CHAPTER 3

Loading System Images, Microcode Images, and Configuration Files

This chapter describes how to load system images, microcode images, and configuration files. The system images contain the system software, and the configuration files contain commands entered to customize the function of the router. Microcode images contain microcode to be downloaded to various hardware devices. The instructions in this chapter describe how to copy system images from routers to network servers (and vice versa), display and compare different configuration files, and list the system software version running on the router.

This chapter also describes the AutoInstall procedure, which you can use to automatically configure and enable a new router upon startup.

For a complete description of the commands mentioned in this chapter, refer to the “System Image, Microcode Image, and Configuration File Load Commands” chapter in the *Router Products Command Reference* publication.

Note You also can use the **setup** command and its interactive prompts to create a basic configuration file. See the *Router Products Getting Started Guide* for more information.

Cisco’s Implementation of Environment Variables

With Cisco IOS Release 10.3(3), you can use Flash memory cards in the Personal Computer Memory Card International Association (PCMCIA) Flash memory card slots on your Cisco 7500 series. The Cisco 7500 series Route Switch Processor (RSP) card contains two PCMCIA slots.

These Flash memory cards can store executable images and configuration files. The router can now boot images and load configuration files from Flash memory cards as well as from internal flash (or “bootflash” on the Cisco 7500 series), NVRAM, and the network.

Because the Cisco 7500 series can boot images and load configuration files from several locations, it uses 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. These special environment variables are as follows:

- BOOT
- BOOTLDR
- CONFIG_FILE

BOOT Environment Variable

The BOOT environment variable specifies a list of bootable images on various devices. For the Cisco 7500 series, valid devices are internal flash (**bootflash:**), the first PCMCIA slot (**slot0:**), the second PCMCIA slot (**slot1:**), and **tftp**. Once you save the BOOT environment variable to your startup configuration, the router checks the variable upon startup to determine the device and filename of the image to boot.

The router tries to boot the first image in the BOOT environment variable list. If the router is unsuccessful at booting that image, it tries to boot the next image specified in the list. The router tries each image in the list until it successfully boots. If the router cannot boot any image in the BOOT environment variable list, then the router attempts to boot the rxboot image of the Cisco 7500 series.

If an entry in the BOOT environment variable list does not specify a device, the router assumes the device is **tftp**. If an entry in the BOOT environment variable list specifies an invalid device, the router skips that entry.

BOOTLDR Environment Variable

The BOOTLDR environment specifies the flash device and filename containing the rxboot image that the ROM monitor uses. For the Cisco 7500 series, valid devices are **bootflash:**, **slot0:**, and **slot1:**.

This environment variable allows you to have several rxboot images. Moreover, you can instruct the ROM monitor to use a specific rxboot image without having to switch out ROMs. Once you save the BOOTLDR environment variable to your startup configuration, the router checks the variable upon startup to determine which rxboot image to use.

CONFIG_FILE Environment Variable

The CONFIG_FILE environment variable specifies the device and filename of the configuration file to use for initialization (startup). For the Cisco 7500 series, valid devices are **bootflash:**, **nvram:**, **slot0:**, and **slot1:**. Once you save the CONFIG_FILE environment variable to your startup configuration, the router checks the variable upon startup to determine the location and filename of the configuration file to use for initialization.

The router uses the NVRAM configuration during initialization when the CONFIG_FILE environment variable does not exist or when it is null (such as at first-time startup). If the router detects a problem with NVRAM or the configuration it contains, the router enters **setup** mode. Refer to the *Router Products Getting Started Guide* for more information on the **setup** command facility.

Controlling Environment Variables

Although the ROM monitor controls environment variables, you can create, modify, or view them with certain system image commands. To create or modify the BOOT, BOOTLDR, and CONFIG_FILE environment variables, use the **boot system**, **boot bootldr**, and **boot config** system image commands, respectively.

Note When you use these three global configuration commands, you affect only the running configuration. You must save the environment variable settings to your startup configuration to place the information under ROM monitor control and for the environment variables to function as expected. Use the **copy running-config startup-config** or **write memory** command to save the environment variables from your running configuration to your startup configuration.

You can view the contents of the BOOT, BOOTLDR, and the CONFIG_FILE environment variables by issuing the **show boot** command. This command displays the settings for these variables as they exist in the startup configuration as well as in the running configuration if a running configuration setting differs from a startup configuration setting.

Use the **show configuration** command to display the contents of the configuration file pointed to by the CONFIG_FILE environment variable.

For complete information on the commands presented in this section, refer to the *Router Products Command Reference* publication.

Cisco’s rsh and rcp Implementation

One of the first attempts to use the network as a resource in the UNIX community resulted in the design and implementation of the remote shell protocol, which included remote shell (rsh) and remote copy (rcp). Rsh and rcp give users the ability to execute commands remotely and 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 will interoperate with standard implementations of rsh and rcp.

From the router, you can use rsh 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 into the target host.

In other words, you do not need to connect to the system or router 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 routers without connecting to the target router, executing the command, and then disconnecting from the router. This is useful for looking at statistics on many different routers.

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

You can enable rsh support on a Cisco router to allow users on remote systems to execute commands on the router. However, our implementation of rsh does not support an *.rhosts* file. Instead, you 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 in concept and use 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.

The rcp copy 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 only need to have access to a server that supports 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 on the destination file. If the destination file does not exist, rcp creates it for you.

Although our rcp implementation emulates the behavior of the UNIX rcp implementation—copying files among systems on the network—our command syntax differs from the UNIX rcp command syntax. Our rcp support offers a set of copy commands that use rcp as the transport mechanism. These rcp copy commands are similar in style to our TFTP copy commands, but they offer an alternative that provides faster performance and reliable delivery of data. This is 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 on the router to allow users on remote systems to copy files to and from the router.

System Image, Microcode Image, and Configuration File Load Task List

You can perform the tasks in the following sections to load system images, microcode images, and configuration files.

- Use the AutoInstall Procedure
- Format Flash Memory on Cisco 7500 Series
- Enter Configuration Mode
- Modify the Configuration Register Boot Field
- Specify the Startup System Image
- Specify the Startup Configuration File
- Schedule a Reload of the System Image
- Additional Cisco 3000 and Cisco 4000 Copying and Automatic Booting Features
- Change the Buffer Size for Loading Configuration Files
- Compress Configuration Files
- Stop Booting and Manually Load a System Image from ROM Monitor
- Boot Systems That Have Dual-Bank Flash Memory
- Configure a Router as a TFTP Server
- Configure a Router to Support Incoming rcp Requests and rsh Commands
- Configure a Router as a RARP Server
- Configure the Remote Username for rcp Requests
- Specify Asynchronous Interface Extended BOOTP Requests
- Specify MOP Server Boot Requests
- Copy System Images from a Network Server to Flash Memory Using TFTP
- Copy System Images from a Network Server to Flash Memory Using rcp
- Additional Cisco 3000 and Cisco 4000 Flash Upgrade Features
- Copy Bootstrap Images from a Network Server to Flash Memory Using rcp or TFTP
- Use Flash Load Helper to Upgrade Software on Run-from-Flash Systems
- Verify the Image in Flash Memory
- Partition Flash Memory Using Dual Flash Bank
- Copy System Images from Flash Memory to a Network Server Using TFTP
- Copy System Images from Flash Memory to a Network Server Using rcp
- Copy a Configuration File from a Network Server to the Router Using rcp
- Copy a Configuration File from the Router to a Network Server Using TFTP
- Copy a Configuration File from the Router to a Network Server Using rcp
- Display System Image and Configuration Information
- Clear the Configuration Information

- Reexecute the Configuration Commands in Startup Configuration
- Remotely Execute Commands Using rsh
- Use Flash Memory as a TFTP Server
- Manage Flash Files on Cisco 7500 Series
- Load Microcode Images over the Network
- Display Microcode Information

Use the AutoInstall Procedure

This section provides information about AutoInstall, a procedure that allows you to configure a new router automatically and dynamically. The AutoInstall procedure involves connecting a new router to a network on which there is an existing preconfigured router, turning on the new router, and enabling it with a configuration file that is automatically downloaded from a Trivial File Transfer Protocol (TFTP) server.

The following sections provide the requirements for AutoInstall and an overview of how the procedure works. To start the procedure, go to “Perform the AutoInstall Procedure” later in this section.

AutoInstall Requirements

For the AutoInstall procedure to work, your system must meet the following requirements:

- Both routers must be physically attached to the network using one or more of the following interface types: Ethernet, Token Ring, Fiber Distributed Data Interface (FDDI), serial with High-Level Data Link Control (HDLC) encapsulation, or serial with Frame Relay encapsulation. HDLC is the default serial encapsulation. Frame Relay will be automatically configured by the AutoInstall process. If the AutoInstall process fails over HDLC, the router will automatically configure Frame Relay encapsulation.
- The existing preconfigured router must be running Software Release 9.1 or later. For AutoInstall over Frame Relay, routers on both sides must be running Cisco Internetwork Operating System (IOS) Release 10.3 or later.
- The new router must be running Software Release 9.1 or later. For AutoInstall over Frame Relay, the new router must be running Cisco IOS Release 10.3 or later.

Note Only Token Ring interfaces that set ring speed with physical jumpers will support AutoInstall. AutoInstall does not work with Token Ring interfaces for which the ring speed must be set using software configuration commands. If the ring speed is not set, the interface is set to shutdown mode.

- You must complete procedures 1 and either 2 or 3:
    - Procedure 1: A configuration file for the new router must reside on a TFTP server. This file can contain the new router’s full configuration or the minimum needed for the administrator to Telnet into the new router for configuration.

  - Procedure 2: A file named network-confg also must reside on the server. The file must have an Internet Protocol (IP) host name entry for the new router. The server must be reachable from the existing router.
  - Procedure 3: An IP address-to-host name mapping for the new router must be added to a Domain Name System (DNS) database file.
- If the existing router is to help automatically install the new router via a HDLC-encapsulated serial interface using Serial Line Address Resolution Protocol (SLARP), that interface must be configured with an IP address whose host portion has the value 1 or 2. (AutoInstall over Frame Relay does not have this address constraint.) Subnet masks of any size are supported.
- If the existing router is to help automatically install the new router using Frame Relay encapsulated serial interface, that interface must be configured with the following:
  - An IP helper address pointing to the TFTP server. In the following example, 171.69.2.75 is the address of the TFTP server:

    ```
    ip helper 171.69.2.75
    ```
  - A Frame Relay map pointing back to the new router. In the following example, 172.21.177.100 is the IP address of *newrouter*’s serial interface. The PVC identifier is 100:

    ```
    frame-relay map ip 172.21.177.100 100 dlci
    ```
- If the existing router is to help automatically install the new router via an Ethernet, Token Ring, or FDDI interface using BOOTP or Reverse Address Resolution Protocol (RARP), a BOOTP or RARP server also must be set up to map the new router’s Media Access Control (MAC) address to its IP address.
- IP helper addresses might need to be configured to forward the TFTP and DNS broadcast requests from the new router to the host that is providing those services.

Using a DOS-based TFTP Server

AutoInstall over Frame Relay and over other WAN encapsulations support downloading configuration files from UNIX-based and DOS-based TFTP servers. Other booting mechanisms such as RARP and SLARP also support UNIX-based and DOS-based TFTP servers.

The DOS format of the UNIX network-confg file that must reside on the server must be eight characters or less, with a three-letter extension. Therefore, when an attempt to load network-confg fails, AutoInstall automatically attempts to download cisconet.cfg from the TFTP server.

If cisconet.cfg exists and a download succeeds, then the server is assumed to be a DOS machine. The AutoInstall program will then attempt to resolve the host name for the router through host commands in cisconet.cfg.

If cisconet.cfg does not exist or cannot be downloaded, or the program is unable to resolve a host name, DNS will attempt to resolve the host name of the router. If it is unable to resolve the host name through DNS, the router will attempt to download ciscortr.cfg. If the host name is longer than eight characters, it will get truncated to eight characters. For example, a router with a host name “australia” will be treated as “australi” and an attempt will be made to download australi.cfg.

The format of cisconet.cfg and ciscortr.cfg are to be the same as those described for network-confg and hostname-confg.

If neither network-confg nor cisconet.cfg exist and DNS is unable to resolve the host name, the program will attempt to load router-confg, and then ciscortr.cfg if router-confg does not exist or cannot be downloaded. The cycle is repeated three times.

How AutoInstall Works

Once the requirements for using AutoInstall are met, the dynamic configuration of the new router occurs in the following order:

1. The new router acquires its IP address. Depending upon the interface connection between the two routers, the new router's IP address is dynamically resolved by either SLARP requests or BOOTP or RARP requests.
2. The new router resolves its name either through network-confg or cisconet.cfg or through DNS.
3. The new router automatically requests and downloads its configuration file from a TFTP server.
4. If a host name is not resolved, the *newrouter* will attempt to load router-confg or ciscortr.cfg.

Acquiring the New Router's IP Address

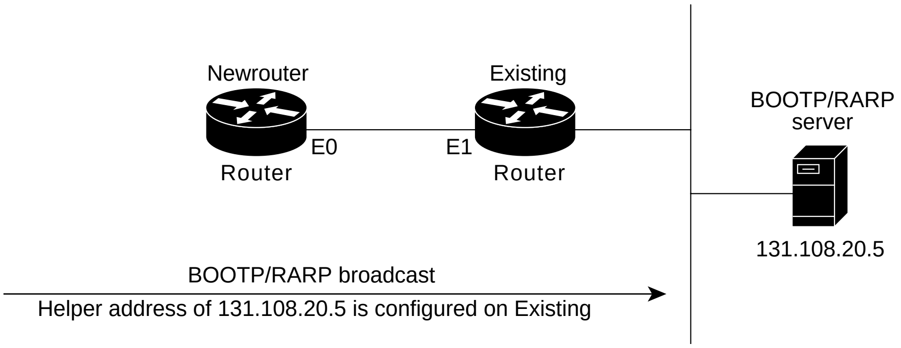
The new router (*newrouter*) resolves its interface's IP addresses by one of the following means:

- If *newrouter* is connected by an HDLC-encapsulated serial line to the existing router (*existing*), *newrouter* sends a SLARP request to *existing.*
- If *newrouter* is connected by an Ethernet, Token Ring, or FDDI interface, it broadcasts BOOTP and RARP requests.
- If *newrouter* is connected by a Frame Relay-encapsulated serial interface, it first attempts the HDLC automatic installation process and then attempts the BOOTP/RARP process over Ethernet, Token Ring or FDDI. If both of these attempts fail, it attempts to automatically install over Frame Relay. In this case, a BOOTP request is sent over the lowest numbered serial or HSSI interface.

The existing router (*existing*) responds in one of the following ways depending upon the request type:

- In response to a SLARP request, *existing* sends a SLARP reply packet to *newrouter.* The reply packet contains the IP address and netmask of *existing*. If the host portion of the IP address in the SLARP response is 1, *newrouter* will configure its interface using the value 2 as the host portion of its IP address and vice versa. (See Figure 3-1.)

Figure 3-1 Using SLARP to Acquire the New Router's IP Address

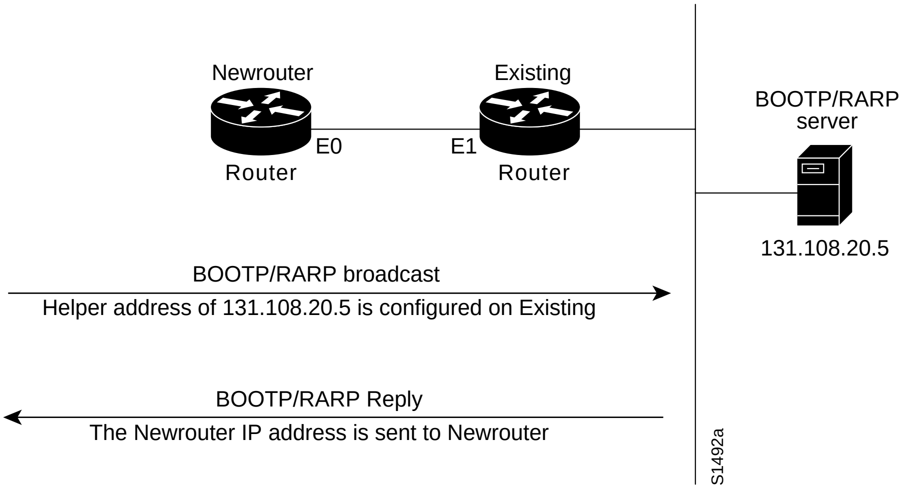


- In response to BOOTP or RARP requests, an IP address is sent from the BOOTP or RARP server to *newrouter.*

  A BOOTP or RARP server must have already been set up to map *newrouter*'s MAC address to its IP address. If the BOOTP server does not reside on the directly attached network segment, routers between *newrouter* and the BOOTP server can be configured using the **ip helper-address** command to allow the request and response to be forwarded between segments, as shown in Figure 3-2.

  AutoInstall over Frame Relay is a special case in which BOOTP is used but the existing router acts as a BOOTP server and responds to the incoming BOOTP request. Only a helper address and a Frame Relay map need to be set up. There is no need for a MAC-to-IP address map on the existing router.

Figure 3-2 Using BOOTP or RARP to Acquire the New Router's IP Address



As of Software Release 9.21, routers can be configured to act as RARP servers.

As soon as one interface resolves its IP address, the router will attempt to resolve its host name. Therefore, only one IP address needs to be set up using SLARP, BOOTP, or RARP.

Resolving the IP Address to the Host Name

The new router resolves its IP address-to-host name mapping by sending a TFTP broadcast requesting the file network-confg, as shown in Figure 3-3.

The network-confg file is a configuration file generally shared by several routers. In this case, it is used to map the IP address of the new router just obtained dynamically to the name of the new router. The file network-confg must reside on a reachable TFTP server and must be globally readable.

The following is an example of a minimal network-confg file that maps the IP address of the new router (131.108.10.2) to the name *newrouter.* The address of the new router was learned via SLARP and is based on *existing*'s IP address of 131.108.10.1.

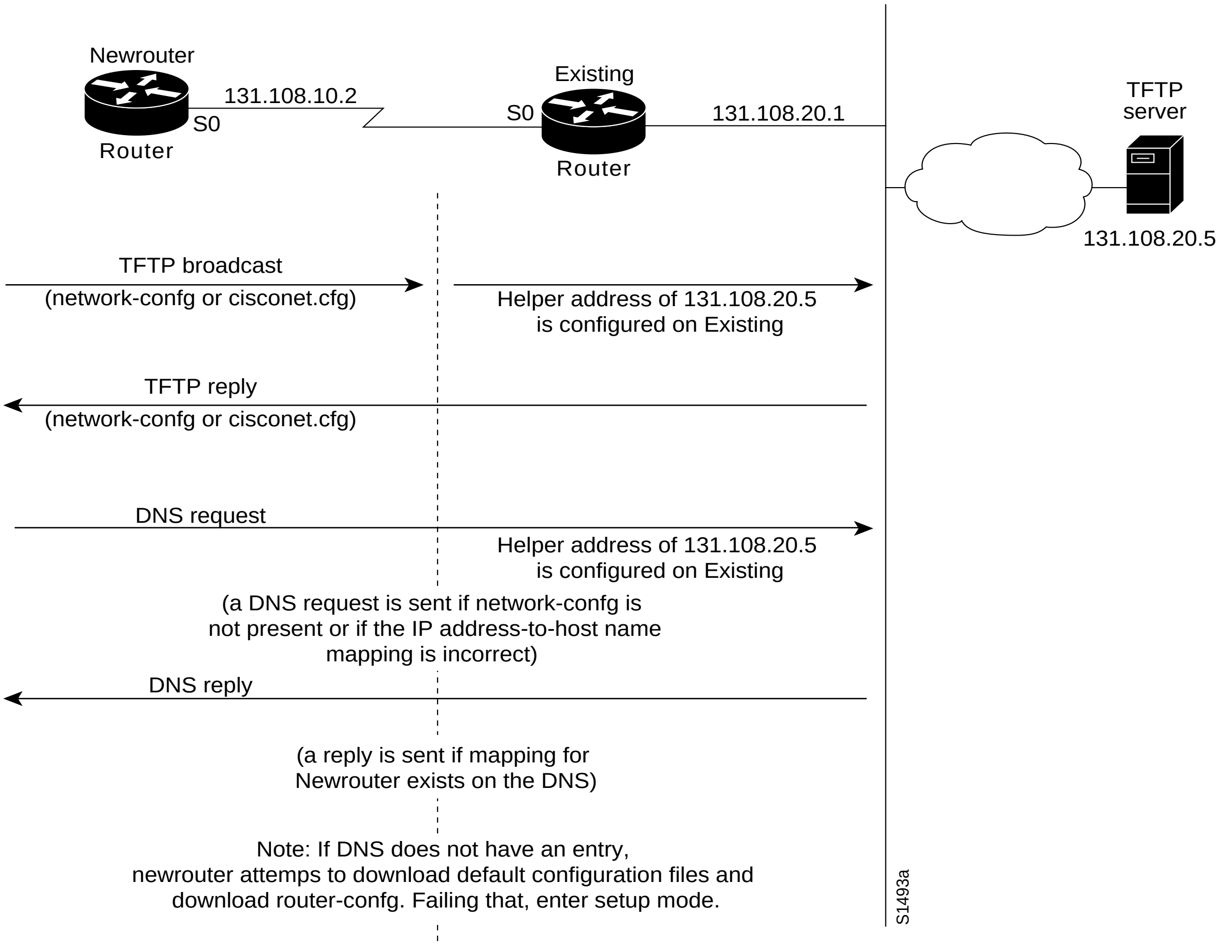
```
ip host newrouter 131.108.10.2
```

If you are not using AutoInstall over Frame Relay, the host portion of the address must be 1 or 2. AutoInstall over Frame Relay does not have this addressing constraint.

If *newrouter* does not receive a network-confg file, or if the IP address-to-host-name mapping does not match the newly acquired IP address, *newrouter* sends a DNS broadcast. If DNS is configured and has an entry that maps *newrouter*'s SLARP, BOOTP, or RARP-acquired IP address to its name, *newrouter* successfully resolves its name.

If DNS does not have an entry that maps the new router's SLARP, BOOTP, or RARP-acquired address to its name, the new router cannot resolve its host name. The new router attempts to download a default configuration file as described in the next section, and failing that, enters **setup** mode (except with AutoInstall over Frame Relay, in which case the router enters user EXEC mode).

Figure 3-3 Dynamically Resolving the New Router's IP Address-to-Host Name Mapping


Newrouter
Router
S0
131.108.10.2
Existing
S0
Router
131.108.20.1
TFTP server
131.108.20.5
TFTP broadcast
(network-confg or cisconet.cfg)
Helper address of 131.108.20.5 is configured on Existing
TFTP reply
(network-confg or cisconet.cfg)
DNS request
Helper address of 131.108.20.5 is configured on Existing
(a DNS request is sent if network-confg is not present or if the IP address-to-host name mapping is incorrect)
DNS reply
(a reply is sent if mapping for Newrouter exists on the DNS)
Note: If DNS does not have an entry, newrouter attemps to download default configuration files and download router-confg. Failing that, enter setup mode.
S1493a

Downloading the New Router's Host Configuration File

After the router successfully resolves its host name, *newrouter* sends a TFTP broadcast requesting the file newrouter-confg. The name newrouter-confg must be in all lowercase, even if the true host name is not. If *newrouter* cannot resolve its host name, it sends a TFTP broadcast requesting the default host configuration file router-confg. The file is downloaded to *newrouter*, where the configuration commands take effect immediately.

When using AutoInstall over Frame Relay, you are put into **setup** mode while the AutoInstall process is running. If the configuration file is successfully installed, the **setup** process is terminated. If you expect the AutoInstall process to be successful, either do *not* respond to the **setup** prompts or respond to the prompts as follows:

```
Would you like to enter the initial configuration dialog? [yes]: no
Would you like to terminate autoinstall? [yes]: no
```

If you do not expect the AutoInstall process to be successful, create a configuration file by responding to the **setup** prompts. The AutoInstall process is terminated transparently.

You will see the following display as the AutoInstall operation is in progress:

```
Please Wait. AutoInstall being attempted!!!!!!!!!!!!!!!!!!!
```

If the host configuration file contains only the minimal information, you must Telnet into *existing*, from there Telnet to *newrouter*, and then run the **setup** command to configure *newrouter*. Refer to the *Router Products Getting Started Guide* for details on the **setup** command.

If the host configuration file is complete, *newrouter* should be fully operational. You can enter the **enable** command (with the system administrator password) at the system prompt on *newrouter*, and then issue the **write memory** command to save the information in the recently obtained configuration file into nonvolatile random-access memory (NVRAM) or the location specified by the CONFIG_FILE environment variable. If a reload occurs, *newrouter* simply loads its configuration file from NVRAM.

If the TFTP request fails, or if *newrouter* still has not obtained the IP addresses of all its interfaces, and those addresses are not contained in the host configuration file, then *newrouter* enters **setup** mode automatically. Setup mode prompts for manual configuration of the router via the console. The new router continues to issue broadcasts to attempt to learn its host name and obtain any unresolved interface addresses. The broadcast frequency will dwindle to every ten minutes after several attempts. Refer to the *Router Products Getting Started Guide* for details on the **setup** command.

Perform the AutoInstall Procedure

To dynamically configure a new router using AutoInstall, complete the following tasks. Steps 1, 2, and 3 are completed by the central administrator. Step 4 is completed by the person at the remote site.

Step 1 Modify the existing router's configuration to support the AutoInstall procedure.

Step 2 Set up the TFTP server to support the AutoInstall procedure.

Step 3 Set up the BOOTP or RARP server if needed. A BOOTP or RARP server is required for AutoInstall using an Ethernet, Token Ring, FDDI, or Frame Relay-encapsulated serial interface. With a Frame Relay-encapsulated serial interface, the existing router acts as the BOOTP server. A BOOTP or RARP server is not required for AutoInstall using an HDLC-encapsulated serial interface.

Step 4 Connect the new router to the network.

Modify the Existing Router's Configuration

You can use any of the following types of interface:

- An HDLC-encapsulated serial line (the default configuration for a serial line)
- An Ethernet, Token Ring, FDDI interface
- A Frame Relay-encapsulated serial line

Use an HDLC-Encapsulated Serial Interface Connection

To set up AutoInstall via a serial line with HDLC encapsulation (the default), you must configure the existing router. Perform the following steps, beginning in global configuration mode:

Task		Command
Step 1	Configure the serial interface that connects to the new router with HDLC encapsulation (the default) and enter interface configuration mode.	**interface serial** *interface-number*[1]
Step 2	Enter an IP address for the interface. The host portion of the address must have a value of 1 or 2. (AutoInstall over Frame Relay does not have this address constraint.)	**ip address** *address mask*[2]
Step 3	Configure a helper address for the serial interface to forward broadcasts associated with the TFTP, BOOTP, and DNS requests.	**ip helper-address** *address*
Step 4	Optionally, configure a DCE clock rate for the serial line, unless an external clock is being used. This step is needed only for DCE appliques.	**clock rate** *bps*
Step 5	Exit configuration mode.	**^Z**
Step 6	Save the configuration changes to your startup configuration. On most platforms, this step saves the configuration to NVRAM. On the Cisco 7500 series, this step saves the configuration to the location specified by the CONFIG_FILE environment variable.	**write memory**

[1] This command and the **clock rate** command are documented in the "Interface Commands" chapter in the *Router Products Command Reference* publication.

[2] This command and the **ip helper-address** command are documented in the "IP Commands" chapter in the *Router Products Command Reference* publication.

In the following example, the existing router's configuration file contains the commands needed to configure the router for AutoInstall on a serial line using HDLC encapsulation:

```
configure terminal
interface serial 0
ip address 131.108.10.1 255.255.255.0
ip helper-address 131.108.20.5
^Z
write memory
```

Use an Ethernet, Token Ring, or FDDI Interface Connection

To set up AutoInstall using an Ethernet, Token Ring, or FDDI interface, you must modify the configuration of the existing router. Perform the following steps, beginning in global configuration mode.

Task		Command
Step 1	Configure a LAN interface and enter interface configuration mode.	**interface** {**ethernet** | **tokenring** | **fddi**} *interface-number* [1]
Step 2	Enter an IP address for the interface.	**ip address** *address mask* [2]
Step 3	Optionally, configure a helper address to forward broadcasts associated with the TFTP, BOOTP, and DNS requests.	**ip helper-address** *address*
Step 4	Exit configuration mode.	**^Z**
Step 5	Save the configuration changes to your startup configuration. On most platforms, this step saves the configuration to NVRAM. On the Cisco 7500 series, this step saves the configuration to the location specified by the CONFIG_FILE environment variable.	**write memory**

[1] This command is documented in the "Interface Commands" chapter in the *Router Products Command Reference* publication.

[2] This command and the **ip helper-address** command are documented in the "IP Commands" chapter in the *Router Products Command Reference* publication.

Typically, the LAN interface and IP address are already configured on the existing router. You might need to configure an IP helper address if the TFTP server is not on the same network as the new router.

In the following example, the existing router's configuration file contains the commands needed to configure the router for AutoInstall on an Ethernet interface:

```
configure terminal
interface Ethernet 0
ip address 131.108.10.1 255.255.255.0
ip helper-address 131.108.20.5
^Z
write memory
```

Use a Frame Relay-Encapsulated Serial Interface Connection

To set up AutoInstall via a serial line with Frame Relay encapsulation, you must configure the existing router. Perform the following tasks, beginning in global configuration mode:

Task		Command
Step 1	Configure the serial interface that connects to the new router and enter interface configuration mode.	**interface serial 0**[1]
Step 2	Configure Frame Relay encapsulation on the interface that connects to the new router.	**encapsulation frame-relay**
Step 3	Create a Frame Relay map pointing back to the new router. or For point-to-point subinterfaces, assign a data link connection identifier (DLCI) to the interface that connects to the new router, and provide the IP address of the serial port on the new router.	**frame-relay map ip** *ip-address dlci*[2] or **frame-relay interface-dlci** *dlci option* [**protocol ip** *ip-address*][2]
Step 4	Enter an IP address for the interface. This step sets the IP address of the existing router.	**ip address** *address mask*[3]

Task		Command
Step 5	Configure a helper address for the TFTP server.	**ip helper-address** *address*
Step 6	Optionally, configure a DCE clock rate for the serial line, unless an external clock is being used. This step is needed only for DCE appliques.	**clock rate** *bps*
Step 7	Exit configuration mode.	**^Z**
Step 8	Save the configuration changes to your startup configuration. On most platforms, this step saves the configuration to NVRAM. On the Cisco 7500 series, this step saves the configuration to the location specified by the CONFIG_FILE environment variable.	**write memory**

1. This command, the **encapsulation** command, and the **clock rate** command are documented in the "Interface Commands" chapter in the *Router Products Command Reference* publication.
2. This command is documented in the "Frame Relay Commands" chapter in the *Router Products Command Reference* publication.
3. This command and the **ip helper-address** command are documented in the "IP Commands" chapter in the *Router Products Command Reference* publication.

You must use a DTE interface on the new router because the network will always provide the clock signal.

In the following example, the existing router's configuration file contains the commands needed to configure the router for Frame Relay AutoInstall on a serial line:

```
configure terminal
interface serial 0
ip address 131.108.20.20 255.255.255.0
encapsulation frame-relay
frame-relay map ip 131.108.10.1 48
ip helper-address 131.108.20.5
^Z
write memory
```

Set Up the TFTP Server

For AutoInstall to work correctly, the new router must be able to resolve its host name and then download a *name*-confg or *name*.cfg file from a TFTP server. The new router can resolve its host name by using a network-confg or cisconet.cfg file downloaded from a TFTP server or by using the DNS.

To set up a TFTP server to support AutoInstall, complete the following tasks. Step 2 includes two ways to resolve the new router's host name. Use the first method if you want to use a network-config file to resolve the new router's host name. Use the second method if you want to use the DNS to resolve the new router's host name.

Task		Command
Step 1	Enable TFTP on a server.	Consult your host vendor's TFTP server documentation and RFCs 906 and 783.
Step 2	If you want to use a network-confg or cisconet.cfg file to resolve the new router's name, create the network-confg or cisconet.cfg file containing an IP address-to-host name mapping for the new router. Enter the **ip host** command into the TFTP config file, not into the router. The IP address must match the IP address that is to be dynamically obtained by the new router.	**ip host** *hostname address*[1]
	or	
	If you want to use the DNS to resolve the new router's name, create an address-to-name mapping entry for the new router in the DNS database. The IP address must match the IP address that is to be dynamically obtained by the new router.	Contact the DNS administrator or refer to RFCs 1101 and 1183.
Step 3	Create the *name*-confg or *name*.cfg file, which should reside in the tftpboot directory on the TFTP server. The *name* part of *name*-confg or *name*.cfg filename must match the host name you assigned for the new router in the previous step. Enter configuration commands for the new router into this file	See the appropriate chapter in this guide for specific commands.

[1] This command is documented in the "IP Commands" chapter in the *Router Products Command Reference* publication.

The *name*-confg or the *name*.cfg file can contain either the new router's full configuration or a minimal configuration.

The minimal configuration file consists of a virtual terminal password and an enable password. It allows an administrator to Telnet into the new router to configure it. If you are using BOOTP or RARP to resolve the address of the new router, the minimal configuration file must also include the IP address to be obtained dynamically using BOOTP or RARP.

You can use the **write network** command to help you generate the configuration file that you will download during the AutoInstall process.

Note The existing router might need to forward TFTP requests and response packets if the TFTP server is not on the same network segment as the new router. When you modified the existing router's configuration, you specified an IP helper address for this purpose.

You can save a minimal configuration under a generic newrouter-confg file. Use the **ip host** command in the network-confg or cisconet.cfg file to specify *newrouter* as the host name with the address you will be dynamically resolving. The new router should then resolve its IP address, host name and minimal configuration automatically. Use Telnet to connect to the new router from the existing router and use the **setup** facility to configure the rest of the interfaces. For example, the line in the network-confg or cisconet.cfg file could be similar to the following:

```
ip host newrouter 131.108.170.1
```

The following host configuration file contains the minimal set of commands needed for AutoInstall using SLARP or BOOTP:

```
enable password letmein
!
line vty 0
password letmein
!
end
```

The preceding example shows a minimal configuration for connecting from a router one hop away. From this configuration, use the **setup** facility to configure the rest of the interfaces. If the router is more than one hop away, you also must include routing information in the minimal configuration.

The following minimal network configuration file maps the new router's IP address, 131.108.10.2, to the host name *newrouter*. The new router's address was learned via SLARP and is based on the existing router's IP address of 131.108.10.1.

```
ip host newrouter 131.108.10.2
```

Set Up the BOOTP or RARP Server

If the new router is connected to the existing router using an Ethernet, Token Ring, or FDDI interface, you must configure a BOOTP or RARP server to map the new router's MAC address to its IP address. If the new router is connected to the existing router using a serial line with HDLC encapsulation or if you are configuring AutoInstall over Frame Relay, the tasks in this section are not required.

To configure a BOOTP or RARP server, complete one of the following tasks:

Task	Command
If BOOTP is to be used to resolve the new router's IP address, configure your BOOTP server.	Refer to your host vendor's manual pages and to RFCs 951 and 1395
If RARP is to be used to resolve the new router's IP address, configure your RARP server.	Refer to your host vendor's manual pages and to RFC 903

Note If the RARP server is not on the same subnet as the new router, use the **ip rarp-server** command to configure the existing router to act as a RARP server. See the section "Configure a Router as a RARP Server" in the "Loading System Images, Microcode Images, and Configuration Files" chapter of this manual.

The following host configuration file contains the minimal set of commands needed for AutoInstall using RARP. It includes the IP address that will be obtained dynamically via BOOTP or RARP during the AutoInstall process. When RARP is used, this extra information is needed to specify the proper netmask for the interface.

```
interface ethernet 0
ip address 131.108.10.2 255.255.255.0
enable password letmein
!
line vty 0
password letmein
!
end
```

Connect the New Router to the Network

Connect the new router to the network using either an HDLC-encapsulated or Frame Relay-encapsulated serial interface or an Ethernet, Token Ring, or FDDI interface. After the router successfully resolves its host name, *newrouter* sends a TFTP broadcast requesting the file *name*-confg or *name*.cfg. The router name must be in all lowercase, even if the true host name is not. The file is downloaded to the new router where the configuration commands take effect immediately. If the configuration file is complete, the new router should be fully operational. To save the complete configuration to NVRAM, complete the following tasks in privileged EXEC mode:

Task		Command
Step 1	Enter privileged mode at the system prompt on the new router.	**enable** *password*[1]
Step 2	Save the information from the *name*-config file to your startup configuration. On most platforms, this step saves the configuration to NVRAM. On the Cisco 7500 series, this step saves the configuration to the location specified by the CONFIG_FILE environment variable.	**write memory**

[1] This command is documented in the “User Interface Commands” chapter in the *Router Products Command Reference* publication.

Caution Verify that the existing and new routers are connected before entering the **write memory** EXEC command to save configuration changes. Use the **ping** EXEC command to verify connectivity. If an incorrect configuration file is downloaded, the new router will load NVRAM configuration information before it can enter AutoInstall mode.

If the configuration file is a minimal configuration file, the new router comes up, but with only one interface operational. Complete the following steps to connect to the new router and configure it:

Task		Command
Step 1	Establish a Telnet connection to the existing router.	**telnet** *existing* [1]
Step 2	From the existing router, establish a Telnet connection to the new router.	**telnet** *newrouter*
Step 3	Enter privileged EXEC mode.	**enable** *password*
Step 4	Enter **setup** mode to configure the new router.	**setup** [2]

[1] This command and the **enable** command are documented in the “User Interface Commands” chapter in the *Router Products Command Reference* publication.

[2] This command is documented in the *Router Products Getting Started Guide.*

Format Flash Memory on Cisco 7500 Series

On the Cisco 7500 series, 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 system requires a monlib file for the format operation. The monlib file is the ROM monitor library. The ROM monitor uses the monlib file to access files in the Flash file system.

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 of a Route Switch Processor (RSP) card on a Cisco 7500 series.

```
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

You also format a Flash memory card to recover from locked blocks. 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 to reformat 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.

Enter Configuration Mode

To enter configuration mode, enter the EXEC command **configure** at the privileged-level EXEC prompt. The router responds with the following prompt asking you to specify the terminal, NVRAM (memory), or a file stored on a network server (network) as the source of configuration commands:

```
Configuring from terminal, memory, or network [terminal]?
```

Each of these three methods is described in the next three sections.

The router accepts one configuration command per line. You can enter as many configuration commands as you want.

You can add comments to a configuration file describing the commands you have entered. Precede a comment with an exclamation point (!). Comments are *not* stored in NVRAM or in the active copy of the configuration file. In other words, comments do not show up when you list the active configuration with the **write terminal** EXEC command or list the configuration in NVRAM with the **show configuration** EXEC command. Comments are stripped from the configuration file when it is loaded to the router. However, you can list the comments in configuration files stored on a TFTP or Maintenance Operation Protocol (MOP) server.

Configure the Router from the Terminal

To configure the router from the terminal, complete the following tasks:

Task		Command
Step 1	Enter configuration mode selecting the terminal option.	**configure terminal**
Step 2	Enter the necessary configuration commands.	See the appropriate chapter for specific configuration commands.
Step 3	Quit configuration mode.	**^Z**
Step 4	Save the configuration changes to your startup configuration. On most platforms, this step saves the configuration to NVRAM. On the Cisco 7500 series, this step saves the configuration to the location specified by the CONFIG_FILE environment variable.	**write memory**

In the following example, the router is configured from the terminal. The comment “The following command provides the router host name” identifies the purpose of the next command line. The **hostname** command changes the router name from router1 to router2. By pressing **^Z**, the user quits configuration mode. The command **write memory** loads the configuration changes into NVRAM.

```
Router1# configure terminal
Router1(config)# !The following command provides the router host name.
```

```
Router1(config)# hostname router2
^Z
Router2# write memory
```

NVRAM stores the current configuration information in text format as configuration commands, recording only nondefault settings. The memory is checksummed to guard against corrupted data.

As part of its startup sequence, the router startup software always checks for configuration information in NVRAM. If NVRAM holds valid configuration commands, the router executes the commands automatically at startup. If the router detects a problem with the NVRAM or the configuration it contains, it enters setup mode and prompts for configuration. Problems can include a bad checksum for the information in NVRAM or the absence of critical configuration information. See the publication *Troubleshooting Internetworking Systems* for troubleshooting procedures. See the *Router Products Getting Started Guide* for details on setup information.

On the Cisco 7500 series, the router startup software uses the configuration pointed to by the CONFIG_FILE environment variable to start up. When the CONFIG_FILE environment variable does not exist or is null (such as at first-time startup), the router uses NVRAM as the default startup device. When the router uses NVRAM to start up and the system detects a problem with NVRAM or the configuration it contains, the router enters **setup** mode. Refer to the *Router Products Getting Started Guide* for more information on the **setup** command facility. For more information on environment variables, refer to the "Cisco's Implementation of Environment Variables" section in this chapter.

Configure the Router from Memory

On all platforms *except* the Cisco 7500 series, you can configure the router to execute the commands located in NVRAM. On the Cisco 7500 series, the same command configures the router to execute the configuration specified by the CONFIG_FILE environment variable.

To configure the router to execute the commands located in NVRAM or to execute the configuration specified by the CONFIG_FILE environment variable, complete the following task in privileged EXEC mode:

Task	Command
Configure the router to execute the commands located in NVRAM. or On the Cisco 7500 series, configure the router to execute the configuration specified by the CONFIG_FILE environment variable.	**configure memory**

Configure the Router from the Network

You can configure the router from the network by copying a configuration from a network server to your running or startup configuration. The following two sections explain these tasks.

Copy a Configuration File Directly to the Running Configuration

You can configure the router by retrieving and modifying the configuration file stored on one of your network servers. To do so, complete the following tasks:

Task		Command
Step 1	Enter configuration mode with the network option.	**configure network** or **copy rcp running-config** **copy tftp running-config**
Step 2	At the system prompt, select a host or network configuration file. The network configuration file contains commands that apply to all network servers and terminal servers on the network. The host configuration file contains commands that apply to one network server in particular.	**host** or **network**
Step 3	At the system prompt, enter the optional IP address of the remote host from which you are retrieving the configuration file.	*ip-address*
Step 4	At the system prompt, enter the name of the configuration file or accept the default name.	*filename*
Step 5	Confirm the configuration filename that the system supplies.	y

In the following example, the router is configured from the file *tokyo-config* at IP address 131.108.2.155:

```
Router1# configure network
Host or network configuration file [host]?
IP address of remote host [255.255.255.255]? 131.108.2.155
Name of configuration file [tokyo-confg]?
Configure using tokyo-confg from 131.108.2.155? [confirm] y
Booting tokyo-confg from 131.108.2.155:!! [OK - 874/16000 bytes]
```

Copy a Configuration File Directly to the Startup Configuration

You can copy a configuration file directly to your startup configuration without affecting the running configuration. On all platforms except the Cisco 7500 series, this task loads a configuration file directly into NVRAM without affecting the running configuration.

On the Cisco 7500 series, this task loads a configuration file directly into the location specified by the CONFIG_FILE environment variable without affecting the running configuration. If the CONFIG_FILE environment variable specifies NVRAM, the command functions as on all other platforms.

To copy a configuration file directly to the startup configuration, perform the following task in EXEC mode:

Task	Command
Load a configuration file directly into NVRAM or directly into the location specified by the CONFIG_FILE environment variable.	**configure overwrite-network** or **copy rcp startup-config** **copy tftp startup-config**

Modify the Configuration Register Boot Field

The configuration register boot field determines whether or not the router loads an operating system image, and if so, where it obtains this system image. The following sections describe the router's process for using the configuration register boot field, your process for setting this field, and the tasks you must perform to modify the configuration register boot field.

How the Router Uses the Boot Field

The lowest four bits of the 16-bit configuration register (bits 3, 2, 1, and 0) form the boot field. The following boot field values determine if the router loads an operating system and where the router obtains the system image:

- When the entire boot field equals 0-0-0-0, the router does not load a system image. Instead, the router enters ROM monitor or "maintenance" mode from which you can enter ROM monitor commands to manually load a system image.
- When the entire boot field equals 0-0-0-1, the router loads the system image found in boot ROMs.
- When the entire boot field equals a value between 0-0-1-0 and 1-1-1-1, the router loads the system image specified by **boot system** commands in the startup configuration file. When the startup configuration file does not contain **boot system** commands, the router loads a default system image stored on a network server.

When loading a default system image from a network server, the router uses the configuration register settings to determine the default system image filename for booting from a network server. The router forms the default boot filename by starting with the word *cisco* and then appending the octal equivalent of the boot field number in the configuration register, followed by a hyphen (-) and the processor type name (cisco*nn-cpu*). See the appropriate hardware installation guide for details on the configuration register and default filename.

Setting the Boot Field

You must correctly set the configuration register boot field to ensure that your router loads the operating system image as you intend. To set the boot field, follow this general procedure:

Step 1 Obtain the current configuration register setting. This setting is a hexadecimal value.

Step 2 Modify the current configuration register setting to reflect the way in which you want the router to load a system image. To do so, change the least significant hexadecimal digit to one of the following:

- 0 to load the system image manually using the **boot** command in ROM monitor mode.
- 1 to load the system image from boot ROMs. On the Cisco 7500 series, this setting configures the system to automatically load the system image from bootflash.
- 2—F to load the system image from **boot system** commands in the startup configuration file or from a default system image stored on a network server.

For example, if the current configuration register setting is 0x101 and you want to load a system image from **boot system** commands in the startup configuration file, you would change the configuration register setting to 0x102.

Step 3 Reboot the router to make your changes to the configuration register take effect.

Perform the Boot Field Modification Tasks

You modify the boot field from either the hardware configuration register or the software configuration register, depending on the platform.

Use the hardware configuration register to modify the boot field of a

- CGS
- MGS
- AGS+
- Cisco 7000 series that contains boot ROMs earlier than Cisco IOS Release 10

The hardware configuration register can be changed only on the processor card or with dual in-line package (DIP) switches located at the back of the router. For information on modifying the hardware configuration register, refer to the appropriate hardware installation guide.

Use the software configuration register to modify the boot field of a

- Cisco 2000 series
- Cisco 2500 series
- Cisco 3000 series
- Cisco 4000 series
- Cisco 4500 series
- Cisco 7000 series that contains Cisco IOS Release 10 or later boot ROMs
- Cisco 7500 series

To modify the software configuration register boot field, complete the following tasks:

Task		Command
Step 1	Obtain the current configuration register setting.	**show version**
Step 2	Enter configuration mode, selecting the terminal option.	**configure terminal**
Step 3	Modify the existing configuration register setting to reflect the way in which you want the router to load a system image.	**config-register** *value*
Step 4	Exit configuration mode.	**^Z**
Step 5	Reboot the router to make your changes take effect.	**reload**

Use the **show version** EXEC command to list the current configuration register setting. In ROM monitor mode, use the **o** command to list the value of the boot field in the configuration register.

In the following example, the **show version** command indicates that the current configuration register is set so that the router does not automatically load an operating system image. Instead, it enters ROM monitor mode and waits for user-entered ROM monitor commands. The new setting instructs the router to a load a system image from commands in the startup configuration file or from a default system image stored on a network server.

```
Router1# show version
GS Software, Version 9.0(1)
Copyright (c) 1986-1992 by cisco Systems, Inc.
Compiled Fri 14-Feb-92 12:37

System Bootstrap, Version 4.3

Router1 uptime is 2 days, 10 hours, 0 minutes
```

```
System restarted by reload
System image file is unknown, booted via tftp from 131.108.13.111
Host configuration file is "thor-boots", booted via tftp from 131.108.13.111
Network configuration file is "network-confg", booted via tftp from
131.108.13.111

CSC3 (68020) processor with 4096K bytes of memory.
X.25 software.
Bridging software.
1 MCI controller (2 Ethernet, 2 Serial).
2 Ethernet/IEEE 802.3 interface.
2 Serial network interface.
32K bytes of non-volatile configuration memory.
Configuration register is 0x0

Router1# configure terminal
Router1(config)# config-register 0xF
^Z

Router1# reload
```

Specify the Startup System Image

You can enter multiple boot commands in the NVRAM configuration or in the BOOT environment variable to provide backup methods for loading a system image onto the router. There are three ways to load a system image:

- From Flash memory—Flash memory allows you to copy new system images without changing EPROMs. Information stored in Flash memory is not vulnerable to network failures that might occur when loading system images from servers.
- From a network server—In case Flash memory becomes corrupted, specifying a system image to be loaded from a network server using TFTP, rcp, or MOP provides a backup boot method for the router. For the Cisco 4500 series and the Cisco 7500 series, you can specify a bootstrap image to be loaded from a network server using TFTP or rcp.
- From ROM—In case of both network failure and Flash memory corruption, specifying a system image to be loaded from ROM provides a final backup boot method. System images stored in ROM may not always be as complete as those stored in Flash memory or on network servers.

Note The Cisco 7500 series cannot boot from ROM.

You can enter the different types of boot commands in any order in the NVRAM configuration or in the BOOT environment variable. If you enter multiple boot commands, the router tries them in the order they are entered.

Load from Flash Memory

Use the following sections to configure your Cisco 2500 series, Cisco 3000 series, Cisco 4000 series, Cisco 7000 series, Cisco 7500 series, AccessPro PC card, and AGS+ to boot from 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.

In the Cisco 7000 series, Flash memory is located on the Route Processor (RP) card. In the Cisco 7500 series, Flash memory is located on the RSP card or on a Flash memory card inserted in one of the PCMCIA slots (slot 0 or slot 1) of the RSP card. You can store or boot software images in Flash memory as necessary. Flash memory can reduce the effects of network failure by reducing dependency on files that can only be accessed over the network.

Note Booting from ROM is faster than booting from Flash memory. However, if you are booting from a network server, Flash memory is faster and more reliable.

What You Can Do from Flash Memory

Flash memory allows you to do the following:

- Copy the system image to Flash memory using TFTP.
- Copy the system image to Flash memory using rcp.
- For the Cisco 4500 and 7500 series, copy a bootstrap image to Flash memory using TFTP or rcp.
- Boot a router from Flash memory either automatically or manually.
- Copy the Flash memory image to a network server using TFTP or rcp.
- For the Cisco 4500 and 7500 series, copy the Flash memory bootstrap image to a network server using TFTP or rcp.

Flash Memory Features

Flash memory features include the following:

- Flash memory can be remotely loaded with multiple system software images through TFTP or rcp transfers (one transfer for each file loaded).
- On the Cisco 7000 series, 4 MB of Flash memory storage are provided.
- You can boot a router manually or automatically from a system software image stored in Flash memory. You can also boot directly from ROM, or you can boot from a network server using TFTP or rcp.
- Flash memory provides write protection against accidental erasing or reprogramming.

Security Precautions

Take the following precautions when loading from Flash memory:

- Flash memory provides write protection against accidental erasing or reprogramming. The write-protect jumper, located next to the Flash components on a Cisco 7000 series RP, can be removed to prevent reprogramming of internal Flash memory. You must install the jumper when programming is required. The Cisco 7500 series does not support such write protection.

- Flash memory cards contain a write protect switch that you can use to protect data. You must set the switch to *unprotected* to write data to the Flash memory card.
- The system image stored in Flash memory can be changed only from privileged EXEC level on the console terminal.

Flash Memory Configuration Process for Cisco 2500, Cisco 3000 Series, Cisco 4000 Series, Cisco 7000 Series, AccessPro, and AGS+

To configure your Cisco 2500, Cisco 3000 series, Cisco 4000 series, Cisco 7000 series, AccessPro, and AGS+ systems to boot from Flash memory, follow this general procedure. Refer to the appropriate hardware installation and maintenance publication for complete instructions on installing the hardware and for information about the jumper settings required for your configuration.

Step 1 On the Cisco 2500, Cisco 3000, and AccessPro, you cannot run the system from Flash memory and copy to it at the same time. Therefore, do *one* of the following:

- Partition Flash memory or use Flash load helper to allow the system to run from Flash memory while you copy to it. See the "Partition Flash Memory Using Dual Flash Bank" and "Use Flash Load Helper to Upgrade Software on Run-from-Flash Systems" sections for more information.
- Set the system to load and run a system image from boot ROMs. See the "Modify the Configuration Register Boot Field" section for more information.

Step 2 On the Cisco 2500, Cisco 3000, and AccessPro, if you ran the image from boot ROMs, reload the system image.

Step 3 Copy a system image to Flash memory using TFTP or rcp. See the "Copy System Images from a Network Server to Flash Memory Using TFTP" or "Copy System Images from Flash Memory to a Network Server Using rcp" section for more information on performing this step.

Step 4 Configure the system to automatically boot from the desired file in Flash memory. You might need to change the configuration register value. See the "Modify the Configuration Register Boot Field" section for more information on the modifying configuration register.

Step 5 Save your configurations.

Step 6 Power-cycle and reboot your system to ensure that all is working as expected.

Flash Memory Configuration Process for Cisco 4500 Series

To configure your Cisco 4500 router to boot from Flash memory using a bootstrap image, follow this general procedure:

Step 1 Copy the bootstrap image into Flash memory using rcp or TFTP. See the "Copy Bootstrap Images from a Network Server to Flash Memory Using rcp or TFTP" section for more information on performing this step.

Step 2 Configure your system to automatically boot from Flash memory. You might need to change the configuration register value. See the "Modify the Configuration Register Boot Field" section for more information on modifying the configuration register.

Step 3 Save your configurations.

Step 4 Reboot your system to ensure that all is working as expected.

Flash Memory Configuration Process for Cisco 7500 Series

For the Cisco 7500 series, the configuration process is similar to the previous two processes, except you can specify the Flash device that contains the rxboot image. When you receive your the Cisco 7500 series router, bootflash contains the rxboot image. You can change the location of this image to a Flash memory card inserted in a PCMCIA slot. To specify the rxboot image Flash device, you set the BOOTLDR environment variable.

Note When no BOOTLDR environment variable exists, the default rxboot image on the Cisco 7500 series is the first image file in bootflash.

The Cisco 7500 series configuration process is as follows:

Step 1 Set the BOOTLDR environment variable if you want to change the location of the rxboot image that ROM uses for booting.

Step 2 Optionally, use rcp or TFTP to update the system image that resides in bootflash or on one of the Flash memory cards inserted in a PCMCIA slot. Performing this step allows you to update a degraded system image with one that is not degraded.

Step 3 Configure your system to automatically boot from the desired file in Flash memory. You might need to change the configuration register value. See the "Modify the Configuration Register Boot Field" section for more information on modifying the configuration register.

Step 4 Save your configurations.

Step 5 Power-cycle and reboot your system to ensure that all is working as expected.

Perform Flash Memory Configuration Tasks

Flash memory configuration tasks discussed in this section include the following:

- Set the BOOTLDR Environment Variable (optional for the Cisco 7500 series)
- Configure the Router to Automatically Boot from an Image in Flash Memory

Set the BOOTLDR Environment Variable

To set the BOOTLDR environment variable on your Cisco 7500 series router, perform the following tasks, beginning in privileged EXEC mode:

Task		Command
Step 1	Verify that bootflash contains the rxboot image.	**dir** [**/all** | **/deleted**] [**/long**] [*device***:**][*filename*]
Step 2	Enter the configuration mode from the terminal.	**configure terminal**
Step 3	Set the BOOTLDR environment variable to specify the Flash device and filename of the rxboot image. This step modifies the runtime BOOTLDR environment variable.	**boot bootldr** *device***:***filename*
Step 4	Exit configuration mode.	**^Z**

Task		Command
Step 5	Save this runtime BOOTLDR environment variable to your startup configuration.	**copy running-config startup-config** or **write memory**
Step 6	Optionally, verify the contents of the BOOTLDR environment variable.	**show boot**

The following example sets the BOOTLDR environment variable to change the location of the rxboot image from internal Flash to slot 0.

```
Router# dir bootflash:
-#- -length- -----date/time------ name
1   620      May 04 1995 26:22:04 rsp-boot-m
2   620      May 24 1995 21:38:14 config2

7993896 bytes available (1496 bytes used)
Router# configure terminal
Router (config)# boot bootldr slot0:rsp-boot-m
^Z
Router# copy running-config startup-config
[ok]
Router# show boot
BOOT variable = slot0:rsp-boot-m
CONFIG_FILE variable = nvram:
Current CONFIG_FILE variable = slot0:router-config

Configuration register is 0x0

Router#
```

Configure the Router to Automatically Boot from an Image in Flash Memory

To configure a router (including the AGS+, Cisco 3000 series, Cisco 4000 series, Cisco 7000 series, and Cisco 7500 series systems) to automatically boot from an image in Flash memory, perform the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Enter the filename of an image stored in Flash memory.	**boot system flash** [*filename*] **boot system flash flash:**[*filename*] **boot system flash bootflash:**[*filename*] **boot system flash slot0:**[*filename*] **boot system flash slot1:**[*filename*] (Cisco 7500 series only)
Step 3	Set the configuration register to enable loading of the system image from Flash memory.	**config-register** *value*[1]
Step 4	Exit configuration mode.	**^Z**
Step 5	Save the configuration file to your startup configuration. On most platforms, this step saves the configuration to NVRAM. On the Cisco 7500 series, this step saves the configuration to the location specified by the CONFIG_FILE environment variable.	**write memory**

Task		Command
Step 6	Optionally, verify the contents of the startup configuration.	**show configuration**
Step 7	Power-cycle and reboot the system to ensure that all works as expected	**reload**

[1] Refer to the "Modify the Configuration Register Boot Field" section for more information on systems that can use this command to modify the software configuration register.

If you enter more than one image filename, the router tries them in the order entered.

If a filename already appears in the configuration file and you want to specify a new filename, remove the existing filename with the **no boot system flash** *filename* command.

Note The **no boot system** configuration command disables all **boot system** configuration commands regardless of argument. Specifying the **flash** keyword or the *filename* argument with the **no boot system** command disables only the commands specified by these arguments.

The following example shows how to configure the router to automatically boot from an image in Flash memory:

```
Router# configure terminal
Router (config)# boot system flash gsnew-image
^Z
Router# write memory
[ok]
Router# reload
[confirm]

%SYS-5-RELOAD: Reload requested
System Bootstrap, Version 4.6(0.16), BETA SOFTWARE
Copyright (c) 1986-1995 by cisco Systems
RP1 processor with 16384 Kbytes of memory
F3: 1871404+45476+167028 at 0x1000

Booting gsnew-image from flash memory RRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR [OK - 1916912/13767448 bytes]
F3: 1871404+45476+167028 at 0x1000

              Restricted Rights Legend

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

             cisco Systems, Inc.
              170 West Tasman Drive
              San Jose, California 95134
```

```
GS Software (GS7), Version 10.2,
Copyright (c) 1986-1995 by cisco Systems, Inc.
Compiled Thu 05-Nov-94 14:16 by mlw
```

Load from a Network Server

You can configure the router to load a system image from a network server using TFTP, rcp, or MOP to copy the system image file.

To do this, the configuration register boot field must be set to the correct value. See the “Modify the Configuration Register Boot Field” section. Use the **show version** command to list the current configuration register setting.

If you do not boot from a network server using MOP and you omit both the **tftp** and the **rcp** keywords, by default the system image that you specify is booted from a network server using TFTP.

Note If you are using a Sun workstation as a network server and TFTP to transfer the file, set up the workstation to enable verification and generation of UDP checksums. See the Sun documentation for details.

For increased performance and reliability, boot from a system image from a network server using rcp. The rcp implementation uses the Transmission Control Protocol (TCP), which ensures reliable delivery of data. If you boot the router from a network server using rcp, the router software searches for the system image on the server relative to the directory of the remote username, if the remote server has a directory structure, for example, as do UNIX systems.You cannot explicitly specify a remote username when you issue the boot command. Instead, the host name configured for the router is used.

You can also boot from a compressed image on a network server. One reason to use a compressed image is to ensure that there is enough memory available for storage. On routers that do not contain a run-from-ROM image in EPROM, when the router boots software from a network server, the image being booted and the running image both must fit into memory. If the running image is large, there might not be room in memory for the image being booted from the network server.

If there is not enough room in memory to boot a regular image from a network server, you can produce a compressed software image on any UNIX platform using the **compress** command. Refer to your UNIX platform’s documentation for the exact usage of the **compress** command.

To specify the loading of a system image from a network server, complete the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Specify the system image file to be booted from a network server using TFTP, rcp or MOP.	**boot system** [**tftp** | **rcp**] *filename* [*ip-address*] **boot system mop** *filename* [*mac-address*] [*interface*]
Step 3	Set the configuration register to enable loading of the system image from a network server.	**config-register** *value*[1]

Task		Command
Step 4	Exit configuration mode.	**^Z**
Step 5	Save the configuration changes to your startup configuration. On most platforms, this step saves the configuration to NVRAM. On the Cisco 7500 series, this step saves the configuration to the location specified by the CONFIG_FILE environment variable.	**write memory**

[1] Refer to the "Modify the Configuration Register Boot Field" section for more information on systems that can use this command to modify the software configuration register.

In the following example, the router is configured to use rcp to netboot from the *testme5.tester* system image file on a network server at IP address 131.108.0.1:

```
Router1# configure terminal
Router1(config)# boot system rcp testme5.tester 131.108.0.1
^Z
Router1# write memory
```

Load from ROM

To specify the use of the ROM system image as a backup to other boot instructions in the configuration file, complete the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Specify use of the ROM system image as a backup image.	**boot system rom**
Step 3	Set the configuration register to enable loading of the system image from ROM.	**config-register** *value*[1]
Step 4	Exit configuration mode.	**^Z**
Step 5	Save the configuration changes to your startup configuration. On most platforms, this step saves the configuration to NVRAM. On the Cisco 7500 series, this step saves the configuration to the location specified by the CONFIG_FILE environment variable.	**write memory**

[1] Refer to the "Modify the Configuration Register Boot Field" section for more information on systems that can use this command to modify the software configuration register.

In the following example, the router is configured to boot from ROM:

```
Router1# configure terminal
Router1(config)# boot system rom
Router (config)# config-register 0x0101
^Z
Router1# write memory
```

Use a Fault-Tolerant Booting Strategy

Occasionally network failures make booting from a network server impossible. To lessen the effects of network failure, consider the following booting strategy. After Flash is installed and configured, you might want to configure the router to boot in the following order:

1. Boot an image from Flash.
2. Boot an image from a system file on a network server.
3. Boot from ROM image.

This boot order provides the most fault-tolerant booting strategy. Perform the following tasks to allow the router to boot first from Flash, then from a system file from a network server, and finally from ROM:

Task		Command
Step 1	Enter configuration mode from the terminal	**configure terminal**
Step 2	Configure the router to boot from Flash memory.	**boot system flash** [*filename*] **boot system flash bootflash:**[*filename*] **boot system flash slot0:**[*filename*] **boot system flash slot1:**[*filename*] (Cisco 7500 series only)
Step 3	Configure the router to boot from a system filename.	**boot system** [**rcp** | **tftp**] *filename* [*ip-address*]
Step 4	Configure the router to boot from ROM.	**boot system rom**
Step 5	Set the configuration register to enable loading of the system image from a network server or Flash.	**config-register** *value* [1]
Step 6	Exit configuration mode.	**^Z**
Step 7	Save the configuration file to your startup configuration. On most platforms, this step saves the configuration to NVRAM. On the Cisco 7500 series, this step saves the configuration to the location specified by the CONFIG_FILE environment variable.	**write memory**

[1] Refer to the “Modify the Configuration Register Boot Field” section for more information on systems that can use this command to modify the software configuration register.

The following example illustrates the order of the commands needed to implement this strategy. In the example, the router is configured to first boot an internal Flash image called *gsxx*. Should that image fail, the router will boot the configuration file *gsxx* from a network server. If that method should fail, then the system will boot from ROM.

```
Router# configure terminal
Router(config)# boot system flash gsxx
Router(config)# boot system gsxx 131.131.101.101
Router(config)# boot system rom
Router(config)# config-register 0x010F
^Z
Router# write memory
[ok]
```

Using this strategy, a router has three alternative sources from which to boot. These alternative sources help lessen the negative effects of a failure on network or file server from which the system image is copied.

Specify the Startup Configuration File

Configuration files can be stored on network servers. You can configure the router to automatically request and receive two configuration files from the network server:

- Network configuration file
- Host configuration file

The first file the server attempts to load is the network configuration file. This network configuration file contains information that is shared among several routers. For example, it can be used to provide mapping between IP addresses and host names.

The second file the server attempts to load is the host configuration file. This file contains commands that apply to one router in particular. Both the network and host configuration files must reside on a network server reachable using TFTP, rcp, or MOP, and must be readable.

You can specify an ordered list of network configuration and host configuration filenames. The router scans this list until it successfully loads the appropriate network or host configuration file.

In addition to storing configuration files on a network servers, with the Cisco 7500 series, you can store configuration files in NVRAM and on Flash memory cards. The CONFIG_FILE environment variable specifies the device and filename of the configuration file to use during initialization. For more information on environment variables, refer to the "Cisco's Implementation of Environment Variables" section in this chapter.

You can set the CONFIG_FILE environment variable to specify the startup configuration on a Cisco 7500 series.

Download the Network Configuration File

To configure the router to download a network configuration file from a server upon restart, complete the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Enter the network configuration filename to download a file using TFTP, rcp, or MOP.	**boot network mop** *filename* [*mac-address*] [*interface*] **boot network** [**tftp** | **rcp**] *filename* [*ip-address*]
Step 3	Enable the router to automatically load the network file upon restart.	**service config**
Step 4	Exit configuration mode.	**^Z**
Step 5	Save the configuration changes to your startup configuration. On most platforms, this step saves the configuration to NVRAM. On the Cisco 7500 series, this step saves the configuration to the location specified by the CONFIG_FILE environment variable.	**write memory**

For step 2, if you do not specify a network configuration filename, the router uses the default filename *network-confg.* If you omit both the **tftp** and the **rcp** keywords, the router assumes that the you are using TFTP to transfer the file and that the server whose IP address you specify supports TFTP.

If you configure the router to download the network configuration file from a network server using rcp and the server has a directory structure as do UNIX systems, the router software searches for the system image on the server relative to the directory of the remote username. The router host name is used as the remote username.

You can specify more than one network configuration file. The router tries them in order until it loads one successfully. This procedure can be useful for keeping files with different configuration information loaded on a network server.

Download the Host Configuration File

To configure the router to download a host configuration file from a server upon restart, complete the following tasks. Step 2 is optional. If you do not specify a host configuration filename, the router uses its own name to form a host configuration filename by converting the router name to all lowercase letters, removing all domain information, and appending -confg. If no host name information is available, the router uses the default host configuration filename *router-confg*.

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Optionally, enter the host configuration filename to be download using MOP, rcp, or TFTP.	**boot host mop** *filename [mac-address] [interface]* **boot host** [**tftp** | **rcp**] *filename [ip-address]*
Step 3	Enable the router to automatically load the host file upon restart.	**service config**
Step 4	Exit configuration mode.	**^Z**
Step 5	Save the configuration changes to your startup configuration. On most platforms, this step saves the configuration to NVRAM. On the Cisco 7500 series, this step saves the configuration to the location specified by the CONFIG_FILE environment variable.	**write memory**
Step 6	Reset the router with the new configuration information.	**reload**

You can specify more than one host configuration file. The router tries them in order until it loads one successfully. This procedure can be useful for keeping files with different configuration information loaded on a network server.

In the following example, the router is configured to boot from the host configuration file *hostfile1* and from the network configuration file *networkfile1*:

```
Router1# configure terminal
Router1(config)# boot host hostfile1
Router1(config)# boot network networkfile1
Router1(config)# service config
^Z
Router1# write memory
```

If the network server fails to load a configuration file during startup, it tries again every ten minutes (the default setting) until a host provides the requested files. With each failed attempt, the network server displays a message on the console terminal. If the network server is unable to load the specified file, it displays the following message:

```
Booting host-confg... [timed out]
```

Refer to the *Troubleshooting Internetworking Systems* publication for troubleshooting procedures. If there are any problems with the configuration file pointed to in NVRAM, or if the configuration register is set to ignore NVRAM, the router will enter the **setup** command facility. See the *Router Products Getting Started Guide* for details on the **setup** command.

Download the CONFIG_FILE Environment Variable Configuration on Cisco 7500 Series

In addition to loading startup configuration files from a server, on the Cisco 7500 series, you can configure the router to load a startup configuration file specified by the CONFIG_FILE environment variable. To do so, complete the following tasks, beginning in EXEC mode:

Task		Command
Step 1	Copy the configuration file to the device from which the router will load the file upon restart.	**copy** **copy flash** **copy rcp** **copy running-config** **copy startup-config** **copy tftp**
Step 2	Enter configuration mode from the terminal.	**configure terminal**
Step 3	Set the CONFIG_FILE environment variable. This step modifies the runtime CONFIG_FILE environment variable.	**boot config** *device***:***filename*
Step 4	Exit configuration mode.	**^Z**
Step 5	Save this runtime CONFIG_FILE environment variable to your startup configuration.	**copy running-config startup-config** or **write memory**
Step 6	Optionally, verify the contents of the CONFIG_FILE environment variable.	**show boot**

When saving the runtime CONFIG_FILE environment variable to the startup configuration, the router saves a complete version of the configuration file to the location specified by the CONFIG_FILE environment variable and a distilled version to NVRAM. A distilled version is one that does not contain access list information. If NVRAM contains a complete configuration file, the router prompts you to confirm your overwrite of the complete version with the distilled version. If NVRAM contains a distilled configuration, the router does not prompt you for confirmation and proceeds with overwriting the existing distilled configuration file in NVRAM.

The following example copies the running configuration file to the first PCMCIA slot of the RSP card in a Cisco 7500 series. This configuration is then used as the startup configuration when the system is restarted.

```
Router# copy running-config slot0:config2
Router# configure terminal
Router (config)# boot config slot0:config2
^Z
Router# copy running-config startup-config
[ok]
Router# show boot
BOOT variable = slot0:rsp-boot-m
CONFIG_FILE variable = nvram:
Current CONFIG_FILE variable = slot0:config2
```

```
Configuration register is 0x010F

Router#
```

Schedule a Reload of the System Image

You may want to schedule a reload of the system image to occur on the router at a later time (for example, late at night or during the weekend when the router is used less), or you may want to synchronize a reload network-wide (for example, to perform a software upgrade on all routers in the network).

Note A scheduled reload must take place within approximately 24 days.

To configure the router to reload the Cisco IOS software at a later time, perform one of the following tasks in privileged EXEC command mode:

Task	Command
Schedule a reload of the software to take effect in the specified minutes or hours and minutes.	**reload in** [*hh***:**]*mm* [*text*]
Schedule a reload of the software to take place at the specified time (using a 24-hour clock).	**reload at** *hh***:***mm* [*month day* | *day month*] [*text*]

If you specify the month and day, the reload is scheduled to take place at the specified time and date. If you do not specify the month and day, the reload takes place at the specified time on the current day (if the specified time is later than the current time), or on the next day (if the specified time is earlier than the current time). Specifying 00:00 schedules the reload for midnight.

Note The **at** keyword can only be used if the system clock has been set on the router (either through NTP, the hardware calendar, or manually). The time is relative to the configured time zone on the router. To schedule reloads across several routers to occur simultaneously, the time on each router must be synchronized with NTP.

To display information about a previously scheduled reload or to determine if a reload has been scheduled on the router, perform the following task in EXEC command mode:

Task	Command
Display reload information including the time the reload is scheduled to occur, and the reason for the reload if it was specified when the reload was scheduled.	**show reload**

To cancel a previously scheduled reload, perform the following task in privileged EXEC command mode:

Task	Command
Cancel a previously scheduled reload of the software.	**reload cancel**

The following example illustrates how to use the **reload** command to reload the software on the router on the current day at 7:30 p.m.:

```
router#reload at 19:30
Reload scheduled for 19:30:00 UTC Wed Jun 5 1996 (in 2 hours and 25 minutes)
Proceed with reload? [confirm]
router#
```

The following example illustrates how to use the **reload** command to reload the software on the router at a future time:

```
router#reload at 02:00 jun 20
Reload scheduled for 02:00:00 UTC Thu Jun 20 1996 (in 344 hours and 53 minutes)
Proceed with reload? [confirm]
router#
```

The following example illustrates how to use the **reload cancel** command to stop a scheduled reload:

```
router #reload cancel
router#
***
*** --- SHUTDOWN ABORTED ---
***
```

Additional Cisco 3000 and Cisco 4000 Copying and Automatic Booting Features

On the Cisco 3000 and 4000 systems that do not run from Flash memory, the following automatic booting features are implemented:

- When the software is set for automatic booting, if there is a total boot failure, the system brings up the ROM image.
- When the software is set for automatic booting, the system retries the netboot commands in the system configuration file up to five more times. The timeouts between each consecutive attempt are 2 seconds, 4 seconds, 16 seconds, 256 seconds, and 300 seconds. See the section about loading from a network server in this chapter.
- When the software is set for automatic booting, if all boot commands in the system configuration are netbooting commands and they all fail, the system attempts to boot the first valid file in Flash memory.
- When the software is set for automatic booting and the boot commands specified in the configuration file fail, one of the following occurs:
  - If the default boot ROM software bit in the configuration register is ON, the system boots up the ROM image without any retries.
  - If the default boot ROM software bit in the configuration register is OFF, no image is loaded and the system stops and presents the ROM monitor prompt. From the ROM monitor prompt, you can manually boot the system.
- The system searches for boot filename in Flash memory. If a filename is not specified, the software searches through the entire Flash directory for a bootable file instead of looking at just the first file.
- The system attempts to recognize the boot file in Flash memory. If the file is recognized, the software decides whether it is bootable by performing the following checks:
  - For run-from-Flash images, determines whether it is loaded at the correct execution address.
  - For run-from-RAM images, determines whether the system has enough RAM to execute the image.
- Additional user interface features for copying include the following:
  - Separate source and destination filenames
  - Extensive confirmation prompts and warning messages

Change the Buffer Size for Loading Configuration Files

The buffer that holds the configuration commands is generally the size of NVRAM. Complex configurations may need a larger configuration file buffer size. To change the buffer size, complete the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Change the buffer size to use for netbooting a host or network configuration file.	**boot buffersize** *bytes*
Step 3	Exit configuration mode.	**^Z**
Step 4	Save the configuration changes to your startup configuration. On most platforms, this step saves the configuration to NVRAM. On the Cisco 7500 series, this step saves the configuration to the location specified by the CONFIG_FILE environment variable.	**write memory**

In the following example, the buffer size is set to 50000 bytes:

```
Router1# configure terminal
Router1(config)# boot buffersize 50000
^Z
Router1# write memory
```

Compress Configuration Files

On the Cisco 7000 series, Cisco 4000 series, Cisco 3000 series, and AGS+ routers that have NVRAM, you can compress configuration files. To do so, perform the following tasks:

Task		Command
Step 1	Install the new ROMs.	Refer to the appropriate hardware installation and maintenance publication.
Step 2	Specify that the configuration file is to be compressed.	**service compress-config**
Step 3	Enter the privileged EXEC mode.	**enable** [*password*][1]
Step 4	Enter the new configuration.	Use TFTP or rcp to copy the new configuration. If you try to load a configuration that is more than three times larger than the NVRAM size, the following error message is displayed: “[buffer overflow - *file-size*/*buffer-size* bytes].” or **configure terminal**
Step 5	Save the configuration changes to your startup configuration. On most platforms, this step saves the configuration to NVRAM. On the Cisco 7500 series, this step saves the configuration to the location specified by the CONFIG_FILE environment variable.	**write memory**

[1] This command is documented in the “User Interface Commands” chapter of the *Router Products Command Reference*.

Installing new ROMs is a one-time operation and is necessary only if you do not already have Cisco IOS Release 10 in ROM. Before you can load a configuration that is larger than the size of NVRAM, you must issue the **service compress-config** command. The **configure terminal** command works only if you have Release 10 boot ROMs. It does not work with Release 9.17 ROMs or earlier versions.

Stop Booting and Manually Load a System Image from ROM Monitor

If your router does not find a valid system image, or if its configuration file is corrupted at startup, and the configuration register is set to enter ROM monitor mode, the system might enter memory ROM monitor mode. From this mode, you can manually load a system image from Flash memory, from a network server file, or from ROM.

You can force the router to stop booting and enter ROM monitor mode by pressing the **Break** key during the first 60 seconds of startup.

Manually Boot from Flash

To manually boot from Flash memory, complete the following tasks:

Task		Command
Step 1	Restart the router.	**reload**
Step 2	Press the **Break** key during the first 60 seconds while the system is starting up.	Break[1]
Step 3	Manually boot the router from Flash.	**boot flash** [*filename*] **boot flash** [*device***:**] *partition-number***:**[*filename*] **boot** *device***:**[*filename*] (Cisco 7500 series only)

[1] This key will not work on the Cisco 7000 unless it has Cisco IOS Release 10 boot ROMs.

In the following example, the router is manually booted from Flash memory. Because the optional *filename* argument is absent, the first file in Flash memory is loaded.

```
> boot flash
F3: 1858656+45204+166896 at 0x1000

Booting gs7-k from flash memory RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR [OK -
1903912/13765276 bytes]
F3: 1858676+45204+166896 at 0x1000

              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
```

In the following example, the **boot flash** command is used with the filename *gs7-k*—the name of the file that is loaded:

```
> boot flash gs7-k
F3: 1858656+45204+166896 at 0x1000
```

```
Booting gs7-k from flash memory RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRR [OK - 1903912/13765276 bytes]
F3: 1858676+45204+166896 at 0x1000

              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
System Bootstrap, Version 4.6(1012) [mlw 99], INTERIM SOFTWARE
Copyright (c) 1986-1992 by cisco Systems
RP1 processor with 16384 Kbytes of memory
```

Manually Boot from a Network File

To manually boot from a network file, complete the following tasks in EXEC mode:

Task		Command
Step 1	Restart the router.	**reload**
Step 2	Press the **Break** key during the first 60 seconds while the system is starting up.	Break[1]
Step 3	Manually boot the router.	**b** *filename* [*ip-address*]

[1] This key will not work on the Cisco 7000 unless it has Cisco IOS Release 10 boot ROMs.

In the following example, the router is manually booted from the network file *network1*:

```
>b network1
```

Manually Boot from ROM

To manually boot the router from ROM, complete the following steps in EXEC mode:

Task		Command
Step 1	Restart the router.	**reload**
Step 2	Press the **Break** key during the first 60 seconds while the system is starting up.	Break[1]
Step 3	Manually boot the router from ROM.	**boot**

[1] This key will not work on the Cisco 7000 unless it has Cisco IOS Release 10 boot ROMs.

On the Cisco 7500 series, the **boot** command loads the first bootable image located in bootflash.

In the following example, the router is manually booted from ROM:

```
>boot
```

Manually Boot Using MOP

You can interactively boot system software using MOP. Typically, you would do this to verify that system software has been properly installed on the MOP boot server before configuring the router to automatically boot the system software image.

To manually boot the router using MOP, perform the following tasks in EXEC mode:

Task		Command
Step 1	Restart the router.	**reload**
Step 2	Press the **Break** key during the first 60 seconds while the system is starting up.	Break[1]
Step 3	Manually boot the router using MOP.	**boot mop** *filename* [*mac-address*] [*interface*]

[1] This key will not work on the Cisco 7000 unless it has Cisco IOS Release 10 boot ROMs.

The Cisco 7500 series does not support the **boot mop** command.

In the following example, the router is manually booted from a MOP server:

```
>boot mop network1
```

Use the System Image Instead of Reloading

To return to EXEC mode from the ROM monitor to use the system image instead of reloading, perform the following task in ROM monitor mode:

Task	Command
Return to EXEC mode to use the system image.	**continue**

Boot Systems That Have Dual-Bank Flash Memory

The AccessPro card and the Cisco 4500 have two banks of Flash memory on one SIMM, referred to as dual-bank Flash memory. The dual-bank Flash SIMM is used for storing system images.

Dual-bank Flash provides partitioning support on systems that can accommodate only one Flash SIMM device. See the "Partition Flash Memory Using Dual Flash Bank" section for information about how to partition Flash memory.

Copy a Boot Image on a Cisco 4500

You can retrieve a boot image from a TFTP, rcp, or MOP server. This image is copied into boot Flash memory. You can also copy the boot image from the boot Flash to a TFTP or rcp server.

To retrieve a boot image from a TFTP or rcp server, perform the following task in EXEC mode:

Task	Command
Copy a boot image from a TFTP or rcp server.	**copy** [**tftp** | **rcp**] **bootflash**

To retrieve a boot image from a MOP server, perform the following task in EXEC mode:

Task	Command
Copy a boot image from a MOP server.	**copy mop bootflash**

To copy a boot image from boot Flash to a TFTP server, perform the following task in EXEC mode:

Task	Command
Copy a boot image to a TFTP server.	**copy bootflash tftp**

Verify a Boot Image’s Checksum on a Cisco 4500

To verify the checksum of a boot image in Flash memory, perform the following task in EXEC mode:

Task	Command
Verify the checksum of a boot image.	**copy verify bootflash**

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**

Configure 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 router’s configuration.

To specify TFTP server operation for a router, complete the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Specify TFTP server operation.	**tftp-server flash** [*partition-number***:**]*filename1* [**alias** *filename2*] [*access-list-number*] **tftp-server rom alias** *filename1* [*access-list-number*] **tftp-server flash** *device***:***filename* (Cisco 7500 series only)
Step 3	Exit configuration mode.	**^Z**
Step 4	Save the configuration file to your startup configuration. On most platforms, this step saves the configuration to NVRAM. On the Cisco 7500 series, this step saves the configuration to the location specified by the CONFIG_FILE environment variable.	**write memory**

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.

The transfer session might still succeed even if TFTP generates these characters, but the output is useful for diagnosing the transfer failure. For troubleshooting procedures, refer to the *Troubleshooting Internetworking Systems* publication.

In the following example, the system uses TFTP to send a copy 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 uses 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
```

Configure a Router to Support Incoming rcp Requests and rsh Commands

You configure a local authentication database to control access to the router by remote users. A local authentication database is similar in concept and use to a UNIX *.rhosts* file. To allow remote users to execute rcp or rsh commands on the router, configure entries for those users in the router's authentication database.

Each entry configured in the authentication database identifies the local user, the remote host, and the remote user. To be allowed to remotely execute commands on the router, the remote user must specify all three values—the local username, the remote host name, and the remote username—and therefore must be apprised of the local username. For rsh users, you can also grant a user permission to execute privileged EXEC commands remotely.

An entry that you configure in the router authentication database differs from an entry in a UNIX *.rhosts* file in several ways, the most salient of which is the inclusion of a local username. Because the *.rhosts* file on a UNIX system resides in the home directory of a local user account, an entry in a UNIX *.rhosts* file does not need to include the local username. The local username is determined from the user account. Because our routers do not inherently support the concept of accounts, you must specify the local username along with the remote host name and the remote username in each authentication database entry that you configure. You can specify the router host name as the local username.

To make the local username available to remote users, you need to communicate the username to the network administrator or the remote user. To allow a remote user to execute a command on the router, our rcp implementation requires that the local username sent by the remote user match the local username configured in the database entry.

The router software uses DNS to authenticate the remote host's name and address. Because DNS can return several valid IP addresses for a host name, the router software checks the address of the requesting client against all of the IP addresses for the named host returned by DNS. If the address sent by the requester is considered invalid, that is, it does not match any address listed with DNS for the host name, then the router software will reject the remote-command execution request.

Note that if no DNS servers are configured for the router, then the router cannot authenticate the host in this manner. In this case, the router software will send a broadcast request to attempt to gain access to DNS services on another server. If DNS services are not available, you must use the **no ip domain-lookup** command to disable the router's attempt to gain access to a DNS server by sending a broadcast request.

If DNS services are not available and, therefore, you bypass the DNS security check, the router software will accept the request to remotely execute a command *only if* all three values sent with the request match exactly the values configured for an entry in the local authentication file.

If DNS is enabled but you do not want to use DNS for rcmd queries, use the **no ip rcmd domain-lookup** command.

To ensure security, the router is *not* enabled to support rcp requests from remote users by default. When the router is not enabled to support rcp, the authorization database has no effect.

To configure the router to allow users on remote systems to copy files to and from the router and execute commands on the router, perform the tasks in one of the first sections and, if desired, the task in the third section:

- Configure the Router to Accept rcp Requests from Remote Users
- Configure the Router to Allow Remote Users to Execute Commands Using rsh
- Turn Off DNS Lookups for rcp and rsh

Configure the Router to Accept rcp Requests from Remote Users

To configure the router to support incoming rcp requests, complete the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Create an entry in the local authentication database for each remote user who is allowed to execute rcp commands on the router.	**iip rcmd remote-host** *local-username* {*ip-address* | *host*} *remote-username* [**enable** [*level*]]
Step 3	Enable the router to support incoming rcp requests.	**ip rcmd rcp-enable**

To disable the router from supporting incoming rcp requests, use the **no ip rcmd rcp-enable** command.

Note When the router's support for incoming rcp requests is disabled, you can still use the rcp commands to copy images from remote servers. The router's support for incoming rcp requests is distinct from its capacity for outgoing rcp requests.

The following example shows how to add two entries for remote users to the router's authentication database and then enable the router to support remote copy requests from remote users. The users, named *netadmin1* on the remote host at IP address 131.108.15.555 and *netadmin3* on remote host at IP address 131.108.101.101, 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 router's host name *Router1* as the local username. The fourth command enables the router to support for rcp requests from remote users.

```
configure terminal
ip rcmd remote-host Router1 131.108.15.55 netadmin1
ip rcmd remote-host Router1 131.108.101.101 netadmin3
ip rcmd rcp-enable
```

Note In Cisco IOS Release 10.3, the **ip** keyword has been added to **rcmd** commands. If you are upgrading from Release 10.2 to 10.3, this keyword is automatically added to any **rcmd** commands you have in your Release 10.2 configuration files.

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

To configure the router as an rsh server, complete the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Create an entry in the local authentication database for each remote user who is allowed to execute rsh commands on the router.	**ip rcmd remote-host** *local-username* {*ip-address* | *host*} *remote-username* [**enable** [*level*]]
Step 3	Enable the router to support incoming rsh commands.	**ip rcmd rsh-enable**

To disable the router from supporting incoming rsh commands, use the **no ip rcmd rsh-enable** command.

Note When the router is disabled, you can still issue a remote shell command to be executed on other routers that support the remote shell protocol and on UNIX hosts on the network.

The following example shows how to add two entries for remote users to the router's authentication database, and enable the router to support rsh commands from remote users. The users, named *rmtnetad1* and *netadmin4*, are both on the remote host at IP address 131.108.101.101. 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 fourth command enables the router for to support rsh commands issued by remote users.

```
configure terminal
ip rcmd remote-host Router1 131.108.101.101 rmtnetad1
ip rcmd remote-host Router1 131.108.101.101 netadmin4 enable
ip rcmd rsh-enable
```

Turn Off DNS Lookups for rcp and rsh

To bypass the DNS security check when DNS services are configured but not available, perform the following task in global configuration mode:

Task	Command
Bypass the DNS security check.	**no ip rcmd domain-lookup**

The router software will accept the request to remotely execute a command only if all three values sent with the request match exactly the values configured for an entry in the local authentication file.

Configure a Router as a RARP Server

You can configure the router as a RARP server. With this feature, RARP requests can be answered by the router, thereby allowing the router to make possible diskless booting of various systems, such as Sun workstations or PCs, on networks where the client and server are on separate subnets.

To configure the router as a RARP server, perform the following task in interface configuration mode:

Task	Command
Configure the router as a RARP server.	**ip rarp-server** *ip-address*

In the following example, the router is configured to act as a RARP server. Figure 3-4 illustrates the network configuration.

Figure 3-4 Configuring a Router as a RARP Server

```
! 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 128.105.2.5 0800.2002.ff5b arpa
interface ethernet 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 128.105.3.100
! Portmapper broadcasts from this interface are sent to the Sun Server.
ip helper-address 128.105.3.100
```

The Sun client and server machine’s IP addresses must use the same major network number due to a limitation of the current SunOS rpc.bootparamd daemon.

Configure the Remote Username for rcp Requests

From the router, you can use rcp to remotely copy files to and from network servers and hosts if those systems support rcp. You do not need to configure the router to issue remote copy requests from the router using rcp. However, to prepare to use rcp from the router for remote copying, you can perform an optional configuration process to specify the remote username to be sent on each rcp request.

The rcp protocol requires that a client send the remote username on an rcp request. By default, the router software sends the remote username associated with the current TTY (terminal) process, if that name is valid, for rcp commands.

Note For Cisco, TTYs are commonly used in communications servers. The concept of TTYs 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).

If the username for the current TTY process is not valid, the router software sends the host name as the remote username. For boot commands using rcp, the router software sends the router host name by default. You cannot explicitly configure the remote username.

If the remote server has a directory structure, for example, as do UNIX systems, rcp performs its copy operations along the following lines:

- When copying from the remote server, rcp searches for the system image or configuration file to be copied relative to the directory of the remote username.
- When copying to the remote server, rcp writes the system image or configuration file to be copied relative to the directory of the remote username.
- When booting an image, rcp searches for the image file on the remote server relative to the directory of the remote username.

To override the default remote username sent on rcp requests, complete the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal	**configure terminal**
Step 2	Specify the remote username.	**ip rcmd remote-username** *username*

To remove the remote username and return to the default value, use the **no ip rcmd remote-username** command.

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.

Specify Asynchronous Interface Extended BOOTP Requests

The Boot Protocol (BOOTP) server for asynchronous interfaces supports the extended BOOTP requests specified in RFC 1084. The following command is useful in conjunction with using the auxiliary port as an asynchronous interface.

To configure extended BOOTP requests for asynchronous interfaces, perform the following task in global configuration mode:

Task	Command
Configure extended BOOTP requests for asynchronous interfaces.	**async-bootp** *tag* [:*hostname*] *data*

You can display the extended BOOTP requests by performing the following task in EXEC mode:

Task	Command
Show parameters for BOOTP requests.	**show async-bootp**

Specify MOP Server Boot Requests

To change the router's parameters for retransmitting boot requests to a MOP server, complete the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Change MOP server parameters.	**mop device-code {cisco | ds200**} **mop retransmit-timer** *seconds* **mop retries** *count*
Step 3	Exit configuration mode.	**^Z**
Step 4	Save the configuration changes to your startup configuration. On most platforms, this step saves the configuration to NVRAM. On the Cisco 7500 series, this step saves the configuration to the location specified by the CONFIG_FILE environment variable.	**write memory**

By default, when the router transmits a request that requires a response from a MOP boot server and the server does not respond, the message will be retransmitted after four seconds. If the MOP boot server and router are separated by a slow serial link, it may take longer than four seconds for the router to receive a response to its message. Therefore, you might want to configure the router to wait longer than four seconds before retransmitting the message if you are using such a link.

In the following example, if the MOP boot server does not respond within 10 seconds after the router sends a message, the router will retransmit the message:

```
Router# configure terminal
Router (config)# mop retransmit-timer 10
^Z
Router# write memory
```

Copy System Images from a Network Server to Flash Memory Using TFTP

To copy a system image from a TFTP server to Flash memory, complete the following tasks:

Task		Command
Step 1	Make a backup copy of the current system software image.	See the instructions in the section “Copy System Images from Flash Memory to a Network Server Using TFTP” later in this chapter.
Step 2	Copy a system image to Flash memory.	**copy tftp flash** **copy tftp** *file_id* (Cisco 7500 series only)
Step 3	When prompted, enter the IP address or domain name of the server.	*ip-address* or *name*
Step 4	When prompted, enter the filename of the server system image.	*filename*

Note Be sure there is ample space available before copying a file to Flash memory. Use the **show flash** command and compare the size of the file you want to copy to the amount of available Flash memory shown. If the space available is less than the space required by the file you want to copy, the copy process will continue, but the entire file will not be copied into Flash. The failure message “buffer overflow - *xxxx*/*xxxx*,” will appear, where *xxxx/xxxx* is the number of bytes read in relation to the number of bytes available.

The server system image copied to the Flash memories for the AGS+, AGS, MGS, and CGS must be at least Software Version 9.0 or later. For Cisco 3000, Cisco 4000, and Cisco 7000 series, the server system image must be at least Software Version 9.1 or later.

Note When you are upgrading or changing to a different Cisco IOS release, refer to the appropriate release notes for information on system requirements and limitations.

When you issue the **copy tftp flash** command, the system prompts you for the IP address or domain name of the TFTP server. This server can be another router serving ROM or Flash system software images. The system then prompts you for the filename of the software image to copy.

For the **copy tftp flash** and **copy tftp** *file_id* commands, when there is free space available in Flash memory, you are given the option of erasing the existing Flash memory before writing onto it. If no free Flash memory space is available, or if the Flash memory has never been written to, the erase routine is required before new files can be copied. The system will inform you of these conditions and prompt you for a response. Note that on the AGS+ and Cisco 7000 series, the Flash memory is erased at the factory before shipment.

The *file_id* argument of the **copy tftp** *file_id* command specifies a device and filename as the destination of the copy operation. You can omit the device, entering only **copy tftp** *filename*. When you omit the device, the system uses the default device specified by the **cd** command. On the Cisco 7500 series, you can choose **bootflash:**, **slot0:**, or **slot1:** as the Flash memory device.

If you attempt to copy a file into Flash memory that is already there, a prompt informs you that a file with the same name already exists. This file is “deleted” when you copy the new file into Flash. The first copy of the file still resides within Flash memory, but it is rendered unusable in favor of the newest version, and is listed with the “deleted” tag when you use the **show flash** command. If you terminate the copy process, the newer file is marked “deleted” because the entire file was not copied and is not valid. In this case, the original file in Flash memory is valid and available to the system.

Following is sample output (copying a system image named *gs7-k*) of the prompt you will see when using the **copy tftp flash** command when Flash memory is too full to copy the file. The filename *gs7-k* can be in either lowercase or uppercase; the system will see *GS7-K* as *gs7-k*. If more than one file of the same name is copied to Flash, regardless of case, the last file copied will become the valid file.

```
env-chassis# copy tftp flash
IP address or name of remote host [255.255.255.255]? dirt
Translating "DIRT"...domain server (255.255.255.255) [OK]

Name of file to copy ? gs7-k
Copy gs7-k from 131.108.13.111 into flash memory? [confirm]
Flash is filled to capacity.
Erasure is needed before flash may be written.
Erase flash before writing? [confirm]
Erasing flash EPROMs bank 0

Zeroing bank...zzzzzzzzzzzzzzzz
Verify zeroed...vvvvvvvvvvvvvvvv
Erasing bank...eeeeeeeeeeeeeeee

Erasing flash EPROMs bank 1

Zeroing bank...zzzzzzzzzzzzzzzz
Verify zeroed...vvvvvvvvvvvvvvvv
Erasing bank...eeeeeeeeeeeeeeee

Erasing flash EPROMs bank 2

Zeroing bank...zzzzzzzzzzzzzzzz
Verify zeroed...vvvvvvvvvvvvvvvv
Erasing bank...eeeeeeeeeeeeeeee

Erasing flash EPROMs bank 3

Zeroing bank...zzzzzzzzzzzzzzzz
Verify zeroed...vvvvvvvvvvvvvvvv
Erasing bank...eeeeeeeeeeeeeeee

Loading from 131.108.1.111:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 [OK - 1906676/4194240 bytes]
Verifying via checksum...
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvv
Flash verification successful. Length = 1906676, checksum = 0x12AD
```

Note If you enter **n** after the “Erase flash before writing?” prompt, the copy process continues. If you enter **y**, the erase routine begins. Make certain you have ample Flash memory space before entering **n** at the erasure prompt.

Following is sample output from copying a system image named *gs7-k* into the current Flash configuration, in which a file of the name *gs7-k* already exists:

```
env-chassis# copy tftp flash
IP address or name of remote host [131.108.13.111]?
Name of file to copy ? gs7-k
File gs7-k already exists; it will be invalidated!
Copy gs7-k from 131.108.13.111 into flash memory? [confirm]
2287500 bytes available for writing without erasure.
Erase flash before writing? [confirm]n
Loading from 131.108.1.111:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 1906676/2287500 bytes]
Verifying via checksum...
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvv
Flash verification successful. Length = 1902192, checksum = 0x12AD
```

In the following example, the Flash security jumper is not installed, so you cannot write files to Flash memory.

```
Everest# copy tftp flash
Flash: embedded flash security jumper(12V)
       must be strapped to modify flash memory
```

Note To terminate this copy process, press **Ctrl**-^ (the Ctrl, Shift, and 6 keys on a standard keyboard) simultaneously. Although the process will terminate, the partial file copied before the termination will remain until the entire Flash memory is erased. Refer to the *Troubleshooting Internetworking Systems* publication for procedures on how to resolve Flash memory problems.

On the Cisco 7500 series, the following example copies the *router-config* file from a TFTP server to the Flash memory card inserted in slot 0 of the Route Switch Processor (RSP) card. The copied file has the name *new-config*.

```
Router# copy tftp:router-config slot0:new-config
```

You can copy normal or compressed images to Flash memory. You can produce a compressed system image on any UNIX platform using the **compress** command. Refer to your UNIX platform's documentation for the exact usage of the **compress** command.

The following example shows sample output from copying a system image named IJ09140Z into the current Flash configuration.

```
Router# copy tftp flash
IP address or name of remote host [255.255.255.255]? server1
Name of tftp filename to copy into flash []? IJ09140Z
copy IJ09140Z from 131.131.101.101 into flash memory? [confirm] <Return>
xxxxxxxx bytes available for writing without erasure.
erase flash before writing? [confirm] <Return>
Clearing and initializing flash memory (please wait)####...
Loading from 101.2.13.110: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!... [OK - 324572/524212 bytes]
Verifying checksum...
```

```
VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV...
Flash verification successful. Length = 1204637, checksum = 0x95D9
```

The series of pound signs (#) indicates that each Flash device is being cleared and initialized; one per device. Different router platforms use different ways of indicating that Flash is being cleared. The exclamation points (!) indicate the copy process. The series of Vs indicates that a checksum is calculated. An O would have indicated an out-of-order packet. A period (.) would have indicated a timeout. The last line in the sample configuration indicates that the copy is successful.

Copy System Images to Flash Memory Using MOP

On all platforms except the Cisco 7500 series, you can copy a system image from a MOP server to Flash memory. To do so, perform the following task in EXEC mode:

Task	Command
Copy a boot image using MOP.	**copy mop flash**

Note When you are upgrading or changing to a different Cisco IOS release, refer to the appropriate release notes for information on system requirements and limitations.

The following example shows a sample output from the **copy mop flash** command. In this example, the system image *junk*, which already exists in Flash memory, is copied to Flash memory, and there is enough memory to copy the file without erasing any existing files.

```
Router# copy mop flash

System flash directory:
File  Length   Name/status
  1   984      junk [deleted]
  2   984      junk
[2096 bytes used, 8386512 available, 8388608 total]
Source file name? junk
Destination file name [junk]?

Erase flash device before writing? [confirm]
Flash contains files. Are you sure you want to erase? [confirm]

Copy 'junk' from server
  as 'junk' into Flash WITH erase? [yes/no]yes
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
Loading junk from 1234.5678.9abc via Ethernet0: !
[OK - 984/8388608 bytes]

Verifying checksum...  OK (0x14B3)
Flash copy took 0:00:01 [hh:mm:ss]
```

Copy System Images from a Network Server to Flash Memory Using rcp

You can copy a system image from a network server to Flash memory using rcp. For the rcp command to execute properly, an account must be defined on the network server for the remote username. You can override the default remote username sent on the rcp copy request by configuring the remote username. 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. The rcp protocol implementation copies the system image from the remote server relative to the directory of the remote username if the remote server has a directory structure, for example, as do UNIX systems.

To copy a system image from a network server to Flash memory using rcp, complete the following tasks:

Tasks		Command
Step 1	Make a backup copy of the current system software image.	See the instructions in the section "Copy System Images from Flash Memory to a Network Server Using rcp" later in this chapter.
Step 2	Enter configuration mode from the terminal. This step is only required if you are going to override the default remote username (see Step 3).	**configure terminal**
Step 3	Specify the remote username. This step is optional, but recommended.	**ip rcmd remote-username** *username*
Step 4	Exit configuration mode.	**^Z**
Step 5	Copy the system image from the network server to Flash memory using rcp.	**copy rcp flash** **copy rcp** *file_id* (Cisco 7500 series only)
Step 6	When prompted, enter the IP address or domain name of the network server.	*ip-address* or *name*
Step 7	When prompted, enter the filename of the server system image to be copied.	*filename*

The **copy** command automatically displays the Flash memory directory, including the amount of free space. On Cisco 2500, Cisco 3000, and Cisco 4000 systems, if the file being downloaded to Flash memory is an uncompressed system image, the **copy** command automatically determines the size of the file being downloaded and validates it with the space available in Flash memory.

The server system image copied to the Flash memories must be Cisco IOS Release10.2 or later.

Note When you are upgrading or changing to a different Cisco IOS release, refer to the appropriate release notes for information on system requirements and limitations.

When you issue the **copy rcp flash** or **copy rcp** *file_id* command, the system prompts you for the IP address or domain name of the server. This server can be another router serving Flash system software images. The system then prompts you for the filename of the software image to copy. With the **copy rcp flash** command, the system also prompts you to name the system image file that will reside in Flash memory once the copy is complete. You can use the filename of the source file, or you can choose another name.

When free space is available in Flash memory, you are given the option of erasing the existing Flash memory before writing onto it. If no free Flash memory space is available, or if the Flash memory has never been written to, the erase routine is required before new files can be copied. The system informs you of these conditions and prompts you for a response. If you accept the erasure, the system prompts you again to confirm before erasing. Note that the Flash memory is erased at the factory before shipment.

If you attempt to copy a file into Flash memory that is already there, a prompt will tell you that a file with the same name already exists. The older file is "deleted" when you copy the new file into Flash. The first copy of the file still resides within Flash memory, but is rendered unusable in favor of the newest version, and will be listed with the "deleted" tag when you use the **show flash** command. If you abort the copy process, the newer file will be marked "deleted" because the entire file was not copied. In this case, the original file in Flash memory is valid and available to the system.

The following example copies a system image named *mysysim1* from the *netadmin1* directory on the remote server named *SERVER1.CISCO.COM* with an IP address of 131.108.101.101 to the router's Flash memory. To ensure that enough Flash memory is available to accommodate the system image to be copied, the router software allows you to erase the contents of Flash memory first.

```
Router1# configure terminal
Router1# ip rcmd remote-username netadmin1
^Z
Router# copy rcp flash

System flash directory:
File name/status
 1 mysysim1
[2076072 bytes used, 21080 bytes available]

Address or name of remote host[UNKNOWN]? 131.108.101.101
Name of file to copy? IJ09140Z
Copy IJ09140z from SERVER1.CISCO.COM?[confirm]

Checking for file 'mysysim1' on SERVER1.CISCO.COM...[OK]

Erase Flash device before writing?[confirm]
Are you sure?[confirm]
Erasing device...ezeeze...erased.

Connected to 131.108.101.101

Loading 2076007 byte file IJ09140Z:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!![OK]

Verifying checksum... (0x87FD)...[OK]
Router#
```

The exclamation points (!) indicate that the copy process is taking place. Each exclamation point (!) indicates that ten packets have been transferred successfully.

Note If you enter **n** after the "Erase Flash device before writing?" prompt, the copy process continues. If you enter **y** and you confirm the erasure, the erasing routine begins. Make certain you have ample Flash memory space before entering **n** at the erasure prompt.

You can copy normal or compressed images to Flash memory. You can produce a compressed system image on any UNIX platform using the **compress** command. Refer to your UNIX platform's documentation for the exact usage of the **compress** command.

Additional Cisco 3000 and Cisco 4000 Flash Upgrade Features

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 display warnings where necessary.

Copy Bootstrap Images from a Network Server to Flash Memory Using rcp or TFTP

For the Cisco 4500 router, you can copy a bootstrap image stored on a network server to Flash memory using rcp or TFTP. Before you perform the copy operation, back up the system image or bootstrap image in Flash memory to a network server.

Note When you are upgrading or changing to a different Cisco IOS release, refer to the appropriate release notes for information on system requirements and limitations.

The rcp protocol requires that a client send the remote username on each rcp request. When you copy a bootstrap image from a network server using rcp, the router software sends the remote username associated with the current TTY (terminal) process, if that name is valid. If the TTY username is invalid, the router software uses the router host name as the both the remote and local usernames. You can configure a different remote username to be sent to the server. The rcp protocol implementation searches for the bootstrap image to copy from the remote server relative to the directory of the remote username, if the remote server has a directory structure, for example, as do UNIX systems.

Note For Cisco, TTYs are commonly used in communications servers. The concept of TTYs 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).

For the rcp command to execute properly, an account must be defined on the destination server for the remote username.

When you request the bootstrap image to copy using rcp, the router prompts you for the name or address of the server and the name of the file to be copied. It provides an option to erase existing Flash memory before writing onto it, and allows you to confirm the erasure. The entire copying process takes several minutes and will differ from network to network.

To copy a bootstrap image from a network server to Flash memory on a Cisco 4500 router using either rcp or TFTP, complete the following tasks:

Tasks		Command
Step 1	Make a backup copy of the current system or bootstrap software image.	See the instructions in the section "Copy System Images from Flash Memory to a Network Server Using rcp" or the section "Copy System Images from Flash Memory to a Network Server Using TFTP" in this chapter.
Step 2	Enter configuration mode from the terminal. This step is required if you are going to override the default remote username (see Step 3).	**configure terminal**
Step 3	If the copy is performed using rcp, specify the remote username. This step is optional, but recommended.	**ip rcmd remote-username** *username*
Step 4	Exit configuration mode.	**^Z**
Step 5	Copy the bootstrap image from the network server to Flash memory using rcp or TFTP.	**copy** {**rcp** | **tftp**} **bootflash**
Step 6	When prompted, enter the IP address or domain name of the server.	*ip-address* or *name*
Step 7	When prompted, enter the filename of the bootstrap image to be copied from the server.	*filename*

Before booting the router from Flash memory, verify that the checksum of the bootstrap image in Flash memory matches the checksum listed in the README file that was distributed with the system software image. The checksum of the bootstrap image in Flash memory is displayed at the bottom of the display output when you issue the copy request. The README file was copied to the server automatically when you installed the system software.

Caution If the checksum value does not match the value in the README file, do not reboot the router. Issue the copy request and compare the checksums again. If the checksum is repeatedly wrong, copy the original bootstrap image back into Flash memory *before* you reboot the router from Flash memory. If you have a bad image in Flash memory and try to boot from Flash, the router will start the system image contained in ROM (assuming the system is not configured to boot from a network server).

If you use rcp to copy the bootstrap image from a personal computer used as a file server, the computer must support rsh. If you use TFTP to copy the bootstrap image from a personal computer used as a file server, the computer must be configured as a TFTP server.

The following example shows how to copy a bootstrap image from the server to Flash memory:

```
Router1# configure terminal
Router1# ip rcmd remote-username netadmin1
^Z
Router1# copy rcp bootflash

System flash directory:
File name/status
 1 btxx
[2076072 bytes used, 21080 bytes available]

Address or name of remote host[UNKNOWN]? 131.108.1.111
Name of file to copy? btxx
Copy btxx from UTOPIA.CISCO.COM?[confirm]

Checking for file ‘btxx’ on UTOPIA.CISCO.COM...[OK]

Erase flash device before writing?[confirm]
Are you sure?[confirm]
Erasing device ...ezeeze...erased.

Connected to 131.108.1.111

Loading 2076007 byte file btxx:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!![OK]

Verifying checksum... (0x87FD)...[OK]
Router1#
```

The exclamation points (!) indicate that the copy process is taking place. Each exclamation point (!) indicates that ten packets have been transferred successfully.

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 an additional bank of Flash memory.

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. If the boot ROMs do not support Flash load helper, you must perform the Flash upgrade manually. See the “Manually Boot from Flash” section in this chapter.

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 in either an erased state or with a file that cannot boot.

In run-from-Flash systems, the software image in the router is stored in and executed from the Flash EPROM (as opposed to being executed 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 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. By doing this in the event of a catastrophic failure during the upgrade, at least the boot ROM image can be brought up as a last resort rather than have the system wait at the ROM monitor's 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 the system image.
- Notifies users logged into 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 there is a failure 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 Using Dual Flash Bank" section in this chapter.

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, then perform the following tasks 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 the ROM monitor prompt 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 re-initiating the **copy tftp flash** command, you must set the boot bits 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 respond. 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]? 131.108.1.111
```

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 131.108.1.111....
Loading from 131.108.13.111:
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 TFTP 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 131.108.1.111 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 (in other words, if 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 the boot bits in the configuration register. If the boot bits are zero, 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 **b flash** command at the ROM monitor prompt.

If the boot bits are nonzero, 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.

Verify the Image in Flash Memory

Before booting from Flash memory, verify that the checksum of the image in Flash memory matches the checksum listed in the README file that was distributed with the system software image. The checksum of the image in Flash memory is displayed at the bottom of the screen when you issue the **copy tftp flash**, **copy rcp flash**, or **copy rcp bootflash** commands. The README file was copied to the network server automatically when you installed the system software image on the server.

Caution If the checksum value does not match the value in the README file, do not reboot the router. Issue the copy request and compare the checksums again. If the checksum is repeatedly wrong, copy the original system software image bootstrap image back into Flash memory *before* you reboot the router from Flash memory. If you have a bad image in Flash memory and try to boot from Flash, the router will start the system image contained in ROM (assuming netbooting is not configured). If ROM does not contain a fully functional system image, the router will not function and will have to be reconfigured through a direct console port connection.

Partition Flash Memory Using Dual Flash Bank

Dual Flash bank allows you to partition two banks of Flash memory into two separate, logical devices so the router can hold and maintain two different software images. No downtime is required to write software into Flash memory while running software that is in another bank of Flash memory.

Dual Flash bank is supported on low-end systems that have at least two banks of Flash memory, including systems that support a single SIMM that has two banks of Flash memory (the AccessPro card and the Cisco 4500). Systems that support dual Flash bank include the AccessPro PC card, Cisco 2500 series, Cisco 3000 series, and Cisco 4000 series. CiscoFlash MIB variables support dual Flash bank. Refer to the *Cisco Management Information Base (MIB) User Quick Reference* for more information on these variables.

To use dual Flash bank, you must have at least two banks of Flash memory; a bank is a set of 4 chips. The minimum partition size is the size of a bank.

There are several benefits to partitioning Flash memory:

- 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 it causes no network disruption or downtime. After the download is complete, you can switch over 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.

You might use Flash load helper rather than dual Flash bank 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 you 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 in this chapter for information about using Flash load helper.

Understand Relocatable Images

Partitioning requires that run-from-Flash images be loaded into different Flash memory banks at different physical addresses. This means that images must be relocatable. A relocatable image is an image that contains special relocation information that allows the following:

- The image to relocate itself whenever it is loaded into RAM for execution
- A download program with appropriate support to relocate the image before it is stored in Flash memory, so that the image can run in place in Flash memory, regardless of where in Flash memory it is stored

Run-from-Flash systems (that is, the Cisco 2500 series and some Cisco 3000 series) used to run nonrelocatable images that needed to be stored in Flash memory at a specific address. This means that the image had to be stored as the first file in Flash memory. If the image is stored at any other location in Flash memory, it could not be executed in Flash memory, nor could the image be executed from RAM. The relocatable image overcomes this limitation.

With Flash partitioning, the run-from-Flash images will not work unless they are loaded into the first device as the first file. This requirement defeats the purpose of partitioning. However, relocatable images can be loaded into any Flash partition (and not necessarily as the first file within the partition) and executed in place.

Note that unless the image is downloaded as the first file in the first partition, this download must be performed by an image that recognizes relocatable images.

A relocatable image is an image that is Cisco IOS Release 10.0(6) or later. A nonrelocatable image is an image that was created before the relocatable image era (and hence does not recognize relocatable images). The following are nonrelocatable images:

- Any image from a release prior to Cisco IOS Release 10.0
- Any Release 10.0 image prior to Release 10.0(6)
- Release 10.2
- Release 10.3

You can identify a relocatable image by its name. The naming convention for image names for storage on a UNIX system is as follows:

platform-capabilities-type

The letter "l" in the type field indicates a relocatable image. Examples of some relocatable image names are:

- *igs-i-l*—IP-only image
- *igs-d-l*—desktop feature image
- *igs-bpx-l*—enterprise image

Only the "igs" prefix images used by the Cisco 3000 series and Cisco 2500 series are available as relocatable images. Images distributed on floppy disks might have different naming conventions.

For backward compatibility, the relocatable images have been linked to execute as the first file in the first Flash memory bank. This makes the images similar to previous Flash memory images. Thus, if you download a relocatable image into a nonrelocatable image system, the image will run correctly from Flash memory.

Dual Flash Bank Configuration Task List

To use dual Flash bank, perform the tasks in one or more of the following sections:

- Partition Flash Memory
- Download a File into a Flash Partition
- Manually Boot from Flash
- Configure the Router to Automatically Boot from Flash Memory
- Configure a Flash Partition as a TFTP Server

See the "Boot Systems That Have Dual-Bank Flash Memory" section in this chapter for information about booting systems that have two banks of Flash memory.

See the "Display System Image and Configuration Information" section in this chapter for information about monitoring dual Flash bank.

To upgrade your software, you must erase Flash memory when you are prompted during the download. This is to ensure that the image is downloaded as the first file in Flash memory.

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*]

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 two partitions.

Download a File into a Flash Partition

To download a file into a Flash partition, perform one of the following tasks in EXEC mode:

Task	Command
Download a file from a TFTP server into a Flash partition.	**copy tftp flash**
Download a file from a MOP server into a Flash partition.	**copy mop flash**
Download a file from an rcp server into a Flash partition.	**copy rcp flash**

The prompts displayed after you execute these tasks indicate the method by which the download can be done into each partition. The possible methods are as follows:

- None—There is no known way to copy into the partition.
- RXBOOT-Manual—You must manually reload to the rxboot image in ROM in order to copy the image.
- RXBOOT-FLH—The copy will be done using the Flash load helper software in boot ROM; that is, it will be done automatically.
- Direct—The copy can done directly.

If the image download can be done into more than one partition, you are prompted for the partition number. Enter any of the following at the partition number prompt to obtain help:

- **?**—Display the directory listings of all partitions.
- **?1**—Display the directory of the first partition.
- **?2**—Display the directory of the second partition.
- **q**—Quit the copy command.

Manually Boot from Flash

To manually boot the router from Flash memory, perform one of the following tasks in ROM monitor mode:

Task	Command
Boot the first bootable file found in any partition.	**boot flash** or **boot flash flash:**
Boot the first bootable file from the specified partition.	**boot flash** *partition-number***:** or **boot flash flash:***partition-number***:**
Boot the specified file from the first partition.	**boot flash** *filename* or **boot flash flash:***filename*
Boot the specified file from the specified partition.	**boot flash** *partition-number***:***filename* or **boot flash flash:***partition-number***:***filename*

The result of booting a relocatable image from Flash memory depends on where and how the image was downloaded into Flash memory. Table 3-1 describes various means by which an image could be downloaded and the corresponding result of booting from Flash memory.

Table 3-1 Downloading an Image and Booting from Flash

Method of Downloading	Result of Booting from Flash
The image was downloaded as the first file by a nonrelocatable image.	The image will execute in place from Flash memory, just like a run-from-Flash image.
The image was downloaded not as the first file by a nonrelocatable image.	The nonrelocatable image would not have relocated the image before storage in Flash memory. This image will not be booted.
The image was downloaded as the first file by a relocatable image.	The image will execute in place from Flash memory, just like a run-from-Flash image.
The image was downloaded not as the first file by a relocatable image (including download into the second partition).	The relocatable image relocates the image before storage in Flash memory. Hence, the image will execute in place from Flash memory, just like any other run-from-Flash image.

Configure the Router to Automatically Boot from Flash Memory

To configure the router to boot automatically from Flash memory, perform one of the following tasks in global configuration mode:

Task	Command
Boot the first bootable file found in any partition.	**boot system flash** or **boot system flash:**
Boot the first bootable file from the specified partition.	**boot system flash** *partition-number***:** or **boot system flash flash:***partition-number***:**
Boot the specified file from the first partition.	**boot system flash** *filename* or **boot system flash flash:***filename*
Boot the specified file from the specified partition.	**boot system flash** *partition-number***:***filename* or **boot system flash flash:***partition-number***:***filename*

The result of booting a relocatable image from Flash memory depends on where and how the image was downloaded into Flash memory. Table 3-1 shown earlier describes various means by which an image could be downloaded and the corresponding result of booting from Flash memory.

Configure a Flash Partition as a TFTP Server

To configure a Flash partition as a TFTP server, perform one of the following tasks in global configuration mode:

Task	Command
Specify a file.	**tftp-server flash** *filename1*
Specify a file in the first partition of Flash.	**tftp-server flash** *filename1*
Specify a file in the specified partition of Flash.	**tftp-server flash** *partition-number***:***filename1*

Once you have specified TFTP server operation, exit configuration mode and save the configuration information to NVRAM.

Copy System Images from Flash Memory to a Network Server Using TFTP

You can copy a system image back to a TFTP network server. In some implementations of TFTP, you must first create a “dummy” file on the TFTP server and give it read, write, and execute permissions before copying a file over it. Refer to your TFTP documentation for more information.

The network server copy of the system image can serve as a backup copy and also can be used to verify that the copy in Flash is the same as on the original file on disk. To copy the system image to a network server, perform the following task:

Task		Command
Step 1	(Optional) If you do not already know it, learn the exact spelling of the system image filename in Flash memory. On the Cisco 7500 series, you can learn the spelling of the system image filename on a specified Flash memory device.	**show flash all** **show flash** [*device***:**] (Cisco 7500 series only)
Step 2	Copy the system image from Flash memory to a TFTP server. On the Cisco 7500 series, you can copy the system image from a specified Flash memory device to a TFTP server.	**copy flash tftp** or **copy** *file_id* **tftp** (Cisco 7500 series only)
Step 3	When prompted, enter the IP address or domain name of the TFTP server.	*ip-address* or *name*
Step 4	When prompted, enter the filename of the system image in Flash memory.	*filename*

The following example uses the **show flash all** command to learn the name of the system image file and the **copy flash tftp** command to copy the system image to a TFTP server. The name of the system image file (xk09140z) is listed near the end of the **show flash all** output.

```
Router# show flash all
2048K bytes of flash memory on embedded flash (in XX).
   ROM   socket    code    bytes       name
    0      U42     89BD   0x40000     INTEL 28F020
    1      U44     89BD   0x40000     INTEL 28F020
    2      U46     89BD   0x40000     INTEL 28F020
    3      U48     89BD   0x40000     INTEL 28F020
    4      U41     89BD   0x40000     INTEL 28F020
    5      U43     89BD   0x40000     INTEL 28F020
    6      U45     89BD   0x40000     INTEL 28F020
    7      U47     89BD   0x40000     INTEL 28F020
  security jumper(12V) is installed,
  flash memory is programmable.
file  offset      length        name
 0     0x40        1204637     xk09140z
  [903848/2097152 bytes free]

Router# copy flash tftp
IP address of remote host [255.255.255.255]? 101.2.13.110
filename to write on tftp host? xk09140z
writing xk09140z !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
successful tftp write.
Router#
```

To stop the copy process, press **Ctrl**-^. Refer to the *Troubleshooting Internetworking Systems* publication for procedures on how to resolve Flash memory problems.

The following example uses the **show flash** [*device***:**] command on a Cisco 7500 series to display the name of the system image file to copy. In the example, the Flash memory device containing the system image is the second PCMCIA slot. The file to copy is *test*. The example uses the **copy** *file_id* **tftp** command, to copy *test* to a TFTP server.

```
Gouda#show flash slot1:
-#- ED --type-- --crc--- -seek-- nlen -length- -----date/time------ name
1   .. 1        46A11866 2036C   4    746      May 16 1995 16:24:37 test
```

```
Gouda#copy slot1:test tftp
IP address of remote host [255.255.255.255]? 101.2.13.110
filename to write on tftp host? [test]y
writing test !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
successful tftp write.
Gouda#
```

Once you have configured Flash memory, you might want to configure the system (using the **configure terminal** command) with the **no boot system flash** configuration command to revert to booting from ROM (for example, if you do not yet need this functionality, if you choose to netboot from a network server, or if you do not have the proper image in Flash memory). After you enter the **no boot system flash** command, use the **write memory** command to save the new configuration command to NVRAM.

This procedure on the Cisco 7000 series also requires changing the jumper on the processor's configuration register. Refer to the appropriate hardware installation and maintenance manual for instructions.

Copy System Images from Flash Memory to a Network Server Using rcp

You can copy a system image back to a network server. This copy of the system image can serve as a backup copy and also can be used to verify that the copy in Flash memory is the same as on the original file on disk.

The rcp protocol requires that a client send the remote username on each rcp request to the server. When you copy a bootstrap image from Flash memory to a network server using rcp, the router software sends the remote username associated with the current TTY (terminal) process, if that name is valid. If the TTY remote username is invalid, the router software uses the router host name as the both the remote and local usernames.

Note For Cisco, TTYs are commonly used in communications servers. The concept of TTYs 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).

You can configure a different remote username to be sent to the server. The rcp protocol implementation writes the system image relative to the directory associated with the remote username on the network server, if the server has a directory structure, for example, as do UNIX systems.

For the rcp command to execute properly, an account must be defined on the destination 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.

To stop the copy process, press **Ctrl-**^. Refer to the *Troubleshooting Internetworking Systems* publication for procedures on how to resolve Flash memory problems.

If you copy the system image to a personal computer used as a file server, the computer must support the rcp protocol.

To copy the system image to a network server, perform the following tasks:

Task		Command
Step 1	(Optional) If you do not already know it, learn the exact spelling of the system image filename in Flash memory. On the Cisco 7500 series, you can learn the spelling of the system image filename on a specified Flash memory device.	**show flash all** **show flash** [*device*:] (Cisco 7500 series only)
Step 2	Enter configuration mode from the terminal. This step is required only if you are going to override the default remote username (see Step 3).	**configure terminal**
Step 3	(Optional) Specify the remote username. This step is optional, but recommended.	**ip rcmd remote-username** *username*
Step 4	Exit configuration mode.	**^Z**
Step 5	Using rcp, copy the system image in Flash memory to a network server.	**copy flash rcp** **copy** *file_id* **rcp** (Cisco 7500 series only)
Step 6	When prompted, enter the IP address or domain name of the rcp server.	*ip-address* or *name*
Step 7	When prompted, enter the filename of the system image in Flash memory.	*filename*

The following example copies the system image *gsxx* to a network server using rcp:

```
Router# configure terminal
Router# ip rcmd remote-username netadmin1
^Z
Router# copy flash rcp
System flash directory:
File name/status
 1 gsxx
[2076072 bytes used, 21080 bytes available]

Name of file to copy? gsxx
Address or name of remote host [UNKNOWN]? 131.108.1.111
File name to write to? gsxx
Verifying checksum for ‘gsxx’ (file # 1)...[OK]

Writing gsxx !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Router#
```

The exclamation points (!) indicate that the copy process is taking place. Each exclamation point (!) indicates that ten packets have been transferred successfully.

The following example copies a system image file called *test* from the second PCMCIA slot on a Cisco 7500 series to a network server using rcp:

```
Router# configure terminal
Router(config)# ip rcmd remote-username netadmin1
^Z
Router# copy slot1:test rcp
System flash directory:
File name/status
 1 test
[2076072 bytes used, 21080 bytes available]

Name of file to copy? [test] y
Address or name of remote host [UNKNOWN]? 131.108.1.111
File name to write to? test
Verifying checksum for ‘test’ (file # 1)...[OK]

Writing test !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Router#
```

Copy a Configuration File from a Network Server to the Router Using rcp

You can copy a configuration file from a network server to the local router using rcp. You might use this process to restore a configuration file to the router if you have backed up the file to a server. If you replace a router and want to use the configuration file that you created for the original router, you could restore that file instead of recreating it. You can also use this process to copy to the router a different configuration that is stored on a network server.

There are two ways to copy a configuration file from a network server using rcp:

- Copy the file to NVRAM. You can copy a configuration file from a network server to the router’s NVRAM.
- Copy and run the file. You can copy a configuration file from a network server to the router and run that configuration from RAM.

The rcp protocol requires that a client send the remote username on each rcp request to a network server. When you issue a request to copy a configuration file from a network server using rcp and copy it to NVRAM or copy and run it, the router sends a default remote username unless you override the default by configuring a remote username. As the default value of the remote username, the router software sends the remote username associated with the current TTY process, if that name is valid. If the TTY username is invalid, the router software uses the router host name as the both the remote and local usernames. The rcp implementation searches for the configuration file to be copied relative to the directory associated with the remote username on the network server, if the server has a directory structure, for example, as do UNIX systems.

For the rcp copy request to execute successfully, an account must be defined on the network server for the remote username. If you copy the configuration file from a personal computer used as a file server, the remote host computer must support the remote shell protocol.

Copy a Configuration File to the Startup Configuration

You can retrieve the commands stored in a configuration file on a server and write them to the startup configuration. On all platforms except the Cisco 7500 series, the startup configuration is NVRAM. On the Cisco 7500 series, the CONFIG_FILE environment variable specifies the device and filename of the startup configuration.

A host configuration file contains commands that apply to one network server in particular. A network configuration file contains commands that apply to all network servers on a network.

To copy a configuration file from an rcp network server to the startup configuration, perform the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal. This step is only required if you are going to override the default remote username (see Step 2).	**configure terminal**
Step 2	Specify the remote username. This step is optional, but recommended.	**ip rcmd remote-username** *username*
Step 3	Exit configuration mode.	**^Z**
Step 4	Using rcp, copy the configuration file from a network server to the router's NVRAM.	**copy rcp startup-config**
Step 5	When prompted, enter the IP address of the network server.	*ip-address*
Step 6	When prompted, enter the name of the configuration file.	*filename*

On the Cisco 7500 series, the **copy rcp startup-config** command copies the configuration file from the network server to the configuration file pointed to by the CONFIG_FILE environment variable. If you want to write the configuration file from the server to NVRAM on the router, be sure to set the CONFIG_FILE environment variable to NVRAM. Refer to the "Download the CONFIG_FILE Environment Variable Configuration on Cisco 7500 Series" section in this chapter for instructions on setting the CONFIG_FILE environment variable with the **boot config** command.

The following example specifies a remote username of *netadmin1*. Then it copies a host configuration file *host2-confg* from the *netadmin1* directory on the remote server with an IP address of 131.108.101.101 to the router's NVRAM:

```
Rtr2# configure terminal
Rtr2# ip rcmd remote-username netadmin1
^Z
Rtr2# copy rcp startup-config
Address of remote host [255.255.255.255]? 131.108.101.101
Name of configuration file[rtr2-confg]? host2-confg
Configure using rtr2-confg from 131.131.101.101?[confirm]
Connected to 131.131.101.101
Loading 1112 byte file rtr2-confg:![OK]
[OK]
Rtr2#
%SYS-5-CONFIG_NV:Non-volatile store configured from rtr2-config by rcp from
131.108.101.101
```

Copy a Configuration File to the Running Configuration

You can copy a configuration file from an rcp server to the running configuration.

A host configuration file contains commands that apply to one network server in particular. A network configuration file contains commands that apply to all network servers on a network.

To copy a configuration file from an rcp server to the running configuration, perform the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal. This step is only required if you are going to override the default remote username (see Step 2).	**configure terminal**
Step 2	Specify the remote username. This step is optional, but recommended.	**ip rcmd remote-username** *username*
Step 3	Exit configuration mode.	**^Z**
Step 4	Using rcp, copy the configuration file from a network server to the router.	**copy rcp running-config**
Step 5	When prompted, enter the IP address of the server.	*ip-address*
Step 6	When prompted, enter the name of the configuration file.	*filename*

The following example copies a host configuration file named *host1-confg* from the *netadmin1* directory on the remote server with an IP address of 131.108.101.101, and loads and runs that file on the router:

```
Router# configure terminal
Router# ip rcmd remote-username netadmin1
^Z
Router# copy rcp running-config
Host or network configuration file [host]?
Address of remote host [255.255.255.255]? 131.108.101.101
Name of configuration file [Router-confg]? host1-confg
Configure using host1-confg from 131.108.101.101? [confirm]
Connected to 131.108.101.101
Loading 1112 byte file host1-confg:![OK]
Router#
%SYS-5-CONFIG: Configured from host1-config by rcp from 131.108.101.101
```

Copy a Configuration File from the Router to a Network Server Using TFTP

You can copy configuration files from the router to a TFTP server. You might do this task to back up a current configuration file to a server before changing its contents, thereby allowing you to later restore the original configuration file from the server. The configuration file that you copy to usually must already exist on the TFTP server and be globally writable before the TFTP server allows you to write to it.

To store configuration information on a TFTP network server, complete the following tasks in the EXEC mode:

Task		Command
Step 1	Specify that the configuration (in NVRAM or pointed to by the CONFIG_FILE environment variable) be stored on a network server.	**write network**
Step 2	Enter the IP address of the network server.	*ip-address*
Step 3	Enter the name of the configuration file to store on the server.	*filename*
Step 4	Confirm the entry.	**y**

The command prompts you for the destination host's address and a filename, as the following example illustrates.

The following example copies a configuration file from a router to a server:

```
Tokyo# write network
Remote host [131.108.2.155]?
Name of configuration file to write [tokyo-confg]?
Write file tokyo-confg on host 131.108.2.155? [confirm] y
#
Writing tokyo-confg !! [OK]
```

Copy a Configuration File from the Router to a Network Server Using rcp

You can use rcp to copy configuration files from the local router to a network server. You can back up current configuration files to the server before you change a file's contents, and restore the original configuration files from the server at a later time.

You can copy a startup configuration file or a running configuration file to the server.

The rcp protocol requires that a client send the remote username on each rcp request to a server. When you issue a command to copy a configuration file from the router to a server using rcp, the router sends a default remote username unless you override the default by configuring a remote username. As the default value of the remote username, the router software sends the remote username associated with the current TTY (terminal) process, if that name is valid.

Note 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).

If the TTY remote username is invalid, the router software uses the router host name as the both the remote and local usernames. The rcp protocol implementation writes the configuration file to be copied relative to the directory associated with the remote username on the server, if the server has a directory structure, for example, as do UNIX systems.

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

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 you copy the configuration file to a personal computer used as a file server, the computer must support rcp.

To copy a startup configuration file or a running configuration file from the router to a server using rcp, use one of following tasks:

- Copy a Startup Configuration File to an rcp Server
- Copy a Running Configuration File to an rcp Server

Copy a Startup Configuration File to an rcp Server

You can copy the contents of the configuration file in NVRAM to a network server using rcp or TFTP. The copied file can serve as a backup configuration file.

To copy a startup configuration file to a network server using rcp, complete the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal. This step is only required if you are going to override the default remote username (see Step 2).	**configure terminal**
Step 2	Specify the remote username. This step is optional, but recommended.	**ip rcmd remote-username** *username*
Step 3	Specify that the router configuration file in NVRAM should be copied to a network server using rcp.	**copy startup-config rcp**
Step 4	Enter the IP address of the network server.	*ip-address*
Step 5	Enter the name of the configuration file to store on the server.	*filename*
Step 6	Confirm the entry.	**y**

The following example shows how to store a startup configuration file on a server using rcp to copy the file:

```
Rtr2# configure terminal
Rtr2# ip rcmd remote-username netadmin2
^Z
Rtr2# copy startup-config rcp
Remote host[]? 131.108.101.101
Name of configuration file to write [rtr2-confg]?
Write file rtr2-confg on host 131.108.101.101?[confirm]
![OK]
```

Copy a Running Configuration File to an rcp Server

You can copy the running configuration file to a server using rcp or TFTP. The copied file can serve as a backup configuration file.

To store a running configuration file on a server, complete the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal. This step is only required if you are going to override the default remote username (see Step 2).	**configure terminal**
Step 2	Specify the remote username. This step is optional, but recommended.	**ip rcmd remote-username** *username*
Step 3	Specify that the router’s running configuration file should be stored on a network server.	**copy running-config rcp**
Step 4	Enter the IP address of the network server.	*ip-address*

Task		Command
Step 5	Enter the name of the configuration file to store on the server.	*filename*
Step 6	Confirm the entry.	**y**

The following example copies the running configuration file named *Rtr2-confg* to the *netadmin1* directory on the remote host with an IP address of 131.108.101.101:

```
Rtr#2 configure terminal
Rtr2# ip rcmd remote-username netadmin1
^Z
Rtr2# copy running-config rcp
Remote host[]? 131.108.101.101
Name of configuration file to write [Rtr2-confg]?
Write file rtr2-confg on host 131.108.101.101?[confirm]
###![OK]
Connected to 131.108.101.101
Rtr2#
```

Display System Image and Configuration Information

Perform the following tasks in EXEC mode to display information about system software, system image files, and configuration files:

Task	Command
List the system software release version, configuration register setting, and so on.	**show version**
List the contents of the BOOT environment variable, the name of the configuration file pointed to by the CONFIG_FILE environment variable, and the contents of the BOOTLDR environment variable.	**show boot** (Cisco 7500 series only)
List the configuration information stored in NVRAM.	**show configuration**
List the configuration information in running memory.	**write terminal**
List the configuration information stored in a specified file.	**show file** *device*:*filename* (Cisco 7500 series only)
List information about Flash memory, including system image filenames and amounts of memory used and remaining.	**show flash**
List information about Flash memory, including system image filenames, amounts of memory used and remaining, and Flash partitions.	**show flash** [**all** | **chips** | **detailed** | **err** | **partition** *number* [**all** | **chips** | **detailed** | **err**] | **summary**] **show flash** [**all | chips | filesys**] [*device***:**] (Cisco 7500 series only)
View the console output generated during the Flash load helper operation.	**show flh-log**

Refer to the *Router Products Command Reference* for examples of these commands.

You can also use the **o** command in ROM monitor mode to list the configuration register settings on some models.

The Flash memory content listing does not include the checksum of individual files. To recompute and verify the image checksum after the image is copied into Flash memory, complete the following task in EXEC mode:

Task	Command
Recompute and verify the image checksum after the image is copied into Flash memory.	**copy verify**

When you enter this command, the screen prompts you for the filename to verify. By default, it prompts for the last (most recent) file in Flash. Press **Return** to recompute the default file checksum, or enter the filename of a different file at the prompt. Note that the checksum for microcode images is always 0x0000.

On a Cisco 7500 series, you can verify the checksum of individual files on a Flash memory device. You can verify the checksum of a file located in internal Flash (**bootflash:**) or in one of the PCMCIA slots (**slot0:**, **slot1:**). To do so, perform the following task in EXEC mode:

Task	Command
Verify the checksum of a file on a specific Flash memory device.	**verify** [*device*:]*filename*

The following example verifies the *gsxx* file on the Flash memory card inserted in slot 0 of a Cisco 7500's RSP card:

```
Router# verify slot0:gsxx
```

Clear the Configuration Information

To clear your startup configuration, perform the following task in EXEC mode:

Task	Command
Clear the contents of your startup configuration. On most platforms, this command erases the contents of NVRAM. On the Cisco 7500 series, this command erases the configuration specified by the CONFIG_FILE environment variable.	**write erase**

On the Cisco 7500 series, when you use the **write erase** command, the router erases or deletes the configuration pointed to by CONFIG_FILE environment variable. If this variable points to NVRAM, the router erases NVRAM. If the CONFIG_FILE environment variable specifies a Flash memory device and configuration filename, the router deletes the configuration file. That is, the router marks the file as "deleted," rather than erasing it. This feature allows you to recover a "deleted" file. Refer to the "Manage Flash Files on Cisco 7500 Series" section for more information on recovering deleted files.

To erase a saved configuration from a specific Flash device on a Cisco 7500 series, complete the following task in EXEC mode:

Task	Command
Erase or delete a specified configuration file on a specified Flash device.	**erase** [*device*:]*filename* or **delete** [*device*:]*filename*

When you erase or delete a specific file, the system marks the file as deleted, allowing you to later recover a “deleted” file. If you omit the device, the router uses the default device specified by the **cd** command.

If you attempt to erase or 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 erase or delete the last valid system image specified in the BOOT environment variable, the system prompts you to confirm the deletion.

Reexecute the Configuration Commands in Startup Configuration

On all platforms except the Cisco 7500 series, you can reexecute the configuration commands stored in NVRAM. On the Cisco 7500 series, the same command allows you to reexecute the configuration specified by the CONFIG_FILE environment variable.

To reexecute the commands located in the startup configuration, complete the following task in privileged EXEC mode:

Task	Command
Reexecute the configuration commands located in NVRAM. or On the Cisco 7500 series, configure the router to reexecute the configuration specified by the CONFIG_FILE environment variable.	**configure memory**

Remotely Execute Commands Using rsh

You can use rsh to execute commands remotely on network servers that support the remote shell protocol. To use this command, the *.rhosts* files on the network server must include an entry that permits you to remotely execute commands on that host.

The rsh command that you issue is remotely executed from the directory of the account for the remote user that you specify through the **/user** *username* keyword and argument pair, if the remote server has a directory structure, as do UNIX systems.

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

To execute a command remotely on a network server using rsh, perform the following tasks in privileged EXEC mode:

Task		Command
Step 1	Enter privileged EXEC mode.	**enable** [*password*][1]
Step 2	Enter the rsh command to be executed remotely.	**rsh** {*ip-address* | *host*} [**/user** *username*] *remote-command*

[1] This command is documented in the “User Interface Commands” chapter of the *Router Products Command Reference* publication.

The following example shows how to execute a command remotely using rsh:

```
Router# enable
Router1# rsh mysys.cisco.com /u sharon ls -a
.
..
.alias
.cshrc
.emacs
.exrc
.history
.login
.mailrc
.newsrc
.oldnewsrc
.rhosts
.twmrc
.xsession
jazz
Router1#
```

Use Flash Memory as a TFTP Server

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.

In the description that follows, one Cisco 7000 router is referred to as the Flash server, and all other routers are referred to as client routers. Example configurations for the Flash server and client routers include commands as necessary.

Prerequisites

The Flash server and client router must be able to reach one another before the TFTP function can be implemented. Verify this connection by pinging between the Flash server and client router (in either direction) using the **ping** command.

An example use of the **ping** command is as follows:

```
Router# ping 131.131.101.101 <Return>
```

In this example, the IP address of 131.131.101.101 belongs to the client router. Connectivity is indicated by !!!!!, while ... [timed out] or [failed] indicates no connection. If the connection fails, reconfigure the interface, check the physical connection between the Flash server and client router, and ping again.

After you verify the connection, ensure that a TFTP-bootable image is present in Flash memory. 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.

Note 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.

Caution For full functionality, the software residing in the Flash memory 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 Flash memory.

Configure the Flash Server

Perform the following task to configure the Flash server:

Task	Command
Step 1 Enter configuration mode from the terminal.	**configure terminal**
Step 2 Specify the TFTP server operation for a router.	**tftp-server flash** [*partition-number***:**]*filename1* [**alias** *filename2*] [*access-list-number*] or **tftp-server flash** *device***:***filename* (Cisco 7500 series only)

The following example configures the Flash server. This example gives the filename of the software image in the Flash server and one access list (labeled 1). The access list must include the network where the client router resides. Thus, in the example, the network 131.108.101.0 and any client routers on it are permitted access to the Flash server filename *gs7-k.9.17*.

```
Server# configure terminal
Enter configuration commands, one per line.
Edit with DELETE, CRTL/W, and CRTL/U; end with CTRL/Z
Server# tftp-server flash gs7-k.9.17 1
Server# access-list 1 permit 131.108.101.0 0.0.0.255
^Z
Server# write memory <Return>
[ok]
Server#
```

Configure the Client Router

Configure the client router using the **boot system rom** command. Use the **configure terminal** command to enter this commands into the client router's memory configuration. Using these commands on the Cisco 7000 requires changing the jumper on the configuration register of the processor to the pattern 0-0-1-0 (Position 1). For this exercise, the IP address of the Flash server is 131.131.111.111.

Task	Command
Enter configuration mode from the terminal.	**configure terminal**
Boot the router from ROM.	**boot system rom**

Caution Using the **no boot system** command 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.

Following is an example of the use of these commands:

```
Client# configure terminal
Enter configuration commands, one per line.
Edit with DELETE, CRTL/W, and CRTL/U; end with CTRL/Z
Client# no boot system
Client# boot system gs7-k.9.17 131.131.111.111
Client# boot system rom
```

```
^Z
Client# write memory <Return>
[ok]
Server# reload
```

In this example, the **no boot system** command invalidates all other **boot system** commands currently in the configuration memory, and any **boot system** commands entered after this command will be executed first. The second command, **boot system** *filename address*, tells the client router to look for the file gs7-k.9.17 in the (Flash) server with an IP address of 131.131.111.111. Failing this, the client router will boot from its system ROM upon the **boot system rom** command, which is included as a backup in case of a network problem. The **write memory** command copies the configuration to memory, and the **reload** command boots the system.

Caution The system software (gs7-k.9.17) to be booted from the Flash server (131.131.111.111) must reside in Flash memory on the server. If it is not in Flash memory, the client router will boot the Flash server's system ROM.

Use the **show version** command on the client router to verify that the software image booted from the Flash server is the image present in Flash memory.

Following is sample output of the **show version** command:

```
env-chassis> show version
GS Software (GS7), Version 9.1.17
Copyright (c) 1986-1992 by cisco Systems, Inc.
Compiled Wed 21-Oct-92 22:49

System Bootstrap, Version 4.6(0.15)

Current date and time is Thu 10-22-1992 13:15:03
Boot date and time is Thu 10-22-1992 13:06:55
env-chassis uptime is 9 minutes
System restarted by power-on
System image file is "gs7-k.9.17", booted via tftp from 131.131.111.111

RP1 (68040) processor with 16384K bytes of memory.
X.25 software.
Bridging software.
1 Switch Processor.
1 EIP controller (6 Ethernet).
6 Ethernet/IEEE 802.3 interface.
128K bytes of non-volatile configuration memory.
4096K bytes of flash memory on embedded flash (in RP1).
Configuration register is 0x0
```

The important information in this example is contained in the first line (GS Software...) and in the line that begins with "System image file...." The two software types and versions shown indicate the software currently running in RAM in the client router (first line) and the software booted from the Flash server (last line). These two types and versions must be the same.

Note If no bootable image was present in the Flash server memory when the client server was booted, the version currently running (first line of the preceding example) will be the system ROM version of the Flash server by default.

Verify that the software shown in the first line of the previous example is the software residing in the Flash server memory.

Manage Flash Files on Cisco 7500 Series

With the Cisco 7500 series, you must manage files on as many as three different Flash memory devices. To help you manage your Flash files, you can

- Set the System Default Flash Device
- Display the Current Default Flash Device
- Show a List of Files on a Flash Device
- Delete Files on a Flash Device
- Recover Deleted Files on a Flash Device
- Permanently Delete Files on a Flash 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 the slot 0 of the RSP card:

```
cd slot0:
```

Display the Current Default Flash Device

You may want to show the current setting of the **cd** command to see which device is 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 of the RSP card:

```
Gouda>pwd
slot0
Gouda>
```

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:

```
Gouda>cd bootflash:
Gouda>pwd
bootflash
Gouda>
```

Show a List of Files on a Flash Device

You may want to 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:

```
Gouda#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:

```
Gouda#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)
```

Delete Files on a Flash Device

When you no longer need a file on a Flash memory device, you can delete it.

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* or **erase** [*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. 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.

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

```
delete slot0:myconfig
```

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

```
erase 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 of the RSP card:

```
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 EXCE 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.

Load Microcode Images over the Network

Cisco 7000 interface processors and the switch processor (SP) each have a writable control store (WCS). The WCS stores microcode. You can load updated microcode onto the WCS from the onboard ROM or from Flash memory on the route processor (RP) card. On the Cisco 7500 series, you can load updated microcode onto the WCS from bootflash or a Flash memory card inserted in one of the PCMCIA slots of the RSP card.

With this feature, you can update microcode without having physical access to the router, and you can load new microcode without rebooting the system.

The default is to load from the microcode bundled in the system image.

To load microcode from Flash, complete the following task:

Task		Command
Step 1	Copy microcode files into Flash.	**copy tftp flash** or **copy tftp** *file_id* (Cisco 7500 series only) See the section “Copy System Images from a Network Server to Flash Memory Using TFTP” earlier in this chapter for more information about how to copy TFTP images to Flash memory.
Step 2	Load microcode from Flash memory into the WCS.	**microcode** *interface* [**flash** *filename* | **rom** | **system**] (Cisco 7000 series only) **microcode** *interface* [**flash** *file_id* | **system**] (Cisco 7500 series only)
Step 3	Save the configuration changes to your startup configuration. On most platforms, this step saves the configuration to NVRAM. On the Cisco 7500 series, this step saves the configuration to the location specified by the CONFIG_FILE environment variable.	**write memory**

If an error occurs when you are attempting to download microcode, the onboard ROM microcode will be loaded and the interface will remain operational.

Note Microcode images cannot be compressed.

These configuration commands are implemented following one of three events:

- The system is booted.
- A card is inserted or removed.
- The configuration command **microcode reload** is issued.

After you have entered a microcode configuration command and one of these events has taken place, all of the cards are reset, loaded with microcode from the appropriate sources, tested, and enabled for operation.

To signal to the system that all microcode configuration commands have been entered and the processor cards should be reloaded, complete the following task in global configuration mode:

Task	Command
Notify the system that all microcode configuration commands have been entered and the processor cards should be reloaded.	**microcode reload**

If Flash memory is busy because a card is being removed or inserted, or a **microcode reload** command is executed while Flash is locked, the files will not be available and the onboard ROM microcode will be loaded. Issue another **microcode reload** command when Flash memory is available, and the proper microcode will be loaded. The **show flash** command will show if another user or process has locked Flash memory. The **microcode reload** command should not be used while Flash is in use. For example, do not use this command when a **copy tftp flash** or **show flash** command is active.

The **microcode reload** command is automatically added to your running configuration when you issue a microcode command that changes the system's default behavior of loading all processors from ROM.

Display Microcode Information

To display microcode information, perform the following task in EXEC mode:

Task	Command
Display microcode information.	**show microcode**