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 of 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.

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

The following list contains tasks you can do to load system images, microcode images, and configuration files.

- Use the AutoInstall Procedure
- Enter Configuration Mode
- Modify the Configuration Register Boot Field
- Specify the System Image the Router Loads upon Restart
- Specify the Configuration File the Router Loads upon Restart
- Change the Buffer Size for Loading Configuration Files
- Manually Load a System Image
- Configure a Router as a TFTP Server
- Configure a Router as a RARP Server
- Specify MOP Server Boot Requests
- Copy System Images from a Network Server to Flash Memory

- Verify the Image in Flash Memory
- Copy System Images from Flash Memory to a Network Server
- Copy a Configuration File from the Router to a Network Server
- Display System Image and Configuration Information
- Clear the Contents of NVRAM
- Reexecute the Configuration Commands in NVRAM
- Use Flash Memory as a TFTP Server
- Loading Microcode Images over the Network
- Display Microcode Information

Use the AutoInstall Procedure

This section provides information about AutoInstall, a procedure that enables 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 having it immediately enabled with a configuration file that is automatically downloaded from a TFTP server.

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

Requirements

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

- The existing preconfigured router must be running Software Release 8.3 or later.
- The new router must be running Software Release 9.1 or later.
- Both routers must be physically attached to the network by means of one or more of the following interface types: Ethernet, Token Ring, FDDI, or serial with HDLC encapsulation (the default encapsulation).
- Procedures 1 and either 2 *or* 3 must be completed:
    - 1) A configuration file for the new router must reside on a Trivial File Transfer Protocol (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.
    - 2) A file named network-confg also must reside on the server. The file must have an IP host name entry for the new router. The server must be reachable from the existing router.

      or

    - 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 autoinstall the new router via an 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. Subnet masks of any size are supported.

- If the existing router is to help autoinstall 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 MAC address to its IP address.
- IP helper addresses may need to be configured in order to forward the TFTP and DNS broadcast requests from the new router to the host that is providing those services.

How It Works

Once the requirements described in the preceding section are met, the dynamic configuration of the new router occurs in the following order:

- 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/RARP requests.
- The new router resolves its IP address-to-host name mapping.
- The new router automatically requests and downloads its configuration file from a TFTP server.

Acquire the New Router's IP Address

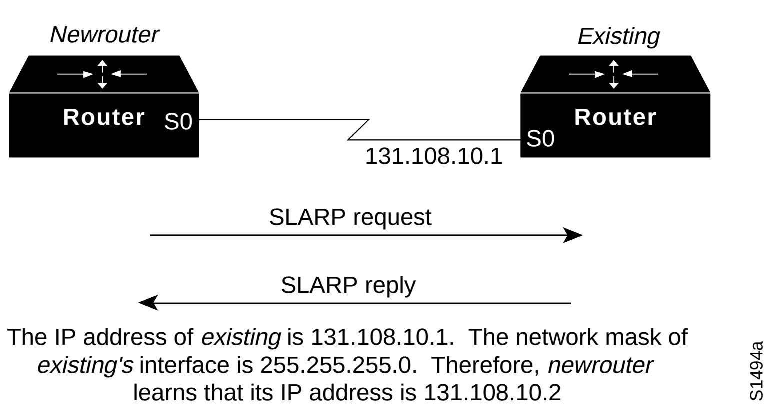
The new router (*newrouter*) resolves its interface's IP addresses by one of the following means:

- 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 to an Ethernet, Token Ring, or FDDI interface, it broadcasts BOOTP and RARP requests.

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 1-1.)

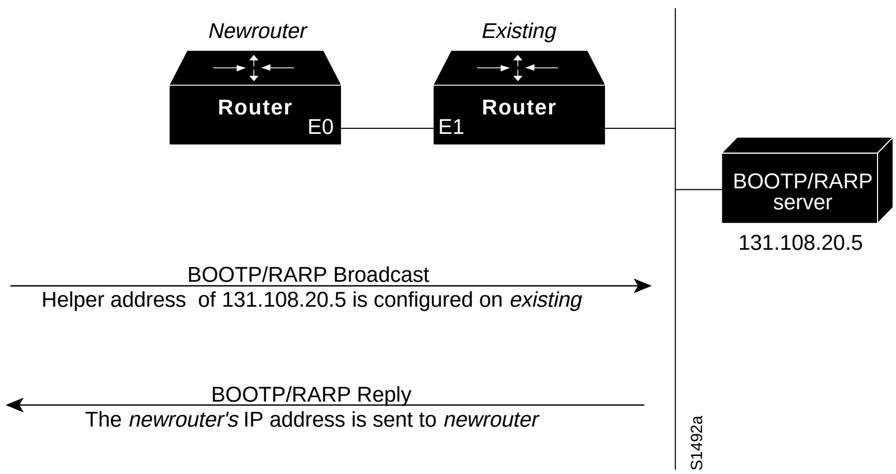
Figure 1-1 Using SLARP to Acquire the New Router's IP Address



- In response to BOOTP/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 the *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 1-2.

Figure 1-2 Using BOOTP/RARP to Acquire the New Router's IP Address



As of the current software release, routers can be configured to act as RARP servers.

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

Resolve 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 1-3.

The network-confg file is a configuration file generally shared by several routers. In this case, it is used to map the IP address 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.

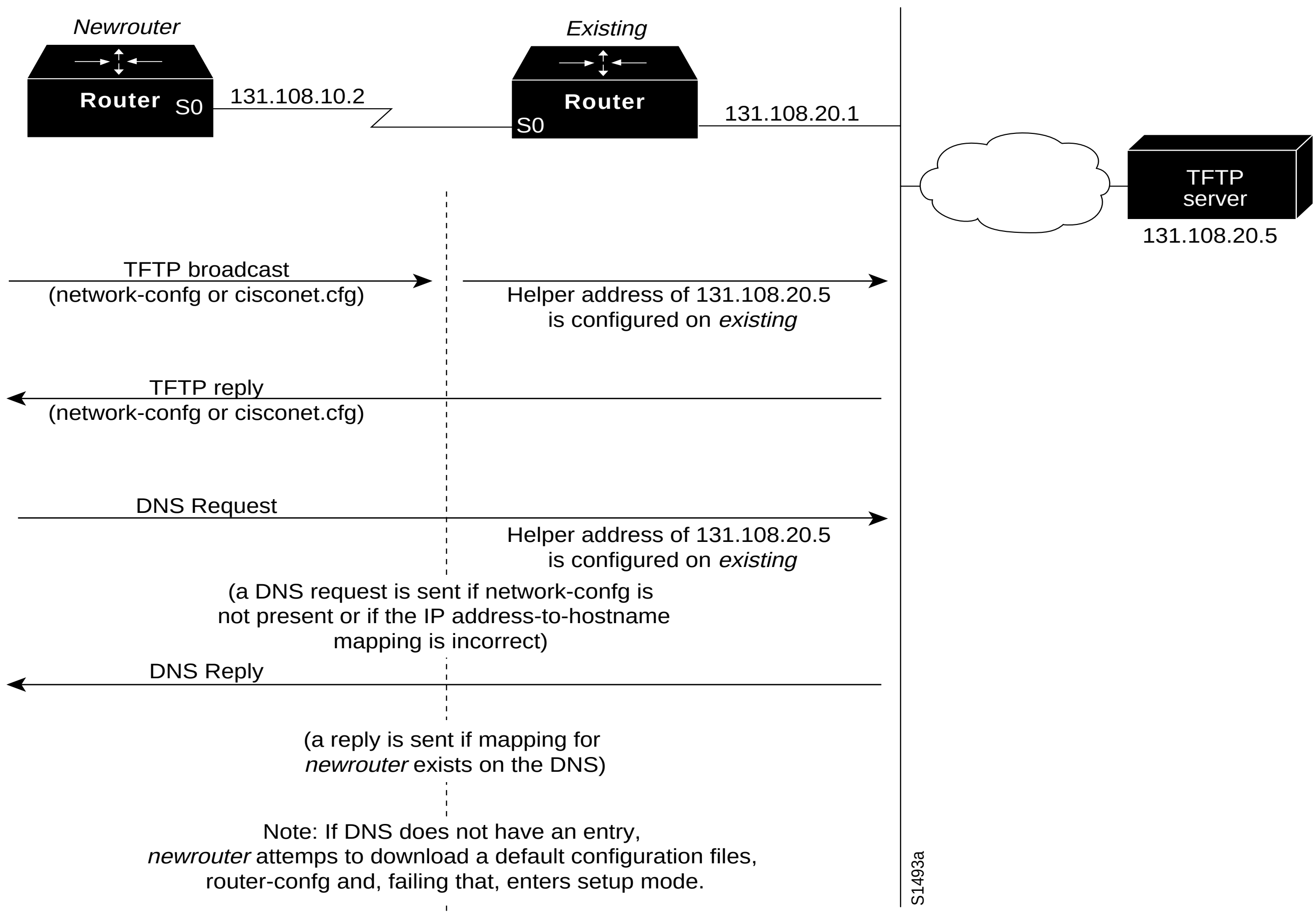
The following is an example of a minimal network-confg file that maps the IP address of the new router (131.108.10.2) to the name *newrouter*. The address of the new router was learned via SLARP and is based on *existing*'s IP address of 131.108.10.1.

```
ip host newrouter 131.108.10.2
```

If *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 Domain Name Service (DNS) broadcast. If DNS is configured and has an entry that maps *newrouter*'s SLARP or BOOTP/RARP-acquired IP address to its name, *newrouter* successfully resolves its name.

If DNS does not have an entry mapping *newrouter*'s SLARP or BOOTP/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.

Figure 1-3 Dynamically Resolving the New Router's IP Address-to-Host Name Mapping



Download 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.

If the host configuration file contains only the minimal information, the administrator 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. The administrator 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 NVRAM. 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.

The following sections describe the steps to perform the AutoInstall procedure.

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.

1. Modify the existing router's configuration to support the AutoInstall procedure.
2. Set up the TFTP server to support the AutoInstall procedure.
3. Set up BOOTP or RARP server if needed (required for AutoInstall using an Ethernet, Token Ring, or FDDI interface; not required for AutoInstall using an HDLC-encapsulated serial interface).
4. Connect the new router to the network.

Modify the Existing Router's Configuration

The interface used to set up AutoInstall can be either of the following types:

- An HDLC-encapsulated serial line

  HDLC encapsulation is the default configuration for a serial line.

- An Ethernet, Token Ring, or FDDI interface

Use a Serial Interface (HDLC Encapsulation) Connection

To set up AutoInstall via a serial line with HDLC encapsulation (the default), complete the following tasks to configure the existing router:

Task		Command
Step 1	Enter configuration mode.	**configure terminal**
Step 2	Configure the serial interface that connects to newrouter with HDLC encapsulation (the default).	**interface serial** *interface-number*
Step 3	Enter an IP address for the interface The host portion of the address must have a value of 1 or 2.	**ip address** *address mask*
Step 4	Configure a helper address for the serial interface to forward broadcasts associated with the TFTP, BOOTP, and DNS requests.	**ip helper-address** *address*

Task		Command
Step 5	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.	**clockrate** *bits per second*
Step 6	Exit configuration mode.	Ctrl-Z
Step 7	Save the configuration changes to NVRAM.	**write memory**

A DTE interface must be used on the new router, because there is no default clock rate for a DCE interface.

Example

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

```
Router1# configure terminal
Router1(config)# interface serial 0
Router1(config)# ip address 131.108.10.1 255.255.255.0
Router1(config)# ip helper-address 131.108.20.5
[Ctrl-Z]
Router1# write memory
```

Use an Ethernet, Token Ring, or FDDI Interface Connection

To set up AutoInstall using an Ethernet, Token Ring, or FDDI interface, complete the following tasks as needed to modify the configuration of the existing router. 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.

Task		Command
Step 1	Enter configuration mode.	**configure terminal**
Step 2	Configure a LAN interface.	**interface ethernet** | **tokenring** | **fddi** *interface-number*
Step 3	Enter an IP address for the interface.	**ip address** *address mask*
Step 4	Optionally, configure a helper address to forward broadcasts associated with the TFTP, BOOTP, and DNS requests.	**ip helper-address** *address*
Step 5	Exit configuration mode.	Ctrl-Z
Step 6	Save the configuration changes to NVRAM.	**write memory**

Example

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

```
Router1# configure terminal
Router1(config)# interface Ethernet 0
Router1(config-if)# ip address 131.108.10.1 255.255.255.0
Router1(config-if)# ip helper-address 131.108.20.5
[Ctrl-Z]
Router1# 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 file from a TFTP server. The new router can resolve its host name by using a network-confg file downloaded from a TFTP server or by using the Domain Name Service (DNS).

To set up a TFTP server to support AutoInstall, complete the following tasks. Steps 2 and 3 are different ways to resolve the new router's host name. Perform step 2 if you want to use a network-config file to resolve the new router's host name. Perform step 3 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 file to resolve the new router's name, create the file network-confg 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*
Step 3	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 4	Create the file <name>-confg, which should reside in the tftpboot directory on the tftp server. The <name> part of <name>-confg must match the host name you assigned for the new router in the previous step. Enter into this file configuration commands for the new router.	See the appropriate chapter in this guide for specific commands.

The <name>-confg 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 may 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 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. Telnet into 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 file could be similar to the following:

```
ip host newrouter 131.108.170.1
```

Example

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.

Example

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 *existing*'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, the steps 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" later in this chapter.

Example

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 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. 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 steps:

Task		Command
Step 1	Enter privileged mode at the system prompt on the new router.	**enable** *password*
Step 2	Save the information from the <name>-config file into NVRAM.	**write memory**

Caution Verify that the existing and new routers are connected before entering the **write memory** EXEC command to save these 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 Telnet to the new router and configure it:

Task		Command
Step 1	Establish a Telnet connection to the existing router.	**telnet** *existing*
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** (Refer to the *Router Products Getting Started Guide*)

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, nonvolatile memory (NVRAM), or a file stored on a network server 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 out of the configuration file when it is loaded to the router. However, you can list the comments in configuration files stored on a TFTP or 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.	Ctrl-Z
Step 4	Save the configuration file modifications to NVRAM.	**write memory**

Example

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 Ctrl-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
[Ctrl-Z]
Router2# write memory
```

Nonvolatile memory 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 nonvolatile memory 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 Guid*e for details on setup information.

Configure the Router from Nonvolatile Memory

You can configure the router from NVRAM by reexecuting the configuration commands stored in NVRAM. To do so, enter the following EXEC command:

configure memory

Configure the Router from a File on a Remote Host

You can configure the router by retrieving and adding to 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**
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**

Example

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]
```

Modify the Configuration Register Boot Field

The order in which the router looks for configuration information depends upon the boot field setting in the configuration register: The configuration register is a 16-bit register. The lowest four bits of the configuration register (bits 3, 2, 1, and 0) form the boot field. To change the boot field and leave all other bits set to their default values, follow these guidelines:

- Set the configuration register value to 0x100 to boot the operating system manually using the **b** command at the ROM monitor prompt. (This value sets the boot field to binary 0000.)

- Set the configuration register to 0x101 to configure the system to automatically boot from ROM. (This value sets the boot field to binary 0001.)
- Set the configuration register to any value from 0x102 to 0x10F to configure the system to use the **boot system** commands in NVRAM. (These values set the boot field to binary 0010-111.) If there are no **boot system** commands in NVRAM, the system uses the configuration register value to form a filename from which to netboot a default system image stored on a network server. (See the appropriate hardware guide for details on default filenames.)

For the Cisco 2000, Cisco 3000, Cisco 4000, or any IGS model running Software Release 9.1 or later, you can change the configuration register by completing the following tasks:

Task		Command
Step 1	Enter configuration mode, selecting the terminal option.	**configure terminal**
Step 2	Modify the default configuration register setting.	**config-register** *<value>*
Step 3	Exit configuration mode.	Ctrl-Z

For routers other than the Cisco 2000, Cisco 3000, Cisco 4000, or IGS models running Software Release 9.1 or later, the configuration register can only be changed on the processor card or with DIP switches located at the back of the router. See the appropriate hardware installation guide for details.

Use the **show version** EXEC command to list the current configuration register setting and the new configuration register setting, if any, that will be used the next time the router is reloaded. In ROM monitor mode, use the **o** command to list the value of the boot field in the configuration register.

Example

In the following example, the configuration register is set so that the router will boot automatically from the Flash memory default file. The last line of the output of the **show version** command indicates that a new configuration register setting (0x10F) will be used the next time the router is reloaded.

```
Router1# configure terminal
Router1(config)# config-register 0x10F
[Ctrl-Z]
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 (will be 0x10F at next reload)

Router1# reload
```

Specify the System Image the Router Loads upon Restart

You can enter multiple boot commands in NVRAM configuration 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 allows you to copy new system images without changing EPROMs. Information stored in Flash is not vulnerable to network failures that may 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 TFTP or MOP server provides a backup boot method for the router.

- 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.

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

Loading from Flash Memory

Flash memory is available for the AGS+, AGS, MGS, CGS, Cisco 2000, Cisco 3000, Cisco 4000, Cisco 7000, and IGS/TR platforms. Depending on the hardware platform, Flash memory might be available as EPROMs, SIMMs or memory cards. Check the appropriate hardware installation and maintenance guide for information about types of Flash memory available on a specific platform.

Flash memory is located on the Route Processor (RP) in the Cisco 7000. Software images can be stored, booted, and rewritten into 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.

Flash memory allows you to:

- Copy the TFTP image to Flash memory
- Boot a router from Flash memory either automatically or manually
- Copy the Flash memory image to a TFTP server

Note Use of Flash memory is subject to the terms and conditions of the software license agreement that accompanies your product.

Flash memory features include the following:

- It can be remotely loaded with multiple system software images through TFTP transfers (one transfer for each file loaded).
- On the Cisco 7000, it provides 4 MB Flash memory storage.
- It allows a router to be booted manually or automatically from a system software image stored in Flash memory. Booting directly from ROM or netbooting from a TFTP file server are still available options.
- It provides write protection against accidental erasing or reprogramming.

Note Booting from ROM is faster than booting from Flash. However, if you are netbooting, Flash is faster and more reliable than booting over your network.

Security Precautions

- Flash memory provides write protection against accidental erasing or reprogramming. The write-protect jumper, located next to the Flash components on the RP, can be removed to prevent reprogramming of the Flash memory, but must be installed when programming is required.
- The system image stored in Flash memory can be changed only from a privileged EXEC command session on the console terminal. This feature offers systemwide security.

Flash Memory Configuration

The following list is an overview of how to configure your 7000 system to boot from Flash memory. It is not a step-by-step set of instructions; rather, it is an overview of the process of using the Flash capability. Refer to the *Cisco 7000 Hardware Installation and Maintenance* publication for complete instructions for installing the hardware and netbooting, and in particular, for the jumper settings required for your configuration.

Step 1 Set your system to boot from ROM software.

Step 2 Restore the system configuration, if necessary.

Step 3 Copy the TFTP image to Flash memory.

Step 4 Configure from the terminal to automatically boot from the desired file in Flash memory.

Step 5 Set your system to boot from a file in Flash memory (requires jumper setting change).

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

Once you have successfully configured Flash memory, you might want to configure the system with the **no boot system flash** command to revert back to booting from ROM.

Configure the router to automatically boot from an image in Flash memory by completing 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*]
Step 3	Set the configuration register to enable loading of the system image from Flash memory.	**config-register** *value* (or use the hardware configuration register)
Step 4	Exit configuration mode.	Ctrl-Z
Step 5	Save the configuration information to NVRAM.	**write memory**

Automatically booting from Flash memory requires changing the processor's configuration register. See the section entitled "Modify the Configuration Register Boot Field" earlier in this chapter. Use the **show version** command to list the current configuration register setting.

The **boot system flash** command boots the first valid file in Flash memory. The **boot system flash** *filename* command boots the system image file specified by *filename*. If you enter more than one **boot system flash** *filename* command, the router tries them in the order entered.

If only one file is present in Flash memory, the *filename* argument is not necessary. The command **boot system flash** will boot that file.

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.

To actually boot the system, perform the following task in EXEC mode:

Task	Command
Boot the system.	**reload**

Example

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) 1986-1994 by cisco Systems
RP1 processor with 16384 Kbytes of memory
F3: 1871404+45476+167028 at 0x1000

Booting gsnew-image from flash memory
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
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.
              1525 O'Brien Drive
              Menlo Park, California 94025

GS Software (GS7), Version 9.17(0.3), BETA SOFTWARE
Copyright (c) 1986-1992 by cisco Systems, Inc.
Compiled Thu 05-Nov-92 14:16 by mlw
```

Loading from a Network Server

Complete the following tasks to specify the loading of a system image from a network server. This process is called *netbooting*.

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Specify the system image file to be netbooted from a TFTP or MOP server.	**boot system** [**tftp**] *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-registe**r *value* (or use the hardware configuration register)
Step 4	Exit configuration mode.	Ctrl-Z
Step 5	Write the configuration information to NVRAM.	**write memory**

The configuration register boot field must be set to the correct value. See "Modify the Configuration Register Boot Field" earlier in this chapter. Use the **show version** command to list the current configuration register setting.

You can also netboot from a compressed image. One reason to use a compressed image is to ensure that there is enough memory available to boot the router. On routers that do not contain a run from ROM image in EPROM, when the router netboots software, 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 netbooted.

If there is not enough room in memory to netboot a regular image, you can produce a compressed software image on any UNIX platform using the compress program. Refer to your UNIX platform's documentation for the exact usage of the compress program.

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

Example

In the following example, the router is configured to netboot from the testme5.tester system image file at IP address 131.108.13.111:

```
Router1# configure terminal
Router1(config)# boot system testme5.tester 131.108.13.111
[Ctrl-Z]
Router1# write memory
```

Loading 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** *hex-value* (or use the hardware configuration register)
Step 4	Exit configuration mode.	Ctrl-Z
Step 5	Save the configuration information to NVRAM.	**write memory**

Example

In the following example, the router is configured to boot a Flash image called image1 first. Should that image fail, the router will boot the configuration file backup1 from a network server. If that method should fail, then the system will boot from ROM.

```
Router1# configure terminal
Router1(config)# boot system flash image1
Router1(config)# boot system backup1 131.108.20.4
Router1(config)# boot system rom
[Ctrl-Z]
Router1# write memory
```

Using a Fault-Tolerant Boot Strategy

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

1. Boot an image from Flash
2. Boot an image from a system filename (netboot)
3. Boot from ROM image

This boot order provides the most fault-tolerant alternative in the netbooting environment. Use the following commands in your configuration to allow you to boot first from Flash, then from a system file, 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*]
Step 3	Configure the router to boot from a system filename.	**boot system** *filename* [*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* (or set the hardware configuration register)
Step 6	Exit configuration mode.	Ctrl-Z
Step 7	Save the configuration information to NVRAM.	**write memory**

Example

The order of the commands needed to implement this strategy is shown in the following example:

```
Router# configure terminal
boot system flash gsxx
boot system gsxx 131.131.101.101
boot system rom
^Z
Router# write memory
[ok]
Router#
```

Using this strategy, a router used primarily in a netbooting environment would have three alternative sources from which to boot. These alternative sources would help cushion the negative effects of a failure with the TFTP file server and of the network in general.

Specify the Configuration File the Router Loads upon Restart

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:

- The *network configuration* file
- The *host configuration* file

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

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

Network Configuration File

To configure the router to download a network configuration file from a server upon restart, complete the following tasks. Step 2 is optional. If you do not specify a network configuration filename, the router uses the default filename network-confg.

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.

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Optionally, enter the network configuration filename.	**boot network mop** *filename* [*MAC address*] [*interface*] **boot network** [**tftp**] *filename* [*address*]
Step 3	Enable the router to automatically load the network file upon restart.	**service config**
Step 4	Exit configuration mode.	Ctrl-Z
Step 5	Save the configuration information to NVRAM.	**write memory**

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.

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.

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Optionally, enter the host configuration filename.	**boot host mop** *filename* [*MAC address*] [*interface*] **boot host** [**tftp**] *filename* [*address*]
Step 3	Enable the router to automatically load the host file upon restart.	**service config**

Task		Command
Step 4	Exit configuration mode.	Ctrl-Z
Step 5	Save the configuration information to NVRAM.	**write memory**
Step 6	Reset the router with the new configuration information.	**reload**

Example

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
[Ctrl-Z]
Router1# write memory
```

If the network server fails to load a configuration file during startup, it tries again every ten minutes (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 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.

Change the Buffer Size for Loading Configuration Files

The buffer that holds the configuration commands is generally the size of nonvolatile memory. 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.	Ctrl-Z
Step 4	Save the configuration information to NVRAM.	**write memory**

Example

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

```
Router1# configure terminal
Router1(config)# boot buffersize 50000
[Ctrl-Z]
Router1# write memory
```

Manually Load a System Image

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 may enter read-only memory (ROM) monitor mode. From this mode, you can manually load a system image from Flash, from a network server file, or from ROM.

You can also enter ROM monitor mode by restarting the router and then pressing the Break key during the first 60 seconds of startup.

Manually Booting 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
Step 3	Manually boot the router.	**b flash** [*filename*]

Examples

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

```
 > b 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. That is the file that will be loaded.

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

Booting gs7-k from flash memory RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
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 Netbooting

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
Step 3	Manually boot the router.	**b** *filename* [*address*]

Example

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

```
>b network1
```

Manually Booting 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
Step 3	Manually boot the router from ROM.	**b**

Example

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

```
>b
```

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 to the requesting host. The TFTP Read Request message must use one of the filenames that are specified in the router's configuration.

The following algorithm is used when deciding whether to send the ROM or Flash image:

- If the specified *filename* is not stored in Flash memory, the ROM image is sent.
- If the specified *filename* exists in Flash memory, a copy of the Flash image is sent.

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 system** *filename* [*IP-access-list*]
Step 3	Exit configuration mode.	Ctrl-Z
Step 4	Save the configuration information to NVRAM.	**write memory**

The TFTP session can sometimes fail. To help determine why a TFTP session failed, TFTP generates an "E" character if it receives an erroneous packet, and an "O" character if it receives an out-of-sequence packet. A period (.) indicates a timeout. The transfer session may 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.

Example

In the following example, the router is configured to send, via TFTP, a copy of the ROM software when it receives a TFTP read request for the file version 9.0. The requesting host is checked against access list 22.

```
tftp-server system version-9.0 22
```

Configure a Router as a RARP Server

You can configure the router as a Reverse Address Resolution Protocol (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** *address*

In the following example, the router is configured to act as a RARP server. Figure 1-4 illustrates the network configuration.

Figure 1-4 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 machines’s IP addresses must use the same major network number due to a limitation of the current SunOS rpc.BOOTParamd daemon.

Specify SLIP Extended BOOTP Requests

The Boot Protocol (BOOTP) server for SLIP 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 SLIP, perform the following task in global configuration mode:

Task	Command
Configure extended BOOTP requests for SLIP.	**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** **mop retransmit-timer** **mop retries**
Step 3	Exit configuration mode.	Ctrl-Z
Step 4	Save the configuration information to NVRAM.	**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.

Example

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:

```
mop retransmit-timer 10
```

Copy System Images from a Network Server to Flash Memory

You can copy a system image from a TFTP server to Flash memory by completing 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" later in this chapter.
Step 2	Copy a system image to Flash memory.	**copy tftp flash**
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. 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. A failure message, `buffer overflow - xxxx/xxxx`, will appear, where `xxxx/xxxx` is the number of bytes read in/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 above. For the IGS/TR, Cisco 2000, Cisco 3000, Cisco 4000, and Cisco 7000, the server system image must be at least Software Version 9.1 or above.

Once you give the **copy tftp flash** command, the system prompts you for the IP address (or domain name) of the TFTP server. This can be another router serving ROM or Flash system software images. You are then prompted for the filename of the software image and 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 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. This 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 and is, therefore, not valid. In this case, the original file in Flash memory is valid and available to the system.

Example

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 lower- 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...
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
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.

Example

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
```

Example

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 abort this copy process, press Ctrl-^ (the Ctrl, Shift, and 6 keys on a standard keyboard) simultaneously. Although the process will abort, the partial file copied before the abort was issued 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.

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

Example

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...
VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV...
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.

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** command. The README file was copied to the TFTP server automatically when you installed the system software image on the TFTP server.

Caution If the checksum value is not correct according to the value in the README file, do not reboot the router. Issue the **copy tftp flash** command and compare the checksums again. If the checksum is repeatedly wrong, copy the original system software 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.

Copy System Images from Flash Memory to a Network Server

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 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	Learn the exact spelling of the system image in Flash memory.	**show flash all**

Task		Command
Step 2	Copy the system image in Flash memory to a TFTP server.	**copy flash tftp**
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*

Example

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.

Once you have configured Flash memory, you may 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, 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 also requires changing the jumper on the processor's configuration register. Refer to the appropriate hardware installation and maintenance manual for instructions.

Copy a Configuration File from the Router to a Network Server

You can copy a configuration file from the router to a network 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 network server, complete the following tasks:

Task		Command
Step 1	Specify that the router configuration file in NVRAM should 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.

Example

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]
```

Display System Image and Configuration Information

Use the following EXEC commands 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 configuration information stored in NVRAM.	**show configuration**
List the configuration information in running memory.	**write terminal**
List information about Flash memory, including system image filenames and amounts of memory used and remaining.	**show flash**
List information about Flash memory, including all the information displayed by the **show flash** command, plus information about vendor, location, individual ROM devices in Flash memory, and invalidated system image files.	**show flash all**

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

The Flash 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.

Clear the Contents of NVRAM

To clear the contents of nonvolatile memory, perform the following task in EXEC mode:

Task	Command
Clear the contents of NVRAM.	**write erase**

Reexecute the Configuration Commands in NVRAM

To reexecute the configuration commands in nonvolatile memory, perform the following task in EXEC mode:

Task	Command
Reexecute the configuration commands in NVRAM.	**configure memory**

Use Flash Memory as a TFTP Server

Flash memory can be used as a Trivial File Transfer Protocol (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 Internet Protocol (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.

Configuring the Flash Server

Use the following privileged EXEC command to configure the Flash server by adding both the **tftp-server system** command and the **access-list** command to the configuration memory:

configure terminal

Example

The following example shows the use of **configure terminal** command to get into configuration mode and configure the Flash server.

```
Server# configure terminal
Enter configuration commands, one per line.
Edit with DELETE, CRTL/W, and CRTL/U; end with CTRL/Z
tftp-server system gs7-k.9.17 1
access-list 1 permit 131.131.101.0 0.0.0.255
^Z
Server# write memory <Return>
[ok]
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 131.131.101.0 and any client routers on it are permitted access to the Flash server filename gs7-k.9.17.

Configuring the Client Router

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.

Configure the client router using the **no boot system** command, the **boot system** command, and the **boot system rom** command. Use the **configure terminal** command to enter these 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.

Example

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
no boot system
boot system gs7-k.9.17 131.131.111.111
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.

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

By default, microcode is loaded from the ROM on each interface card. (This onboard ROM microcode is not the same as the eight ROMs on the RP that contain the system image.)

To load microcode from Flash, complete the following task:

Task		Command
Step 1	Copy microcode files into Flash.	**copy tftp flash** See the section "Copy System Images from a Network Server to Flash Memory" 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-type* [**flash** | **rom**] [*filename*]
Step 3	Retain new configuration information when the system is rebooted	**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 interface 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**