

1. **enable**
2. **show ip http client connection**
3. **show ip http client history**
4. **show ip http client session-module**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip http client connection** **Example:** `Router# show ip http client connection`	Displays details about active HTTP client connections.
Step 3	**show ip http client history** **Example:** `Router# show ip http client history`	Displays the last 20 URLs accessed by the HTTP client.

	Command or Action	Purpose
Step 4	**show ip http client session-module** **Example:** `Router# show ip http client session-module`	Displays details about about sessions (applications) that have registered with the HTTP client.

Configuration Examples for the File Transfer Using HTTP or HTTPs

Configuring HTTP Connection Characteristics for File Transfers Example

The following example shows how to configure the HTTP password and username for connection to a remote server that authenticates all users. The example also shows how to configure the connection for a 20-second idle connection period. The maximum time the HTTP client waits for a connection remains at the default 10 seconds.

```
Router(config)# ip http client connection idle timeout 20
Router(config)# ip http client password Secret
Router(config)# ip http client username User1
Router(config)# do show running-config | include ip http client
```

Downloading a File from a Remote Server Using HTTP or HTTPs Example

The following example shows how to configure the file c7200-i-mx is copied from a remote server to flash memory using HTTP. This example also shows how to enter a username and password from the command line for an HTTP server that authenticates users.

```
Router# copy http://user1:mypassword@209.165.202.129:8080/image_files/c7200-i-mx
flash:c7200-i-mx
```

Uploading a File from Flash to the Remote HTTP Server Example

The following example shows how to copy a file from flash memory to the remote HTTP server. The example shows the prompts and displays that can be expected from transferring a file using the**copy** privileged EXEC command.

```
Router# copy flash:c7200-js-mz.ELL2 http://172.19.209.190/user1/c7200-js-mz.ELL2
Address or name of remote host [172.19.209.190]?
Destination filename [user1/c7200-js-mz.ELL2]?
Storing http://172.19.209.190/user1/c7200-js-mz.ELL2 !!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
17571956 bytes copied in 57.144 secs (307503 bytes/sec)
```

Downloading a File from the Remote HTTP Server to Flash Memory Example

The following example shows how to copy a file from the remote HTTP server to the flash memory. The example shows the prompts and displays that can be expected from transferring a file using the **copy** privileged EXEC command.

```
Router# copy http://172.19.209.190/user1/c7200-i-mz.test  flash:c7200-i-mz.test
Destination filename [c7200-i-mz.test]?
Loading http://172.19.209.190/user1/c7200-i-mz.test
.
.
.
11272788 bytes copied in 527.104 secs (21386 bytes/sec)
```

Uploading a File to a Remote Server Using HTTP or HTTPs

The following example shows how to copy a file to the remote server using HTTP or HTTPs.

```
router#copy flash
: http:
Source filename []? running-config
Address or name of remote host []? 10.1.102.1 Destination filename [pilot-confg]?file1 ...
```

Additional References

The following sections provide information related to transferring files using HTTP or HTTPS.

Related Documents

Related Topic	Document Title
Secure HTTP communications	*HTTPS --HTTP Server and Client with SSL 3.0*
Cisco IOS embedded web server	*HTTP 1.1 Web Server and Client*
Cisco IOS embedded web client	*HTTP 1.1 Client*
Network Management Commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Network Management Command Reference*
Configuration Fundamentals Commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Configuration Fundamentals Command Reference*

Standards

Standards	Title
No new of modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
None	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

RFCs

RFCs	Title
RFC 2616	*Hypertext Transfer Protocol -- HTTP/1.1*, R. Fielding, et al.
RFC 2617	*HTTP Authentication: Basic and Digest Access Authentication*, J. Franks, et al.

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Transferring Files Using HTTP or HTTPS

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 . An account on Cisco.com is not required.

Table 7: Feature Information for Transferring Files Using HTTP or HTTPS

Feature Name	Releases	Feature Information
File Download Using HTTP	12.3(2)T	The File Download Using HTTP feature allows you to copy files from an HTTP server to a Cisco IOS software-based platform.
File Upload Using HTTP	12.3(7)T	
File Transfer Using HTTP	12.3(7)T	The File Transfer Using HTTP feature provides the capability to copy files, such as Cisco IOS image files, core files, configuration files, log files, and scripts to and from a remote server and your local routing device using the Cisco IOS **copy** command and command-line interface. The HTTP copy operation works in the same way as copying from other remote file systems, such as FTP or TFTP. This feature provides support for copying files from a Cisco IOS software-based platform to an HTTP server, using either HTTP or HTTPs.