

# 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 communication server. The instructions in this chapter describe how to copy system images from communication servers to network servers (and vice versa), display and compare different configuration files, and list the system software version running on the communication server.

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

For a complete description of the commands mentioned in this chapter, refer to the of the *Communication Server Command Reference* publication.

---

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

---

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

The following list contains tasks you might 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 Communication Server Loads upon Restart
- Specify the Configuration File the Communication Server Loads upon Restart
- Change the Buffer Size for Loading Configuration Files
- Manually Load a System Image
- Configure a Communication Server as a TFTP Server
- Configure a Communication Server as a RARP Server
- Specify SLIP Extended BootP Requests
- 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 a Network Server to Flash Memory
- Copy a Configuration File from the Communication Server 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

## Use the AutoInstall Procedure

This section provides information about AutoInstall, a procedure that enables you to configure a new communication server automatically and dynamically. The AutoInstall procedure involves connecting a new communication server to a network on which there is an existing preconfigured communication server, turning on the new communication server, 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, the following requirements must be met:

- The existing preconfigured communication server must be running Software Release 8.3 or later.
- The new communication server must be running Software Release 9.1 or later.
- Both communication servers must be physically attached to the network by means of one or more of the following interface types: Ethernet, Token Ring, 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 communication server must reside on a Trivial File Transfer Protocol (TFTP) server. This file can contain the new communication server’s full configuration or the minimum needed for the administrator to Telnet into the new communication server for configuration.
  - 2) A file named network-config also must reside on the server. The file must have an IP host name entry for the new communication server. The server must be reachable from the existing communication server.or
  - 3) An IP address-to-host name mapping for the new communication server must be added to a Domain Name System (DNS) database file.
- If the existing communication server is to help automatically install the new communication server 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 communication server is to help automatically install the new communication server via an Ethernet or Token Ring interface using BootP or Reverse Address Resolution Protocol (RARP), a BootP or RARP server also must be set up to map the new communication server'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 communication server to the host that is providing those services.

## How AutoInstall Works

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

- The new communication server acquires its IP address.

Depending upon the interface connection between the two communication servers, the new communication server's IP address is dynamically resolved by either SLARP requests or BootP/RARP requests.

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

## Acquire the New Communication Server's IP Address

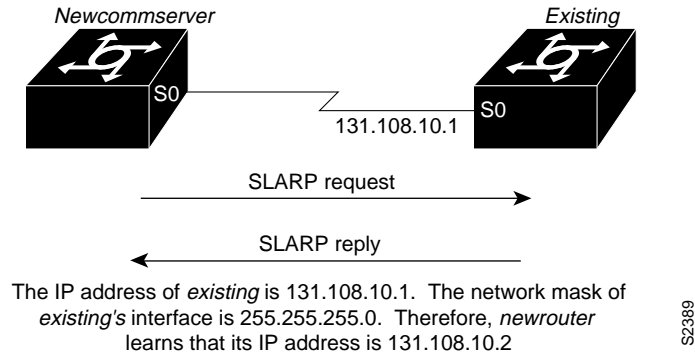
The new communication server (*newcommserver*) resolves its interface's IP addresses by one of the following means:

- If *newcommserver* is connected by an HDLC-encapsulated serial line to the existing communication server (*existing*), *newcommserver* sends a SLARP request to *existing*.
- If *newcommserver* is connected to an Ethernet or Token Ring interface, it broadcasts BootP and RARP requests.

The existing communication server (*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 *newcommserver*. 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, *newcommserver* will configure its interface using the value 2 as the host portion of its IP address, and vice versa. (See Figure 1-1.)

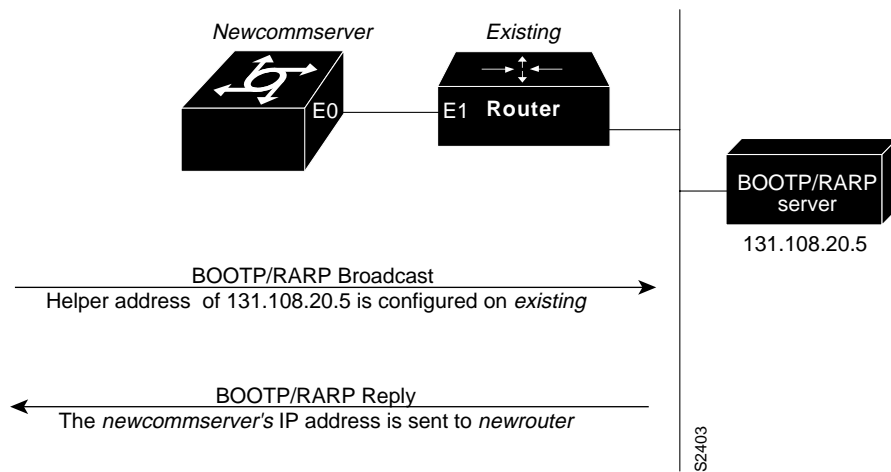
Figure 1-1 Using SLARP to Acquire the New Communication Server's IP Address



- In response to BootP/RARP requests, an IP address is sent from the BootP or RARP server to *newcommserver*.

A BootP or RARP server must have already been set up to map the *newcommserver*'s MAC address to its IP address. If the BootP server does not reside on the directly attached network segment, communication servers between *newcommserver* 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 Communication Server's IP Address



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

As soon as one interface resolves its IP address, the communication server 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 communication server resolves its IP address-to-host name mapping by sending a TFTP broadcast requesting the file `network-config`, as shown in Figure 1-3.

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

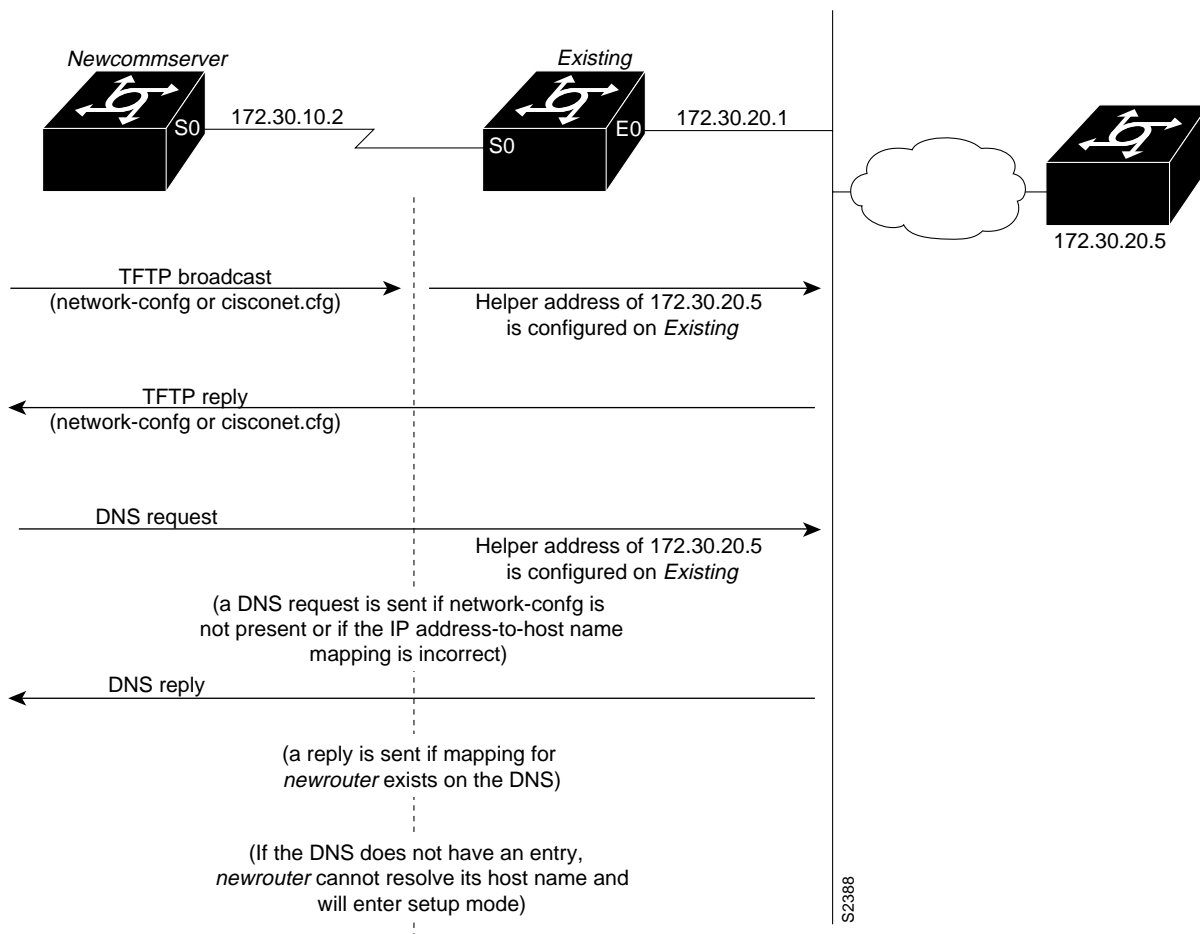
The following is an example of a minimal `network-config` file that maps the IP address of the new communication server (131.108.10.2) to the name `newcommserver`. The address of the new communication server was learned via SLARP and is based on *existing*'s IP address of 131.108.10.1.

```
ip host newcommserver 131.108.10.2
```

If `newcommserver` does not receive a `network-config` file, or if the IP address-to-host name mapping does not match the newly acquired IP address, `newcommserver` sends a Domain Name Service (DNS) broadcast. If DNS is configured and has an entry that maps `newcommserver`'s SLARP or BootP/RARP-acquired IP address to its name, `newcommserver` successfully resolves its name.

If DNS does not have an entry mapping `newcommserver`'s SLARP or BootP/RARP-acquired address to its name, the new communication server cannot resolve its host name. The new communication server 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 Communication Server's IP Address-to-Host Name Mapping



### Download the New Communication Server's Host Configuration File

After the communication server successfully resolves its host name, *newcommserver* sends a TFTP broadcast requesting the file `newcommserver-config`. The name `newcommserver-config` must be in all lowercase, even if the true host name is not. If *newcommserver* cannot resolve its host name, it sends a TFTP broadcast requesting the default host configuration file `communication server-config`. The file is downloaded to *newcommserver* 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 *newcommserver*, and then run the **setup** command to configure *newcommserver*. Refer to the *Communication Server Getting Started Guide* for details on the **setup** command.

If the host configuration file is complete, *newcommserver* should be fully operational. The administrator can enter the **enable** command (with the system administrator password) at the system prompt on *newcommserver*, and then issue the **write memory** command to save the information in the recently obtained configuration file into NVRAM. If a reload occurs, *newcommserver* simply loads its configuration file from NVRAM.

If the TFTP request fails, or if *newcommserver* still has not obtained the IP addresses of all its interfaces, and those addresses are not contained in the host configuration file, then *newcommserver* enters setup mode automatically. Setup mode prompts for manual configuration of the communication server via the console. The new communication server 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 *Communication Server Getting Started Guide* for details on the **setup** command facility.

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

## Perform the AutoInstall Procedure

To dynamically configure a new communication server 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 communication server's configuration to support the AutoInstall procedure.
- Step 2** Set up the TFTP server to support the AutoInstall procedure.
- Step 3** Set up BootP or RARP server if needed (required for AutoInstall using an Ethernet or Token Ring interface; not required for AutoInstall using an HDLC-encapsulated serial interface).
- Step 4** Connect the new communication server to the network.

## Modify the Existing Communication Server'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 or Token Ring 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 communication server:

Task	Command
<b>Step 1</b> Enter global configuration mode.	<b>configure terminal</b>
<b>Step 2</b> Configure the serial interface that connects to the new communication server with HDLC encapsulation (the default).	<b>interface serial <i>interface-number</i></b>
<b>Step 3</b> Enter an IP address for the interface. The host portion of the address must have a value of 1 or 2.	<b>ip address <i>address mask</i></b>
<b>Step 4</b> Configure a helper address for the serial interface to forward broadcasts associated with the TFTP, BootP, and DNS requests.	<b>ip helper-address <i>address</i></b>
<b>Step 5</b> 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.	<b>clockrate <i>bits per second</i></b>
<b>Step 6</b> Exit global configuration mode.	Ctrl-Z
<b>Step 7</b> Save the configuration changes to NVRAM.	<b>write memory</b>

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

### Example

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

```
communication server1# configure terminal
communication server1(config)# interface serial 0
communication server1(config)# ip address 131.108.10.1 255.255.255.0
communication server1(config)# ip helper-address 131.108.20.5
[Ctrl-Z]
communication server1# write memory
```

### Use an Ethernet or Token Ring Interface Connection

To set up AutoInstall using an Ethernet or Token Ring interface, complete the following tasks as needed to modify the configuration of the existing communication server. Typically, the LAN interface and IP address are already configured on the existing communication server. You might need to configure an IP helper address.

Task	Command
Enter configuration mode.	<b>configure terminal</b>
Configure a LAN interface.	<b>interface {ethernet   tokenring} interface-number</b>
Enter an IP address for the interface.	<b>ip address address mask</b>
Optionally, configure a helper address to forward broadcasts associated with the TFTP, BootP, and DNS requests.	<b>ip helper-address address</b>
Exit global configuration mode.	Ctrl-Z
Save the configuration changes to NVRAM.	<b>write memory</b>

### Example

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

```
communication server1# configure terminal
communication server1(config)# interface Ethernet 0
communication server1(config-if)# ip address 131.108.10.1 255.255.255.0
communication server1(config-if)# ip helper-address 131.108.20.5
[Ctrl-Z]
communication server1# write memory
```

### Set up the TFTP Server

For AutoInstall to work correctly, the new communication server must be able to resolve its host name, and then download a *name-confg* file from a TFTP server. The new communication server 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 describes different ways to resolve the new communication server's host name. Perform the first option if you want to use a network-config file to resolve the new communication server's host name. Perform step 3 if you want to use the DNS to resolve the new communication server's host name.

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

The *name*-config file can contain the new communication server'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 communication server to configure it. If you are using BootP or RARP to resolve the address of the new communication server, 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 communication server might need to forward TFTP requests and response packets if the TFTP server is not on the same network segment as the new communication server. When you modified the existing communication server's configuration, you specified an IP helper address for this purpose.

You can save a minimal configuration under a generic *name*-config file. Use the **ip host** command in the network.config file to specify the host name with the address you will be dynamically resolving. The new communication server should then resolve its IP address, host name and minimal configuration automatically. Telnet into the new communication server from one hop away and use the **setup** facility to configure the rest of the interfaces. For example, the line in the network-config file could be similar to the following:

```
ip host newcommserver 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 communication server one hop away. From this configuration, use the **setup** facility to configure the rest of the interfaces. If the communication server is more than one hop away, you must include routing information in the minimal configuration also.

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

```
ip host newcommsserver 131.108.10.2
```

### Set up the BootP or RARP Server

If the new communication server is connected to the existing communication server using an Ethernet or Token Ring interface, you must configure a BootP or RARP server to map the new communication server’s MAC address to its IP address. If the new communication server is connected to the existing communication server 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 communication server’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 communication server’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 communication server, use the **ip rarp-server** command to configure the existing communication server to act as a RARP server. See the section “Configure a Communication Server 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 Communication Server to the Network

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

Task	Command
Step 1 Enter privileged EXEC mode at the system prompt on the new communication server.	<b>enable</b> <i>password</i>
Step 2 Save the information from the <i>name-config</i> file into NVRAM.	<b>write memory</b>



**Caution** Verify that the existing and new communication servers 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 communication server will load NVRAM configuration information before it can enter AutoInstall mode.

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

Task	Command
Step 1 Establish a Telnet connection to the existing communication server.	<b>telnet</b> <i>existing</i>
Step 2 From the existing communication server, establish a Telnet connection to the new communication server	<b>telnet</b> <i>newcommserver</i>
Step 3 Enter privileged EXEC mode.	<b>enable</b>
Step 4 Enable the <b>setup</b> command facility to configure the new communication server.	<b>setup</b> (Refer to the <i>Communication Server Getting Started Guide</i> .)

## Enter Configuration Mode

To enter configuration mode, enter the EXEC command **configure** at the privileged-level EXEC prompt. The communication server 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 methods are described in the next three sections.

The communication server 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 communication server. However, you can list the comments in configuration files stored on a TFTP or MOP server.

## Configure the Communication Server from the Terminal

To configure the communication server from the terminal, complete the following tasks:

Task	Command
Step 1 Enter global configuration mode and select the terminal option.	<b>configure terminal</b>
Step 2 Enter the necessary configuration commands.	See the appropriate chapter for specific configuration commands.
Step 3 Quit global configuration mode.	Ctrl-Z
Step 4 Save the configuration file modifications to NVRAM.	<b>write memory</b>

### Example

In the following example, the communication server is configured from the terminal. The comment “The following command provides the communication server host name” identifies the purpose of the next command line. The **hostname** command changes the communication server name from communication server1 to communication server2. By pressing Ctrl-Z, the user quits configuration mode. The command **write memory** loads the configuration changes into NVRAM.

```
cs1# configure terminal
cs1(config)# !The following command provides the communication server host name.
cs1(config)# hostname communication server2
[Ctrl-Z]
cs2# 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 communication server startup software always checks for configuration information in NVRAM. If NVRAM holds valid configuration commands, the communication server executes the commands automatically at startup. If the communication server detects a problem with the nonvolatile memory or the configuration it contains, the communication

server 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 *Communication Server Getting Started Guide* for details on setup information.

## Configure the Communication Server from Nonvolatile Memory

You can configure the communication server from NVRAM by reexecuting the configuration commands stored in NVRAM. To do so, complete the following task in EXEC mode:

Task	Command
Configure the communication server from NVRAM.	<b>configure memory</b>

## Configure the Communication Server from a File on a Remote Host

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

Task	Command
<b>Step 1</b> Enter global configuration mode with the network option.	<b>configure network</b>
<b>Step 2</b> 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.	<b>host</b> or <b>network</b>
<b>Step 3</b> At the system prompt, enter the optional IP address of the remote host from which you are retrieving the configuration file.	<i>ip-address</i>
<b>Step 4</b> At the system prompt, enter the name of the configuration file or accept the default name.	<i>filename</i>
<b>Step 5</b> Confirm the configuration filename that the system supplies.	<b>y</b>

### Example

In the following example, the communication server is configured from the file `tokyo-config` at IP address 131.108.2.155:

```
cs1# 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-config]?
Configure using tokyo-config from 131.108.2.155? [confirm] y
Booting tokyo-config from 131.108.2.155:!! [OK - 874/16000 bytes]
```

## Modify the Configuration Register Boot Field

The order in which the communication server 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 communication servers 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 communication server. 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 communication server is reloaded. In ROM monitor mode, use the **o** command to list the value of the boot field in the configuration register.

## Specify the System Image the Communication Server Loads upon Restart

You can enter multiple boot commands in NVRAM configuration to provide backup methods for loading a system image onto the communication server. 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. The ASM-CS has Flash memory; the 500-CS does not.
- 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 communication server.
- 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 communication server tries them in the order they are entered.

### Loading from Flash Memory

Flash memory is available for the ASM-CS. With a CSC-MC+ Flash memory card and CSC-MCI controller and appropriate cables, system software images can be written to Flash memory for booting.

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 communication server 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).
- It allows a communication server 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

The following Flash memory security precautions are provided:

- Flash memory provides write protection against accidental erasing or reprogramming.
- The system image stored in Flash memory can be changed only from a privileged EXEC command session on the console terminal. This feature offers system-wide security.

## Flash Memory Configuration

The following list is an overview of how to configure your 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 your hardware manual 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** Boot the system using the **reload** command.

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 communication server to automatically boot from an image in Flash memory by completing the following tasks:

Task	Command
Step 1 Enter global configuration mode from the terminal.	<b>configure terminal</b>
Step 2 Enter the filename of an image stored in Flash memory	<b>boot system flash</b> [ <i>filename</i> ]
Step 3 Set the configuration register to enable loading of the system image from Flash memory.	Use the hardware configuration register
Step 4 Exit global configuration mode.	Ctrl-Z
Step 5 Save the configuration information to NVRAM.	<b>write memory</b>

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 communication server 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 wish 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.	<b>reload</b>

### Example

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

```

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





---

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

---

### Example

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

```
cs1# configure terminal
cs1(config)# boot system testme5.tester 131.108.13.111
[Ctrl-Z]
cs1# 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 global configuration mode from the terminal.	<b>configure terminal</b>
Step 2 Specify use of the ROM system image as a backup image.	<b>boot system rom</b>
Step 3 Set the configuration register to enable loading of the system image from ROM.	Use the hardware configuration register
Step 4 Exit global configuration mode.	Ctrl-Z
Step 5 Save the configuration information to NVRAM.	<b>write memory</b>

### Example

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

```
communication server1# configure terminal
communication server1(config)# boot system flash image1
communication server1(config)# boot system backup1 131.108.20.4
communication server1(config)# boot system rom
[Ctrl-Z]
communication server1# 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 might want to configure the communication server 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 global configuration mode from the terminal. <b>configure terminal</b>
Step 2	Configure the communication server to boot from Flash memory. <b>boot system flash</b> <i>[filename]</i>
Step 3	Configure the communication server to boot from a system filename. <b>boot system</b> <i>filename [address]</i>
Step 4	Configure the communication server to boot from ROM. <b>boot system rom</b>
Step 5	Set the configuration register to enable loading of the system image from a network server or Flash. Set the hardware configuration register
Step 6	Exit global configuration mode. Ctrl-Z
Step 7	Save the configuration information to NVRAM. <b>write memory</b>

### Example

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

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

Using this strategy, a communication server 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 Communication Server Loads upon Restart

Configuration files can be stored on network servers. You can configure the communication server 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 communication servers. 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 communication server 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 communication server scans this list until it successfully loads the appropriate network or host configuration file.

## Network Configuration File

To configure the communication server 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 communication server uses the default filename network-config.

You can specify more than one network configuration file. The communication server 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 global configuration mode from the terminal.	<b>configure terminal</b>
Step 2 Optionally, enter the network configuration filename.	<b>boot network mop</b> <i>filename</i> [ <i>MAC-address</i> ] [ <i>interface</i> ] <b>boot network [tftp]</b> <i>filename</i> [ <i>address</i> ]
Step 3 Enable the communication server to automatically load the network file upon restart.	<b>service config</b>
Step 4 Exit global configuration mode.	Ctrl-Z
Step 5 Save the configuration information to NVRAM.	<b>write memory</b>

## Host Configuration File

To configure the communication server 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 communication server uses its own name to form a host configuration filename by converting the communication server name to all lowercase letters, removing all domain information, and appending “-config.” If no host name information is available, the communication server uses the default host configuration filename cs-config.

You can specify more than one host configuration file. The communication server 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 global configuration mode from the terminal.	<b>configure terminal</b>
Step 2 Optionally enter the host configuration filename.	<b>boot host mop</b> <i>filename</i> [ <i>MAC-address</i> ] [ <i>interface</i> ] <b>boot host [tftp]</b> <i>filename</i> [ <i>address</i> ]
Step 3 Enable the communication server to automatically load the host file upon restart.	<b>service config</b>
Step 4 Exit global configuration mode.	Ctrl-Z
Step 5 Save the configuration information to NVRAM.	<b>write memory</b>
Step 6 Reset the communication server with the new configuration information.	<b>reload</b>

## Example

In the following example, the communication server is configured to boot from the host configuration file `hostfile1` and from the network configuration file `networkfile1`:

```
cs1# configure terminal
cs1(config)# boot host hostfile1
cs1(config)# boot network networkfile1
cs1(config)# service config
[Ctrl-Z]
cs1# 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-config... [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 communication server will enter the **setup** command facility. See the *Communication Server Getting Started Guide* for details on the **setup** command facility.

## 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 global configuration mode from the terminal.	<b>configure terminal</b>
Step 2 Change the buffer size to use for netbooting a host or network configuration file.	<b>boot buffersize bytes</b>
Step 3 Exit global configuration mode.	Ctrl-Z
Step 4 Save the configuration information to NVRAM.	<b>write memory</b>

## Example

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

```
cs1# configure terminal
cs1(config)# boot buffersize 50000
[Ctrl-Z]
cs1# write memory
```

## Manually Load a System Image

If your communication server does not find a valid system image, or if its configuration file is corrupted at startup, 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 communication server 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 communication server.	<b>reload</b>
Step 2 Press the Break key during the first 60 seconds while the system is starting up.	Break
Step 3 Manually boot the communication server.	<b>b flash</b> [ <i>filename</i> ]

### Examples

In the following example, the communication server 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
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR [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
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR [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 communication server.	<b>reload</b>
Step 2 Press the Break key during the first 60 seconds while the system is starting up.	Break
Step 3 Manually boot the communication server.	<b>b filename [address]</b>

### Example

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

```
>b network1
```

## Manually Booting from ROM

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

Task	Command
Step 1 Restart the communication server.	<b>reload</b>
Step 2 Press the Break key during the first 60 seconds while the system is starting up.	Break
Step 3 Manually boot the communication server from ROM.	<b>b</b>

### Example

In the following example, the communication server is manually booted from ROM:

```
>b
```

## Manually Booting Using MOP

To interactively boot system software using MOP, perform the following task:

Task	Command
Interactively boot system software using MOP.	<b>b mop</b>

### Example

In the following example, the image is interactively booted using MOP:

```
> b mop
```

## Configure a Communication Server as a TFTP Server

As a TFTP server host, the communication server 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 communication server’s configuration.

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

- If you do not specify a *filename* in the **tftp-server system** command, the TFTP request is rejected.
- 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 communication server, complete the following tasks:

Task	Command
Step 1 Enter global configuration mode from the terminal.	<b>configure terminal</b>
Step 2 Specify TFTP server operation.	<b>tftp-server system filename [IP-access-list]</b>
Step 3 Exit global configuration mode.	Ctrl-Z
Step 4 Save the configuration information to NVRAM.	<b>write memory</b>

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 communication server 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 Communication Server as a RARP Server

You can configure the communication server as a Reverse Address Resolution Protocol (RARP) server. With this feature, RARP requests can be answered by the communication server, thereby allowing the communication server 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.

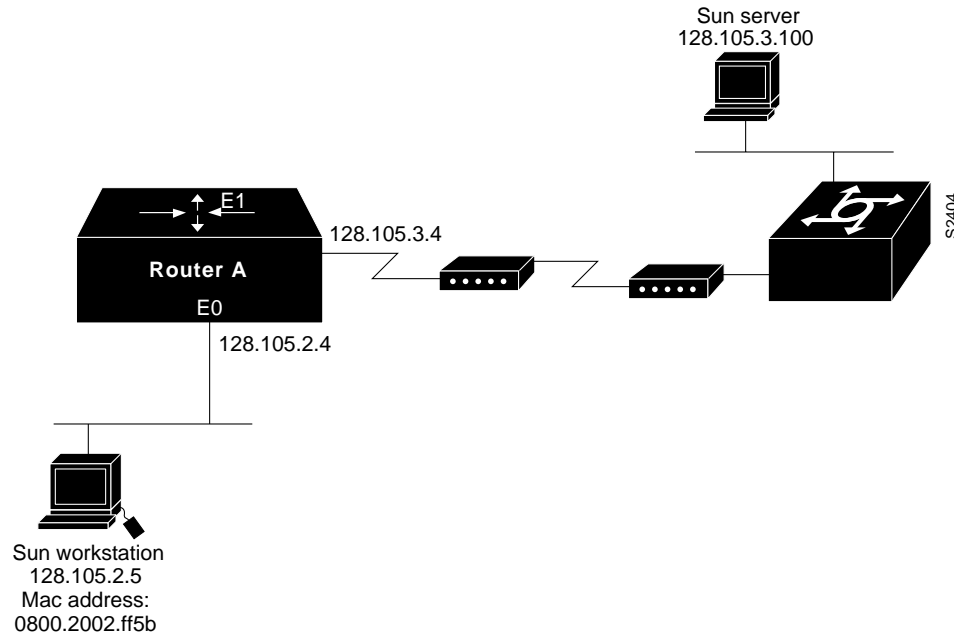
To configure the communication server as a RARP server, perform the following task:

Task	Command
Configure the communication server as a RARP server.	<b>ip rarp-server address</b>

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



Figure 1-4 Configuring a Communication Server as a RARP Server



```

! Allow the communication server to forward broadcast portmapper requests
ip forward-protocol udp 111
! Provide the communication server with the IP address of the diskless sun
arp 128.105.2.5 0800.2002.ff5b arpa
interface ethernet 0
! Configure the communication server 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.	<b>async-bootp</b> 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.	<b>show async-bootp</b>

## Specify MOP Server Boot Requests

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

Task	Command
<b>Step 1</b> Enter configuration mode from the terminal.	<b>configure terminal</b>
<b>Step 2</b> Change MOP server parameters.	<b>mop retransmit-timer</b> <i>seconds</i> <b>mop retries</b> <i>count</i>
<b>Step 3</b> Exit configuration mode.	Ctrl-Z
<b>Step 4</b> Save the configuration information to NVRAM.	<b>write memory</b>

By default, when the communication server 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 communication server are separated by a slow serial link, it may take longer than four seconds for the communication server to receive a response to its message. Therefore, you might want to configure the communication server 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 communication server sends a message, the communication server 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
<b>Step 1</b> 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.
<b>Step 2</b> Copy a system image to Flash memory.	<b>copy tftp flash</b>
<b>Step 3</b> When prompted, enter the IP address or domain name of the server.	<i>ip address or name</i>
<b>Step 4</b> When prompted, enter the filename of the server system image.	<i>filename</i>

---

**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.

---

Once you enter the **copy tftp flash** command, the system prompts you for the IP address (or domain name) of the TFTP server. This can be another communication server 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. 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, regardless of case, is copied to Flash, 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...
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
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...
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
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
```



## 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
<b>Step 1</b> Learn the exact spelling of the system image in Flash memory.	<b>show flash all</b>
<b>Step 2</b> Copy the system image in Flash memory to a TFTP server.	<b>copy flash tftp</b>
<b>Step 3</b> When prompted, enter the IP address or domain name of the TFTP server.	<i>ip address or name</i>
<b>Step 4</b> When prompted, enter the filename of the system image in Flash memory.	<i>filename</i>

### 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.

```

Communication Servercs048K 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]

Communication Server# 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.
Communication Server#
    
```

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 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, 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.

## Copy a Configuration File from the Communication Server to a Network Server

To store configuration information on a network server, complete the following tasks:

Task	Command
<b>Step 1</b> Specify that the communication server configuration file in NVRAM should be stored on a network server.	<b>write network</b>
<b>Step 2</b> When prompted, enter the IP address of the network server (remote host).	<i>ip-address</i>
<b>Step 3</b> Enter the name of the configuration file to store on the server.	<i>filename</i>
<b>Step 4</b> Confirm the entry.	<b>y</b>

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 communication server to a server:

```
Tokyo# write network
Remote host [131.108.2.155]?
Name of configuration file to write [tokyo-config]?
Write file tokyo-config on host 131.108.2.155? [confirm] y
Writing tokyo-config.
[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.	<b>show version</b>
List the configuration information stored in NVRAM.	<b>show configuration</b>
List the configuration information in running memory.	<b>write terminal</b>
List information about Flash memory, including system image filenames and amounts of memory used and remaining.	<b>show flash</b>
List information about Flash memory, including all the information displayed by the <b>show flash</b> command, plus information about vendor, location, individual ROM devices in Flash memory, and invalidated system image files.	<b>show flash all</b>

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

## 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.	<b>write erase</b>

## 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.	<b>configure memory</b>

## Use Flash Memory as a TFTP Server

Flash memory can be used as a Trivial File Transfer Protocol (TFTP) file server for other communication servers on the network. This feature allows you to boot a remote communication server with an image that resides in the Flash server memory.

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

### Prerequisites

The Flash server and client communication server 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 communication server (in either direction) using the **ping** command.

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

```
cs# ping 131.131.101.101 <Return>
```

In this example, the Internet Protocol (IP) address of 131.131.101.101 belongs to the client communication server. 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 communication server, 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 communication server 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 communication server 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 communication server. 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:

Task	Command
Configure the flash server.	<b>configure terminal</b>

### 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, CTRL/W, and CTRL/U;end with CTRL/Z
tftp-server system gs7-k.9.21 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 within which the client communication server resides. Thus, in the example, the network 131.131.101.0 and any client communication servers on it are permitted access to the Flash server filename gs7-k.9.21.



**Caution** Using the **no boot system** command in the following example will invalidate *all* other boot system commands currently in the client communication server system configuration. Before proceeding, determine whether the system configuration stored in the client communication server should first be saved (uploaded) to a TFTP file server so you have a backup copy.

## Configuring the Client Communication Server

To configure the client communication server, perform the following tasks:

Task	Command
Enter configuration mode.	<b>configure terminal</b>
Invalidate all other <b>boot system</b> commands in configuration memory.	<b>no boot system</b>
Look for this file and address. This is the boot file information.	<b>boot system <i>filename address</i></b>
Boot from ROM if the <b>boot system</b> file cannot be found.	<b>boot system rom</b>

## Example

Following is an example of the use of these commands:

```
Client# configure terminal
Enter configuration commands, one per line.
Edit with DELETE, CTRL/W, and CTRL/U;end with CTRL/Z
no boot system
boot system gs7-k.9.21 131.131.111.111
boot system rom
^Z
Client# write memory <Return>
[ok]
Client# reload
```

For this exercise, the IP address of the Flash server is 131.131.111.111.

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 communication server to look for the file `gs7-k.9.21` in the (Flash) server with an IP address of 131.131.111.111. Failing this, the client communication server 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.9.21`) 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 communication server will boot the Flash server's system ROM.

Use the **show version** command on the client communication server 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 9.1.21
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.9.21", 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 communication server (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.

