Maintaining Router Memory

This chapter describes how to maintain and use the different types of memory on your router.

To benefit most from the instructions and organization of this chapter, your router must contain a minimal configuration that allows you to interact with the system software. You can create a basic configuration file using the **setup** command facility. See the user guide for your hardware platform for more information on using **setup** at first-time startup. See the "Using Setup for Configuration Changes" chapter in this publication for information on using **setup** after first-time startup.

For a complete description of the memory commands mentioned in this chapter, refer to the "Router Memory Commands" chapter in the *Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online,

Note One or more of the commands that previously appeared in this chapter have been replaced by new commands. Table 9 maps the old commands to their replacements. The old commands continue to perform their normal functions in the current release, but support for these commands will cease in a future release.

Table 9 Mapping Old Commands to New Commands

Old Command	New Command
copy erase flash	**erase flash**
copy verify or **copy verify flash**	**verify flash** (on all systems except Cisco 1600 series, Cisco 3600 series, Cisco 7000 family) **verify** (on Cisco 1600 series, Cisco 3600 series, Cisco 7000 family)
copy verify bootflash	**verify bootflash**

Maintain Router Memory Task List

You can perform the tasks related to Flash memory in the following sections:

- Display Memory Information
- Compare Types of Memory
- Reallocate DRAM Memory (Cisco 3600 series only)

- Partition Flash Memory
- Use Flash Load Helper to Upgrade Software on Run-from-Flash Systems
- Use Flash Upgrade Features (Cisco 3000 and Cisco 4000 series)
- Format Flash Memory (Cisco 7000 family)
- Device Management
- Erase Boot Flash Memory on a Cisco 4500
- Delete, Erase, and Recover Files on a Device

Format Flash Memory is a required first task if you are using a new PCMCIA Flash memory card on the Cisco 7000 family.

Note These tasks assume you have a minimal configuration that you want to modify.

Display Memory Information

Perform the following tasks in EXEC mode to display information about system memory:

Task	Command
List information about Flash memory, including system image filenames and amounts of memory used and remaining.	**show flash**
List the names of the Flash devices currently supported on the router.	**show flash devices** (Cisco 7000 family only)
List information about Flash memory, including system image filenames, amounts of memory used and remaining, and Flash partitions.	**show** *device***: [all | chips | detailed | err | partition** *number* **| summary**] (Cisco 1600 series and Cisco 3600 series) **show flash** [**all | chips | filesys**] [*device***:**] (Cisco 7000 family only) **show flash** [**all | chips | detailed | err | partition** *number* [**all | chips | detailed | err**] | **summary**] (all other platforms)

Refer to the *Configuration Fundamentals Command Reference* for examples of these commands.

Compare Types of Memory

Your router has many different locations where it can store images, configuration files, and microcode. Refer to your hardware documentation for details on the following:

- Which types of memory your router contains
- Where files can be located
- Where images and boot images are located by default

DRAM

Dynamic random-access memory contains two types of memory:

- Primary, main, or processor memory, which is reserved for the CPU to execute Cisco IOS software and to hold the running configuration and routing tables.
- Shared, packet, or I/O memory, which buffers data transmitted or received by the router's network interfaces.

On the Cisco 3600 series routers, you can use the **memory-size iomem** command to configure the proportion of DRAM devoted to main memory and to shared memory.

EPROM

Erasable Programmable Read Only Memory (EPROM). This memory is often referred to simply as ROM. It sometimes contains the following:

- ROM Monitor, which provides a user interface when the router cannot find a valid image.
- The boot loader/helper software (also called the boot image), which helps the router boot when it cannot find a valid Cisco IOS image in Flash memory.

NVRAM

Non-volatile Random Access Memory (NVRAM) stores the following information:

- Startup configuration file for every platform except the Cisco 7000 family.
- For the Cisco 7000 family, the location of the startup configuration depends on the CONFIG_FILE Environment Variable.
- The software configuration register, which is used to determine which image to use when booting the router

Flash

Flash memory stores the Cisco IOS software image. On some platforms, it can store configuration files or boot images. This section contains the following sections:

- Types of Flash memory
- Flash Memory Device Naming Conventions

Types of Flash memory

Depending on the hardware platform, Flash memory might be available as EPROMs, single in-line memory modules (SIMMs), or Flash memory cards. Check the appropriate hardware installation and maintenance guide for information about types of Flash memory available on a specific platform.

Depending on the platform, flash memory is available in the following forms:

- Internal Flash memory
  - Internal Flash memory often contains the system image.
  - Some platforms have two or more banks of Flash memory on one single in-line memory modules (SIMM). If the SIMM has two banks, it is sometimes referred to as *dual-bank Flash memory*. The banks can be partitioned into separate logical devices. See the "Partition Flash Memory" section for information about how to partition Flash memory.

- Bootflash
  - Bootflash often contains the boot image.
  - Bootflash sometimes contains the ROM Monitor.
- Flash memory PC cards or PCMCIA cards

  A Flash memory card that is inserted in to a Personal Computer Memory Card International Association (PCMCIA) slot. This card is used to store system images, boot images, and configuration files.

  The following platforms contains PCMCIA slots:
  - The Cisco 1600 series routers include one PCMCIA slot.
  - The Cisco 3600 series routers include two PCMCIA slots.
  - The Cisco 7200 series Network Processing Engine (NPE) contains two PCMCIA slots
  - The Cisco 7000 RSP700 card and the Cisco 7500 series Route Switch Processor (RSP) card contain two PCMCIA slots.

  Because the Cisco 3600 series and Cisco 7000 family can boot images and load configuration files from several locations, these systems use special ROM monitor environment variables to specify the location and filename of images and configuration files that the router is to use for various functions. Refer to the "Set Environment Variables" section for details.

  Some ciscoFlash MIB variables support the Flash file system on the Cisco 7000 family.

  Note that the internal Flash and the Flash memory card cannot be used as a contiguous bank of Flash memory.

Flash Memory Device Naming Conventions

Many commands use the *device***:** argument to specify a Flash memory device. A colon (:) always follows the device type. The available device are as follows:

- For Cisco 1600 series routers, the device can only be the Flash memory card inserted into the PCMCIA slot (**flash:**).
- For Cisco 3600 series routers, the device can be internal Flash memory (**flash:**) or a Flash memory PC card inserted in one of the PCMCIA slots (**slot0:** or **slot1:**).
- For Cisco 7000 family routers, the device can be internal Flash memory (**bootflash:**) or a Flash memory PC card inserted in one of the PCMCIA slots (**slot0:** or **slot1:**).

For the Cisco 1600 series and Cisco 3600 series, some commands also include a partition number or a filename, in the form *device*:[*partition-number*:][*filename*]. If a filename follows the partition number, you must also enter a colon after the partition number.

For example, **flash:** means internal Flash memory; **flash:1** means the first partition in internal Flash memory; and **flash:1:c3620-i-mz.112-5P** means a particular file in the first partition in internal Flash memory.

What You Can Do from Flash Memory

You can perform the following tasks using flash memory:

- Copy a system image from a server to Flash memory using TFTP or rcp.
- Copy a system image from Flash memory to a network server using TFTP or rcp.

- Copy an image from a Flash device to another Flash device (Cisco 3600 series and Cisco 7000 family).
- For the Cisco 4500 series, Cisco 7000 family, copy a boot image to Flash memory using TFTP or rcp.
- For the Cisco 4500 series, Cisco 7000 family, copy the Flash memory boot image to a network server using TFTP or rcp.
- Boot a router from a software image stored Flash memory either automatically or manually.

Note The system image stored in Flash memory can be changed only from privileged EXEC level on the console terminal.

Write Protection

Flash memory provides write protection against accidental erasing or reprogramming.

- Some platforms have a write-protect jumper which can be removed to prevent reprogramming of Flash memory. You must install the jumper when programming is required.
- Some platforms have write protect switched on Flash memory cards that you can use to protect data. You must set the switch to *unprotected* to write data to the Flash memory card.

Refer to your hardware documentation for information on security jumpers and write protect switches.

Run from Flash Systems

Many Cisco routers load the system image from flash storage into RAM in order to run the Cisco IOS. However, some platforms, such as the Cisco 1600 Series and Cisco 2500 Series, execute the Cisco IOS directly in Flash memory. These platforms "run from Flash memory" systems.

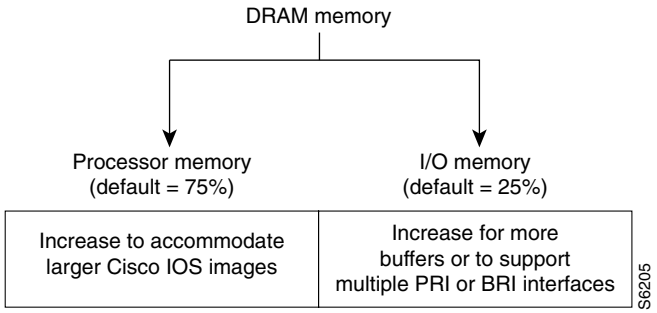
If you want to partition Flash memory, you must use a relocatable image. Relocatable images can be run from any location in Flash and can download images to any location. If you are upgrading from a nonrelocatable image to a relocatable image, you must erase Flash memory during the download so that the image is downloaded as the first file in Flash memory. All images for Run-from-Flash platforms from Cisco IOS Release 11.0 and on are relocatable. See the "Image Naming Conventions" section in the "Loading and Maintaining System Images and Microcode" chapter of the Configuration Fundamentals Configuration Guide to determine if your images are Run-from-Flash images or are relocatable.

Reallocate DRAM Memory (Cisco 3600 series only)

DRAM memory in Cisco 3600 series routers is organized as one contiguous address space divided between processor memory and I/O memory. Depending on the type and number of network interfaces you have configured in the router, you may need to reallocate the DRAM memory partitioned to processor memory and I/O memory.

Cisco manufacturing configures most Cisco 3600 series routers to have 25 percent of the address space allocated to I/O memory and 75 percent allocated to processor memory. But for customer orders that require two or more ISDN PRI interfaces, DRAM memory is configured to provide 40 percent of the address space for I/O memory and 60 percent for processor memory. (See Figure 10.) Cisco Systems performs these DRAM memory adjustments before it ships each router.

Figure 10 Components and Uses of DRAM Memory for Cisco 3600 Series Routers



Note Routers running two or more ISDN PRI interfaces or 12 or more ISDN BRI interfaces require a DRAM memory configuration of 40 percent I/O memory and 60 percent processor memory.

However, there are cases where you may have to manually reallocate the DRAM memory split between processor memory and I/O memory after you have received a router from Cisco Systems.

For example, suppose you receive a Cisco 3640 router with the following running configuration:

- 2 Ethernet and 2 WAN interface card
- 8-port ISDN BRI with an NT1 network module
- IP feature set
- 16 MB of DRAM memory (by default, processor memory = 75%, I/O memory = 25%)
- 4 MB of Flash memory

Later, however, you add a 4-port ISDN BRI network module to the router. You now have 12 ISDN BRI interfaces running on the router. At this point, you must use the **memory-size iomem** command to configure 40 percent of the address space for I/O memory and 60 percent for processor memory.

To view your current mix of processor and I/O memory and reassign memory distribution accordingly, enter the following commands beginning in privileged EXEC mode:

Task		Command
Step 1	View the total amount of memory loaded on the router.	**show version**
Step 2	Determine the amount of free memory.	**show memory**[1]
Step 3	Enter global configuration mode.	**configure terminal**
Step 4	Allocate processor memory and I/O memory.	**memory-size iomem** *I/O-memory-percentage*[2]
Step 5	Exit global configuration mode.	**exit**
Step 6	Save the configuration to NVRAM.	**copy running-config startup-config**
Step 7	Reload the router to run the new image.	**reload**

1. The Free(b) column in the **show memory** command's output shows how much I/O memory is available.
2. The default is 40 percent for I/O memory and 60 percent for processor memory.

Valid I/O memory percentage values are 10, 15, 20, 25, 30, 40 (the default), and 50. A minimum of 4 MB is required for I/O memory. I/O memory size is the specified percentage of total memory size, rounded down to the nearest multiple of 1 MB. The remaining memory is processor memory.

The **memory-size iomem** command does not take effect until you save it to NVRAM using the **copy running-config startup-config** EXEC command and reload the router. However, when you enter the command, the software checks whether the new memory distribution leaves enough processor memory for the currently running Cisco IOS image. If not, the following message appears:

```
Warning: Attempting a memory partition that does not provide enough Processor memory for
the current image.If you write memory now, this version of software may not be able to run.
```

When you enter the **reload** command to run a new image, the software calculates the new processor and I/O memory split. If there is not enough processor memory, it automatically reduces I/O memory to an alternative setting to load the image. If there is still not enough processor memory for the image to run, then you do not have enough DRAM.

Reallocate Processor Memory and I/O Memory Examples

The following example allocates 40 percent of DRAM to I/O memory and the remaining 60 percent to processor memory. The example views the current allocation of memory, changes the allocation, saves the allocation, and reloads the router so the changes can take effect. In the **show memory** command output, the Free(b) column shows how much I/O memory is available

```
Router# show memory
               Head   Total(b)    Used(b)    Free(b)  Lowest(b) Largest(b)
Processor  60913730    3066064     970420    2095644    2090736    2090892
      I/O    C00000    4194304    1382712    2811592    2811592    2805492
--More--

Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# memory-size iomem 40
Router(config)# exit
Router#
Router# copy running-config startup-config
Building configuration...
[OK]

Router# reload

rommon > boot
program load complete, entry point: 0x80008000, size: 0x32ea24
Self decompressing the image :
################################################################################
################################################################################
################################################################## [OK]
```

Partition Flash Memory

On the Cisco 1003, Cisco 1600 series, Cisco 2500 series, Cisco 3000, Cisco 3600 series, Cisco 4000, Cisco 4500, AS5100 series, and the AccessPro card, you can partition banks of Flash memory into separate, logical devices so that the router can hold and maintain two or more different software images. This partitioning allows you to write software into Flash memory while running software in another bank of Flash memory.

Systems that Support Partitioning

To partition Flash memory, you must have at least two banks of Flash memory; a bank is a set of 4 chips. This requirement includes systems that support a single SIMM that has two banks of Flash memory. The minimum partition size is the size of a bank.

CiscoFlash MIB variables support partitioned Flash.

Benefits of Partitioning Flash Memory

Partitioning Flash memory provides the following benefits:

- For any system, partitioning—rather than having one logical Flash memory device—provides a cleaner way of managing different files in Flash memory, especially if the Flash memory size is large.
- For systems that execute code out of Flash memory, partitioning allows you to download a new image into the file system in one Flash memory bank while an image is being executed from the file system in the other bank. The download is simple and causes no network disruption or downtime. After the download is complete, you can switch over to the new image at a convenient time.
- One system can hold two different images, one image acting as a backup for the other. Therefore, if a downloaded image fails to boot for some reason, the earlier running, good image is still available. Each bank is treated as a separate device.

Flash Load Helper versus Dual Flash Bank

Flash load helper is a software option that enables you to upgrade system software on run-from-Flash systems that have a single bank of Flash memory. It is a lower-cost software upgrade solution than dual-bank Flash, which requires two banks of Flash memory on one SIMM. Flash load helper is only available on the Cisco 2500 series and Cisco 3000.

You might use Flash load helper rather than partitioning Flash into two banks for one of the following reasons:

- If you want to download a new file into the same bank from which the current system image is executing.
- If you want to download a file that is larger than the size of a bank, and hence want to switch to a single-bank mode.
- If you have only one single-bank Flash SIMM installed. In this case, Flash load helper is the best option for upgrading your software.

See the "Use Flash Load Helper to Upgrade Software on Run-from-Flash Systems" section for information about using Flash load helper.

Partition Flash Memory

To partition Flash memory, perform the following task in global configuration mode:

Task	Command
Partition Flash memory.	**partition flash** *partitions* [*size1 size2*]
	partition *device***:** [*number-of-partitions*][*partition-size*] (Cisco 1600 series and Cisco 3600 series)

This task will succeed only if the system has at least two banks of Flash and the partitioning does not cause an existing file in Flash memory to be split across the partitions.

For all platform except the Cisco 1600 series and Cisco 3600 series, Flash memory can only be partitioned into two partitions.

For the Cisco 1600 series and Cisco 3600 series, the number of partitions that you can create in a Flash memory device equals the number of banks in the device. Enter the **show** *device***: all** command to view the number of banks on the Flash memory device. The number of partition size entries you set must be equal to the number of specified partitions. For example, the **partition slot0: 2 8 8** command configures two partitions to be 8 MB in size each. The first 8 corresponds to the first partition; the second 8 corresponds to the second partition.

Use Flash Load Helper to Upgrade Software on Run-from-Flash Systems

Flash load helper is a software option that enables you to upgrade system software on run-from-Flash systems that have a single bank of Flash memory. It is a lower-cost software upgrade solution than dual-bank Flash, which requires two banks of Flash memory on one SIMM.

The Flash load helper software upgrade process is simple and does not require additional hardware; however, it does require some brief network downtime. A system image running from Flash can use Flash load helper only if the boot ROMs support Flash load helper. Otherwise, you must perform the Flash upgrade manually. See the "Manually Boot from Flash Memory" section.

Flash load helper is an automated procedure that reloads the ROM-based image, downloads the software to Flash memory, and reboots to the system image in Flash memory. Flash load helper performs checks and validations to maximize the success of a Flash upgrade and minimize the chance of leaving Flash memory either in an erased state or with a file that cannot boot.

In run-from-Flash systems, the software image is stored in and executed from the Flash EPROM rather than from RAM. This method reduces memory cost. A run-from-Flash system requires enough Flash EPROM to hold the image and enough main system RAM to hold the routing tables and data structures. The system does not need the same amount of main system RAM as a run-from-RAM system because the full image does not reside in RAM. Run-from-Flash systems include the Cisco 2500 series and some Cisco 3000 series.

Flash Load Helper Features

Flash load helper includes the following features:

- Confirms access to the specified source file on the specified server before erasing Flash memory and reloading to the ROM image for the actual upgrade.
- Warns you if the image being downloaded is not appropriate for the system.
- Prevents reloads to the ROM image for a Flash upgrade if the system is not set up for automatic booting and the user is not on the console terminal. In the event of a catastrophic failure during the upgrade, Flash load helper can bring up the boot ROM image as a last resort rather than forcing the system to wait at the ROM monitor prompt for input from the console terminal.
- Retries Flash downloads automatically up to six times. The retry sequence is as follows:
  - First try
  - Immediate retry
  - Retry after 30 seconds

  - Reload ROM image and retry
  - Immediate retry
  - Retry after 30 seconds
- Allows you to save any configuration changes made before you exit out of the system image.
- Notifies users logged in to the system of the impending switch to the boot ROM image so that they do not lose their connections unexpectedly.
- Logs console output during the Flash load helper operation into a buffer that is preserved through system reloads. You can retrieve the buffer contents from a running image. The output is useful when console access is unavailable or a failure occurs in the download operation.

Flash load helper can also be used on systems with multiple banks of Flash memory that support Flash memory partitioning. Flash load helper enables you to download a new file into the same partition from which the system is executing an image.

For information about how to partition multiple banks of Flash memory so your system can hold two different images, see the "Partition Flash Memory" section.

Flash Load Helper Configuration Task List

Perform the tasks in the following sections to use and monitor Flash load helper:

- Download a File Using Flash Load Helper
- Monitor Flash Load Helper

Download a File Using Flash Load Helper

To download a new file to Flash memory using Flash load helper, check to make sure that your boot ROMs support Flash load helper and then perform the following task in privileged EXEC mode:

Task	Command
Download a new file to Flash memory.	**copy tftp flash** or **copy mop flash**

The following error message displays if you are in a Telnet session and the system is set for manual booting (the boot bits in the configuration register are zero):

```
ERR: Config register boot bits set for manual booting
```

In case of any catastrophic failure in the Flash memory upgrade, this error message helps to minimize the chance of the system going down to ROM monitor mode and being taken out of the remote Telnet user's control.

The system tries to bring up at least the boot ROM image if it cannot boot an image from Flash memory. Before reinitiating the **copy tftp flash** command, you must set the configuration register boot field to a nonzero value, using the **config-register** global configuration command.

The **copy tftp flash** command initiates a series of prompts to which you must provide responses. The dialog is similar to the following:

```
Router# copy tftp flash

************************** NOTICE ******************************
Flash load helper v1.0
This process will accept the TFTP copy options and then terminate
the current system image to use the ROM based image for the copy.
Router functionality will not be available during that time. If
you are logged in via telnet, this connection will terminate. Users
with console access can see the results of the copy operation.
****************************************************************
```

If terminals other than the one on which this command is being executed are active, the following message appears:

```
There are active users logged into the system.

Proceed? [confirm] y
System flash directory:
File Length  Name/status
1    2251320 abc/igs-kf.914
[2251384 bytes used, 1942920 available, 4194304 total]
```

Enter the IP address or the name of the remote host you are copying from:

```
Address or name of remote host [255.255.255.255]? 172.16.1.111
```

Enter the name of the file you want to copy:

```
Source file name? abc/igs-kf.914
```

Enter the name of the destination file:

```
Destination file name [default = source name]? <Return>
Accessing file 'abc/igs-kf.914' on 172.16.1.111....
Loading from 172.16.13.111:
Erase flash device before writing? [confirm] <Return>
```

If you choose to erase Flash memory, the dialog continues as follows. The **copy tftp flash** operation verifies the request from the running image by trying to copy a single block from the remote TFTP server. Then the Flash load helper is executed, causing the system to reload to the ROM-based system image.

```
Erase flash device before writing? [confirm] y
Flash contains files. Are you sure? [confirm] y
```

If the file does not seem to be a valid image for the system, a warning is displayed and a separate confirmation is sought from you.

```
Copy 'abc/igs-kf.914' from TFTP server
as 'abc/igs-kf.914' into Flash WITH erase? y

%SYS-5-RELOAD: Reload requested
%FLH: rxboot/igs-kf.914r from 172.16.1.111 to flash...
```

If you choose not to erase Flash memory and there is no file duplication, the dialog continues as follows:

```
Erase flash device before writing? [confirm] n
Copy 'abc/igs-kf.914' from TFTP server
as 'abc/igs-kf.914' into Flash WITHOUT erase? y
```

If you choose not to erase Flash memory, and there was file duplication, the dialog continues as follows:

```
Erase flash device before writing? [confirm] n
File 'abc/igs-kf.914' already exists; it will be invalidated!
Invalidate existing copy of 'abc/igs-kf' in flash memory? [confirm] y
Copy 'abc/igs-kf.914' from TFTP server
as 'abc/igs-kf.914' into Flash WITHOUT erase? y
```

If the configuration has been modified but not yet saved, you are prompted to save the configuration:

```
System configuration has been modified. Save? [confirm]
```

If you confirm to save the configuration, you might also receive this message:

```
Warning: Attempting to overwrite an NVRAM configuration previously
written by a different version of the system image. Overwrite the
previous NVRAM configuration? [confirm]
```

Users with open Telnet connections are notified of the system reload, as follows:

```
**System going down for Flash upgrade**
```

If the TFTP process fails, the copy operation is retried up to three times. If the failure happens in the middle of a copy operation so that only part of the file has been written to Flash memory, the retry does not erase Flash memory unless you specified an erase operation. The partly written file is marked as deleted, and a new file is opened with the same name. If Flash memory runs out of free space in this process, the copy operation is terminated.

After Flash load helper finishes copying (whether the copy operation is successful or not), it automatically attempts an automatic or a manual boot, depending on the value of bit zero of the configuration register boot field according to the following:

- If bit zero equals 0, the system attempts a default boot from Flash memory to load up the first bootable file in Flash memory. This default boot is equivalent to a manual **boot flash** command at the ROM monitor prompt.
- If bit zero equals 1, the system attempts to boot based on the boot configuration commands. If no boot configuration commands exist, the system attempts a default boot from Flash memory; that is, it attempts to load the first bootable file in Flash memory.

Monitor Flash Load Helper

To view the system console output generated during the Flash load helper operation, use the image that has been booted up after the Flash memory upgrade. Perform the following task in privileged EXEC mode:

Task	Command
View the console output generated during the Flash load helper operation.	**show flh-log**

If you are a remote Telnet user performing the Flash upgrade without a console connection, this task allows you to retrieve console output when your Telnet connection has terminated due to the switch to the ROM image. The output indicates what happened during the download, and is particularly useful if the download fails.

Use Flash Upgrade Features (Cisco 3000 and Cisco 4000 series)

On the Cisco 3000 series and Cisco 4000 series systems that do not run from Flash memory, the upgrade feature of checks and validations are performed to maximize the success of a Flash upgrade and minimize the chances of leaving Flash memory in either an erased state or with a nonbootable file. The software performs the following checks:

- Confirms that the file will fit into Flash memory (based on the erase option and presence of files in Flash memory). This check is done only for uncompressed system images.
- Attempts to recognize the type of file being downloaded and displays warnings where necessary.

Format Flash Memory (Cisco 7000 family)

On the Cisco 7000 family, you must format a new Flash memory card before using it in a PCMCIA slot. You can also format internal Flash memory (bootflash).

Flash memory cards have sectors that can fail. You can reserve certain Flash memory sectors as "spares" for use when other sectors fail. Use the **format** command to specify between 0 and 16 sectors as spares. If you reserve a small number of spare sectors for emergencies, you do not waste space because you can use most of the Flash memory card. If you specify zero spare sectors and some sectors fail, you must reformat the Flash memory card and thereby erase all existing data.

The format operation requires at least Cisco IOS Release 11.0 system software.

Format Flash Memory Process

Caution The following formatting procedure erases all information in Flash memory. To prevent the loss of important data, proceed carefully.

Use the following procedure to format Flash memory. If you are formatting bootflash, you can skip the first step. If you are formatting a Flash memory card, complete both steps.

Step 1 Insert the new Flash memory card into a PCMCIA slot. Refer to instructions on maintaining the router and replacing PCMCIA cards in your router's hardware documentation for instructions on performing this step.

Step 2 Format Flash memory.

To format Flash memory, complete the following task in EXEC mode:

Task	Command
Format Flash memory.	**format** [**spare** *spare-number*] *device1***:** [[*device2***:**][*monlib-filename*]]

The following example shows the **format** command that formats a Flash memory card inserted in slot 0.

```
Router# format slot0:
Running config file on this device, proceed? [confirm]y
All sectors will be erased, proceed? [confirm]y
Enter volume id (up to 31 characters): <Return>
Formatting sector 1 (erasing)
Format device slot0 completed
```

When the router returns you to the EXEC prompt, the new Flash memory card is successfully formatted and ready for use.

Recovering from Locked Blocks

To recover from locked blocks, reformat the Flash memory card. A locked block of Flash memory occurs when power is lost or a Flash memory card is unplugged during a write or erase operation. When a block of Flash memory is locked, it cannot be written to or erased, and the operation will consistently fail at a particular block location. The only way to recover from locked blocks is by reformatting the Flash memory card with the **format** command.

Caution Formatting a Flash memory card to recover from locked blocks will cause existing data to be lost.

Device Management

If your router has multiple Flash memory devices, you can perform tasks such as changing the default directory, displaying the current device, and listing the files in the device.

Set the System Default Flash Device

You can specify the Flash device that the system uses as the default device. Setting the default Flash device allows you to omit an optional *device***:** argument from related commands. For all EXEC commands that have an optional *device***:** argument, the system uses the device specified by the **cd** command when you omit the optional *device***:** argument. For example, the **dir** command contains an optional *device***:** argument and displays a list of files on a Flash memory device.

To specify a default Flash device, complete the following task from EXEC mode:

Task	Command
Set a default Flash memory device.	**cd** *device***:**

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

```
cd slot0:
```

Display the Current Default Flash Device

To display the current default Flash device specified by the **cd** command, complete the following task from EXEC mode:

Task	Command
Display the current Flash memory device.	**pwd**

The following example shows that the present working device specified by the **cd** command is slot 0:

```
Router> pwd
slot0
Router>
```

The following example uses the **cd** command to change the present working device to bootflash and then uses the **pwd** command to display that present working device:

```
Router> cd bootflash:
Router> pwd
bootflash
Router>
```

Show a List of Files on a Flash Device

You can view a list of the contents of a Flash memory device before manipulating its contents. For example, before copying a new configuration file to a Flash device, you may want to verify that the device 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. You can check the contents a Flash device with the **dir** EXEC command.

To show a list of files on a specified Flash device, complete the following task from EXEC mode:

Task	Command
Display a list of files on a Flash memory device.	**dir** [**/all** | **/deleted**] [**/long**] [*device***:**][*filename*]

The following example instructs the router to list undeleted files for the default device specified by the **cd** command. Notice that the router displays the information in short format because no keywords are used:

```
Router# dir
-#- -length- -----date/time------ name
1   620      May 4 1993 21:38:04 config1
2   620      May 4 1993 21:38:14 config2

7993896 bytes available (1496 bytes used)
```

The following example displays the long version of the same device:

```
Router# dir /long
-#- ED --type-- --crc--- -seek-- nlen -length- -----date/time------ name
1   ... 1        37CEC52E 202EC   7    620      May 4  1993 21:38:04 config1
2   ... 1        37CEC52E 205D8   7    620      May 4  1993 21:38:14 config2

7993896 bytes available (1496 bytes used)
```

Erase Boot Flash Memory on a Cisco 4500

To erase the contents of boot Flash memory, perform the following task at the EXEC prompt:

Task	Command
Erase boot Flash memory.	**erase bootflash**

Delete, Erase, and Recover Files on a Device

Depending on your particular platform, you can erase files, delete files, and recover files in Flash memory.

Erase Files on a Flash Device

When you no longer need a file on a Flash memory device, you can erase it. Erased files cannot be recovered.

To erase a file on a specified Flash device, complete the following task from EXEC mode:

Task	Command
Erase a file in Flash memory.	**erase** [*device***:**]*filename*

Erase a File Example

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... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
```

Delete Files on a Flash 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 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 device, complete the following task from EXEC mode:

Task	Command
Delete a file from a Flash memory device.	**delete** [*device***:**]*filename*

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

If you attempt to delete the configuration 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.

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

```
delete slot0:myconfig
```

Recover Deleted Files on a Flash 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 following task from EXEC mode:

Task	Command
Undelete a deleted file on a Flash memory device.	**undelete** *index* [*device***:**]

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 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, you 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 has not been permanently erased via the **squeeze** command. You can delete and undelete a file up to 15 times.

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

```
undelete slot0: 1
```

Permanently Delete Files on a Flash 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 **show flash** command.

To permanently delete files on a Flash memory device, complete the following task from privileged EXEC mode:

Task	Command
Permanently delete all deleted file on a Flash memory card.	**squeeze** *device***:**

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.