Chapter 4
Configuring the System

4

This chapter describes how to configure the system. These tasks include:

- Setting global system characteristics such as the host name and console banner message
- Defining the size of the system buffers
- Changing the system boot file specifications
- Obtaining the boot file over the network
- Storing and booting system software images with the Flash Memory card (CSC-MC+)
- Establishing system and line passwords and system security
- Defining network services such as the IP Finger protocol
- Enabling and directing the logging of system debugging messages
- Configuring the console and virtual terminal lines

This chapter concludes with alphabetical summaries of the commands described in this chapter.

Configuring the Global System Parameters

The following sections contain procedures and command descriptions for configuring the global system characteristics: host name and passwords, and configuring system security and system management functions. The global configuration commands described in the following sections are entered in configuration mode. See the section "Entering Configuration Mode" in the chapter "First-Time Startup and Basic Configuration" for the procedures to enter into this mode.

Setting the Host Name

Use the **hostname** global configuration command to specify the host name for the network server, which is used in prompts and default configuration file names.

hostname *name*

The argument *name* is the new host name for the network server and is case sensitive. The default host name is *Router*.

Example:

This command changes the host name to *sandbox*.

```
hostname sandbox
```

Displaying Banner Messages

A banner is the message that the EXEC command interpreter displays whenever a user starts any EXEC process or activates a line. The general form of the **banner** global command follows.

banner {**motd**|**exec**|**incoming**} *c text c*
no banner {**motd**|**exec**|**incoming**} *c text c*

The **motd**, **exec**, and **incoming** keywords control when the banner message is displayed. The use of these keywords is described in the following sections.

The argument *c* specifies a delimiting character of your choice. The argument *text* specifies the message to be shown on the screen whenever an interface line is activated.

The default keyword is **motd** if none is specified.

The **no** version of these commands removes the specified banner, and the **no banner** command removes the motd.

Follow **banner** with one or more blank spaces, then type the delimiting character, followed by one or more lines of text, terminating the message with the second occurrence of the delimiting character. There is no limit to the amount of characters that can be used for the banner, with the exception of buffer limits and what is appropriate for a banner.

Example:

The following example uses the pound sign character as a delimiting character:

```
banner motd #
Building power will be off from 7:00 AM until 9:00 AM this coming Tues-
day.
#
```

Note: You cannot use the delimiting character in the banner message.

Displaying a Message-of-the-Day Banner

To specify a general-purpose message-of-the-day (motd) type banner, use the **banner motd** global configuration command.

banner motd *c text c*

This displays a message-of-the-day type banner whenever any type of connection is established, for example, when a line is activated, or when an incoming Telnet connection is created. Use this banner for messages that affect all users of the router (for example, for system reboots).

Note: The command **banner** is equivalent to the command **banner motd**, except that the banner is displayed on incoming connections.

Displaying a Banner with an EXEC Process

To display a message when an EXEC process is created, use the **banner exec** global configuration command.

banner exec *c text c*

This command specifies a message to be displayed when an EXEC process is created (for example, when a TTY line is activated, or an incoming connection is established to a VTY). This banner is designed for messages that affect only interactive terminal users of the router.

Displaying an Incoming Message Banner

To display an incoming message to a particular terminal line, use the **banner incoming** global configuration command.

banner incoming *c text c*

This specifies a message to be displayed on incoming connections to particular terminal lines (for example, lines used for "milking machine" applications).

Note: Messages are never displayed on incoming stream type connections, as they might interfere with printer daemons.

The EXEC banner can be suppressed on certain lines using the **no exec-banner** line subcommand (described in the section "Suppressing Banner Messages" later in this chapter). Lines so configured will not display the EXEC or MOTD banners when an EXEC is created.

Order of Banner Displays

Banners and messages are displayed in the following order:

1. Any **banner motd** message
2. Any **banner incoming** message

   At this point, the user logs in, if required.
3. Any **banner exec** message

Example:

This example illustrates how to display a message-of-the-day banner, and a message that will be displayed when an EXEC process is created. Use the **banner** global configuration commands and **no exec-banner** line subcommand to accomplish these settings.

```
! Both messages are inappropriate for the VTYs.
line vty 0 4
no exec-banner
!
banner exec /
This is Cisco Systems training group server.
Unauthorized access prohibited.
/
!
banner motd /
The server will go down at 6pm for a software upgrade
/
```

Setting the System Buffers

In normal system operation, there are several pools of different sized buffers. These pools grow and shrink based upon demand. Some buffers are temporary and are created and destroyed as warranted. Other buffers are permanently allocated and cannot be destroyed. The **buffers** command allows a network administrator to adjust initial buffer pool settings, as well as the limits at which temporary buffers are created and destroyed.

Note: It is normally not necessary to adjust these parameters; do so only after consulting with Cisco staff. Improper settings could adversely impact router performance.

The full syntax of this command follows:

buffers {**small**|**middle**|**big**|**large**|**huge**} {**permanent**|**max-free**|**min-free**|**initial**} *number*
no buffers {**small**|**middle**|**big**|**large**|**huge**} {**permanent**|**max-free**|**min-free**|**initial**} *number*

First choose the keyword that describes the size of buffers in the pool—small, big, huge, and so on. The default number of the buffers in a pool is determined by the hardware configuration, and can be displayed with the EXEC **show buffers** command.

The next keyword specifies the buffer management parameter to be changed, and can be one of the following:

- **permanent**—The number of permanent buffers that the system tries to allocate. Permanent buffers are normally not deallocated by the system.
- **max-free**—The maximum number of free or unallocated buffers in a buffer pool.
- **min-free**—The minimum number of free or unallocated buffers in a buffer pool.
- **initial**—The number of additional temporary buffers which should be allocated when the system is reloaded. This can be used to ensure that the router has necessary buffers immediately after reloading in a high traffic environment.

The argument *number* specifies the number of buffers to be allocated.

The **no buffers** command with appropriate keywords and argument restores the default buffer values.

Dynamic Buffer Sizing

An optional global configuration command for adjusting huge buffer settings is the **buffers huge size** command. As with the above command, use only after consulting with Cisco staff:

buffers huge size *number*
no buffers huge size *number*

The **buffers huge size** command dynamically resizes all huge buffers to the value that you supply. The buffer size cannot be lowered below the default. The argument *number* specifies the number of buffers to be allocated.

The **no** version of the command with the keyword and argument restores the default buffer values.

Examples:

In the following example, the system will try to keep at least 50 small buffers free.

```
buffers small min-free 50
```

In this example, the system will try to keep no more than 200 medium buffers free.

```
buffers middle max-free 200
```

With the following command, the system will try to create one large temporary extra buffer, just after a reload:

```
buffers large initial 1
```

In this example, the system will try to create one permanent huge buffer:

```
buffers huge permanent 1
```

In this example, the system will resize huge buffers to 20000 bytes:

```
buffers huge size 20000
```

To display statistics about the buffer pool on the network server, use the command **show buffers**. For more information, refer to the section "Monitoring System Processes" in the chapter "Managing and Monitoring the System."

Setting Configuration File Specifications

This section describes the **boot** global configuration commands used to configure boot files. The **boot** command can be used to perform these tasks:

- Change default file names.
- Specify a server host for netbooting configuration files and boot image files.
- Specify the size buffer to configure for netbooting a host or network configuration file.

The commands to load files over the network take effect the next time the software is reloaded, provided they have been written into nonvolatile memory.

Changing the Network Configuration File Name

The network configuration file contains commands that apply to all network servers and terminal servers on a network. The default name of this file is *network-confg*. See the section "Entering Configuration Mode" in the chapter "First-Time Startup and Basic Configuration." To change the name of this file use the **boot network** global configuration command. The full command syntax follows:

boot network *filename* [*address*]
no boot network [*filename address*]

The keyword **network** changes the network configuration file name from *network-confg*. The argument *filename* is the new name for the network configuration file. If you omit the argument *address,* the network server uses the default broadcast address of *255.255.255.255*. If you use *address,* you can specify a specific network host or a subnet broadcast address.

Changing the Host Configuration File Name

The host configuration file contains commands that apply to one network server in particular. To change the host configuration file name, use the **boot host** global configuration command. The full command syntax follows:

boot host *filename* [*address*]
no boot host [*filename address*]

The keyword **host** changes the host configuration file name to a name you specify in the *filename* argument. The network server uses its name to form a host configuration file name. To form this name, the network server converts its name to all lowercase letters, removes all domain information, and appends "-confg." By default, the host file name is *router-confg*.

Obtaining the Boot File over the Network

New versions of the software can be downloaded over the network. Use the **boot system** global configuration command to do this. The full command syntax follows.

boot system *filename* [*address*]
no boot system [*filename address*]

The keyword **system** indicates that the file name and host addresses for booting operating software over the network are in the nonvolatile memory. In this case, the argument *filename* is the file name of the operating software to load, and the argument *address* is the address of the network host holding that file.

Note: The file names of *flash* and *rom* are not allowed, as they are used to indicate that the Flash Memory, or system ROMs, respectively, are to be used for booting system images.

The **boot system** command overrides the processor configuration register setting unless the register specifies the use of default (ROM) operating software. Therefore, to permit netbooting, set the configuration register bits on the processor card to any pattern other than 0-0-0-0 or 0-0-0-1.

Refer to your Cisco hardware installation and reference publications for more information about the processor configuration registers.

Note: The IGS requires 4 MB of RAM to netboot.

Example:

To use the nonvolatile memory option to specify netbooting, place a **boot system** command in the nonvolatile memory. You use this command to specify both the file name of the operating software to load, and the Internet address of the server host holding that file:

```
boot system /usr/local/tftpdir/cisco.ts2 192.7.31.19
```

Manually Booting from ROM

Use the **b** command at the ROM monitor prompt (>) to manually boot the system from the ROM software. The syntax is as follows:

b

The following is an example of the **b** command output:

```
>b
F3:
{ROM Monitor copyrights}
```

Manually Netbooting

Use the **b** command at the ROM monitor prompt (>) to manually netboot the system, as in the following example. Check the appropriate hardware manual for the correct jumper or configuration register setting. The syntax for TFTP netbooting is as follows:

b *filename* [*address*]

The *filename* argument specifies the filename of the image you want loaded. It is case sensitive. The *address* argument is optional and defines the IP address of the host you want to boot from. The following is an example of the **b** command for manually netbooting:

```
>b testme4.test 131.108.15.112
F3:
{ROM Monitor copyrights}
```

Specifying ROMs as a Source of the Router System Image

When netbooting, with or without Flash memories, you can specify that the ROM image is to be booted if other boot images are not available.

boot system rom
no boot system rom

Use the **boot system rom** global configuration command to specify the use of the ROM system image when other **boot system** commands exist in the configuration.

For example, a list specifying two possible internet locations for a system image, with the ROM software being used as a backup, is given below:

```
boot system gs3-bfx.90-1 192.31.7.24
boot system gs3-bfx.83-2 192.31.7.19
boot system rom
```

Specifying a Boot File Buffer Size

To specify the size of the buffer to be used for netbooting a host or a network configuration file, use the **boot buffersize** global configuration command. The full command syntax follows:

boot buffersize *bytes*
no boot buffersize *bytes*

The argument *bytes* specifies the size of the buffer to be used. By default, it is the size of your nonvolatile memory; it is 32 kilobytes if you do not have nonvolatile memory. There is no minimum or maximum size that may be specified.

The EXEC commands **write terminal** and **write network** use the information specified by the **buffersize** keyword when performing their functions (see the section "Entering Configuration Mode" in the chapter "First-Time Startup and Basic Configuration" for more information about these EXEC commands).

Configuring Multiple Instances of the Boot Commands

You can configure multiple instances of the **boot** commands. When issued, each command is executed in order and so can be used to begin a systematic search or to build a specific list. For example, you can issue multiple **boot** commands to build an ordered list of configuration-file-name-and-host-address pairs. The network server scans this list until it successfully loads the appropriate network or host configuration file or system boot image. In this example, the network server looks first for *fred-confg* on network 192.31.7.24 and, if it cannot load that file, then for *wilma-confg* on network 192.31.7.19:

```
boot host /usr/local/tftpdir/fred-confg 192.31.7.24
boot host /usr/local/tftpdir/wilma-confg 192.31.7.19
```

Note: This example uses fictitious file names; the syntax of these file names depends on the TFTP server you are loading the files from.

If the network server cannot find either file, a background process tries at ten-minute intervals (default) to load one or the other of the files.

You may issue multiple instances of all variations of the **boot** command, including the **no boot** forms. This feature can be useful for removing configuration files. To remove a configuration-file-name-and-host-address pair from the list, use the **no boot** command syntax.

Troubleshooting Information when Netbooting

Cisco routers support netbooting over both TFTP and MOP across all supported media types such as Ethernet, FDDI, serial, Token Ring, and HSSI. During a netbooting session, routers behave like hosts; they route via proxy ARP or a default gateway. However, when netbooting, a router ignores routing information, static IP routes, and bridging information. As a result, intermediate routers are responsible for handling ARP and TFTP requests correctly. For serial and HSSI media, ARP is not used.

If you need to netboot from a server, it is recommended that you ping the server from the ROM software. If you are unable to ping the server, there is a problem with the server configuration or hardware. Contact your technical support representative for assistance. See "Useful Information to Provide Technical Support" later in this section for details.

The list that follows contains solutions to common problems that occur when netbooting. Note that these solutions only apply if you were able to successfully ping the server.

Client ARP Request Times Out

When netbooting, the client you netboot from sends an ARP request to the server over every available appropriate network interface (such as an Ethernet port or a Token Ring port). The client expects the server or a router to return an ARP response. If the client does not receive an ARP response from the server or a router, a message similar to the following displays at the client console:

```
Booting gs3-bfx.................[timed out]
```

One possible cause of this message is that intermediate routers are not performing proxy arp. Look for **no ip proxy-arp** in the configuration of the intermediate router. Another possible reason for this message appearing at the client is that the client is using a broadcast address, and the intermediate router does not have an IP helper address defined to point to the TFTP server.

Timeouts and Out-of-Order Packets

When netbooting, it is not unusual for the client to send additional requests before receiving a response to the initial ARP request. This can result in timeouts, out-of-order packets, and multiple responses. Timeouts (shown as periods on a netbooting display) and out-of-order packets (shown as Os) do not necessarily prevent a successful boot. It is acceptable to have timeouts and out-of-order packets. The following examples show successful boots even though a timeout and out-of-order packets have occurred:

```
Booting gs3-bfx from 131.108.1.123: !.!!!!!!!!!!!!!!!!!!!!
```

```
Booting gs3-bfx from 131.108.1.123: !0.0!!!!!!!!!!!!!!!!!!!!
```

Note that intermittent timeouts and out-of-order packets may occur throughout a netbooting session without being cause for concern. Excessive timeouts and out-of-order packets can be caused by bad routing paths on the intermediate routers, an extremely slow server, problems caused by multiple paths, or noise on the line. If your netbooting session appears to have excessive timeouts and out-of-order packets, contact your technical support representative and report the problem. Before calling technical support, you will need to gather some information. See "Useful Information to Provide Technical Support" that follows for details.

Useful Information to Provide Technical Support

Collect the following information for the technical support representative:

- ROM images
- NVRAM configurations for client and adjacent routers
- Debugging output from the adjacent router using the following commands:
  - **debug arp**
  - **debug ip-udp**

Change the last sentence in the first note on page 4-13 to read, "A failure message, `buffer overflow - xxxx/xxxx`, will appear, where xxxx/xxxx is the number of bytes used/total number of bytes."

Storing and Booting System Software Using the Flash Memory Card

The Flash Memory card is an add-in card of Flash memory storage onto which system software images may be stored, booted, and rewritten as necessary. This card is also called the CSC-MC+. The Flash card provides a more fault-tolerant solution to users who only netboot. The Flash card reduces the effects of network failure on system netbooting.

The CSC-MC+ Flash Memory card allows you to:

- Copy the TFTP image to Flash memories
- Automatically or manually boot a router from Flash memories
- Copy the Flash memory image to a TFTP server

The Flash capability requires the appropriate level of system software, firmware, and hardware. A number of prerequisites and caveats apply to the installation and use of Flash as well. Refer to the *Modular Products Hardware Installation and Reference* publication for complete hardware requirements, specifications, caveats, and step-by-step installation instructions.

Note: Use of the Flash Memory card is subject to the terms and conditions of the software license agreement that accompanies your Cisco product.

The features of the Flash Memory card include the following:

- The Flash card can be remotely loaded with multiple system software images through a TFTP transfer.
- The Flash Memory card provides 4 MB Flash memory storage.
- The CSC-MC+ 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.
- The CSC-MC+ memory card can be used by all A+, A, M, and C chassis platforms, and does not require a chassis card slot.
- The CSC-MC+ provides write-protection against accidental erasure or reprogramming of the Flash memories.

- The CSC-MC+ can store up to 32 kilobytes of configuration memory information in NVRAM.
- The NVRAM chips on the CSC-MC+ card have built-in lithium batteries for NVRAM backup in case of power failure. These batteries are designed to last ten years and store data (unpowered) for five years.

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.

Taking Security Precautions

The Flash Memory card provides write-protection against accidental erasure or reprogramming of the Flash memories. The write-protect jumper, located on the front edge of the card, may be removed in order to prevent reprogramming of the Flash memories. The jumper must be installed when programming is required. The system image stored in the Flash Memory card can only be changed from a privileged EXEC command session on the console terminal, which offers system-wide security as well. In general, Cisco does not recommend this feature for remote systems.

Caution: Be sure to save your configuration before you install the Flash card. Download the running configuration to a TFTP file server. This download will save the configuration that had been stored on a CSC-MC card. If your system configuration file has been stored on a CSC-ENVM or a CSC-MT card, the NVRAM on either of these cards would be used, because the nonvolatile memory (NVRAM) on the CSC-MC+ would be ignored.

Configuration Overview

The following is an overview of configuring your system to boot from Flash memories. This is not a step-by-step set of instructions; rather, it is an overview to the process of using the Flash capability. Refer to the *Modular Products Hardware Installation and Reference* publication for complete instructions on installing the hardware and netbooting, and in particular, any jumper setting changes.

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 memories.

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

Step 5: Write your configuration to memory.

Step 6: Set your system to netboot from a file name (requires jumper setting change).

Step 7: Power-cycle and reboot your system to ensure that all is working as expected and that the configuration is stored in NVRAM.

Once you have successfully installed and tested the CSC-MC+ card, you may want to configure the system with the **no boot system flash** command in order to revert back to booting from ROM.

The remainder of this section describes the configuration commands used with the Flash feature.

Verifying Installation and Displaying Flash ROM Statistics

To verify that the appropriate system card is properly connected to the CSC-MC+ card, use the **show flash** or **show flash all** commands.

The **show flash** command displays the total amount of Flash memory present on the Flash card, the type of card connected to the Flash card, any files that may currently exist in Flash memory and their size, and the amounts of Flash memory used and remaining.

The **show flash all** command displays all the above information, but also shows all the information about each Flash memory device.

Once you configure Flash, the **show flash** or **show flash all** commands will display the names of the system software images.

show flash

show flash all

The following shows sample output of the **show flash** command:

```
George#show flash
4096K bytes of flash ROM on MC+ (via MCI)
Contains:
gsxx (1086414)
tsyy (1086414)
[935053/4194304]
```

The following shows sample output of the **show flash all** command:

```
George#show flash all
4096K bytes of flash ROM on MC+ (via MCI)
   ROM 0, U2 , code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 1, U19, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 2, U3 , code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 3, U20, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 4, U4 , code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 5, U21, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 6, U5 , code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 7, U22, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 8, U9 , code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 9, U26, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 10, U10, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 11, U27, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 12, U11, code 0x89BD, size 0x40000, name INTEL 28F020
```

```
   ROM 13, U28, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 14, U12, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 15, U29, code 0x89BD, size 0x40000, name INTEL 28F020
Contains:
gsxx (1086414)[invalidated]
gsxx (1086414)
tsyy (1086414)
[935053/4194304]
```

Note the `[invalidated]` flag at the end of the second example. This flag will appear when a file is rewritten (recopied) into Flash memory, a user aborts, in a network timeout, or a Flash memory overflow. A prompt will tell you that the identical file already exists and that it will be invalidated. The first (now invalidated) copy of the file is still present within Flash memory, but it is rendered unusable in favor of the newest version.

To eliminate any files from Flash (invalidated or otherwise) and free up all available memory space, the entire Flash memory must be erased; individual files cannot be erased from Flash memory.

Both examples illustrate that the Flash memory can store and display multiple, independent software images (`gsxx` and `tsyy`) for booting itself or for TFTP serving software for other Cisco products. This feature would be most useful for storing default system software as a back-up. These images may also be stored in compressed format.

In the second example, `ROM 0` is at location `U2` on the Flash Memory card. The `code` is a vendor code, `size` is in hex bytes, `INTEL` is the vendor name, and the last number is the chip part number.

Copying the TFTP Image to Flash Memories

The **copy tftp flash** command copies (writes) a TFTP image into the current Flash configuration:

copy tftp flash

Caution: The TFTP image copied to the Flash memories must be at least system software Version 9.0 or later. If earlier system software is copied into the Flash memories, the CSC-MC+ card will not be recognized by the host processor card upon the next reboot.

Note: You must make certain there is ample space available before copying a file to Flash. Use the **show flash** command and compare the size of the file you wish to copy to the amount of available Flash memory shown. If the space available is less than the space required by the file you wish 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 give the **copy tftp flash** command, the system prompts you for the IP address (or domain name) of the TFTP server. This may be another Cisco router serving ROM or Flash system software images. You are then prompted for the file name of the software image and you are given the option to erase the existing Flash memory before writing onto it *only* when there is free space available in Flash memory. *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.* You will be prompted for these conditions by the system. The Flash memory is erased at the factory before shipment.

Following is sample output (copying a system image named `gsxx`) of the prompt you will see under these conditions:

```
George#copy tftp flash
IP address or name of remote host [255.255.255.255]? 131.131.101.101
Name of tftp filename to copy into flash []? gsxx
copy gsxx from 131.131.101.101 into flash memory? [confirm]
Flash is filled to capacity. (this line only appears if Flash memory is full)
Erasure is needed before flash may be written.
Erase flash before writing? [confirm]
```

Note: Entering "n" after the "Erase flash before writing?" prompt would continue with the copy process. Entering a "y" would proceed with the erase routine. Make certain you have ample Flash memory space before entering "n" at the erasure prompt.

When you erase the existing Flash, the system clears and initializes each Flash memory, and a pound sign (#) prompt is displayed for each cleared and initialized device (total of 16). The entire copying process takes several minutes and will differ from network to network.

Note: The file name gsxx can be in lower or upper case; the system will see GSxx as gsxx. If both are copied to Flash, the second file copied will become the valid file.

Following is sample output from copying a system image, named `gsxx`, into the current Flash configuration:

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

```
Flash verification successful.  Length = 324572, checksum = 0xE2E2
```

Note: To abort this copy process, press the Ctrl, Shift, and 6 keys simultaneously. The process will abort, however, the partial file copied before the abort was issued will remain until the entire Flash memory is erased.

The series of !'s in the above sample output indicate that the copying process is taking place. The series of V's indicate that a checksum verification of the Flash memory is occurring as it is loaded into memory for boot. This is verified only through data compare during programming of the Flash memory. The last line in the sample configuration indicates that the copy is successful.

Having successfully copied an image onto the Flash, the output of **show flash all** will provide the image name, as in the following sample output:

```
George#show flash all
4096K bytes of flash ROM on MC+ (via MCI)
   ROM 0, U2 , code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 1, U19, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 2, U3 , code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 3, U20, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 4, U4 , code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 5, U21, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 6, U5 , code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 7, U22, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 8, U9 , code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 9, U26, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 10, U10, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 11, U27, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 12, U11, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 13, U28, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 14, U12, code 0x89BD, size 0x40000, name INTEL 28F020
   ROM 15, U29, code 0x89BD, size 0x40000, name INTEL 28F020
Contains:
gsxx (1622132)
[2572172/4194304 bytes free]
```

During the actual copy process, the yellow LED on the CSC-MC+ will be lit, indicating the security (write-protect) jumper is installed. If the security jumper was removed from the Flash card, this would be indicated with the **show flash** command.

Following is sample out put of this write-protected condition:

```
George#show flash
flash memory on MC+ card (via MCI)
security jumper is uninstalled, so flash memory is read-only
```

In this condition, no files could be copied to Flash until the jumper was reinstalled.

You are now ready to boot from Flash. The following sections describe how to automatically and manually boot from the Flash Memory card.

Automatically Booting from Flash Memories

You can automatically boot the system from the Flash memories using the **boot system flash** system configuration command:

boot system flash *filename*
no boot system flash *filename*

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

The **boot system flash** command boots the first valid file in Flash memory. The **boot system flash** *filename* boots the file specified by this file name.

Configure the system to automatically boot from the desired file in Flash memory using the **configure terminal** command and the **boot system flash** *filename* system configuration command.

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

Write the configuration to NVRAM with the **write memory** command. Following this, boot the system with the **reload** command.

The following shows use of this sequence of commands:

```
George#configure terminal
boot system flash gsnew-image
^Z
George#write memory
[ok]
George#reload
[confirm]
System Bootstrap, Version 4.5(0.3)
Copyright (c) 1986-1990 by cisco Systems
CSC3 processor with 4096 Kbytes of memory
F3: 1578668+35572+156084 at 0x1000
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRR.... [OK - 162132/2110400 bytes]
F3: 1586072+36028+156004 at 0x1000
{ROM Monitor copyrights...}
```

Manually Booting from Flash Memories

Use the **b flash** command at the ROM monitor level to manually boot the system, as in the following example. Check the appropriate hardware manual for the correct jumper setting.

boot flash [*filename*]

The optional *filename* argument specifies the filename of the image you want loaded. It is case sensitive.

```
>b flash
F3: 1578668+35572+156084 at 0x1000
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRR.... {reading/booting flash from rom monitor}
{ROM Monitor copyrights...}
```

Copying the Flash Memory Image to a TFTP Server

To copy an image back to a TFTP server use the **copy flash tftp** command. This copy of the system image can serve as a backup copy and may also be used to verify that the copy in Flash is the same as on the original file on disk.

copy flash tftp

The following is an example of the use of this system configuration command:

```
George#copy flash tftp
IP address of remote host [255.255.255.255]? 101.2.13.110
Name of file to copy []?gsxx
writing gsxx !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
George#
```

Note: A domain name can be used in place of the IP address of the remote host.

Once you have successfully installed and tested the CSC-MC+ card and configured Flash memory, you may want to configure the system (**configure terminal**) with the **no boot system flash** configuration command in order to revert to booting from ROM. After this command is entered, use the **write memory** command to save this new configuration command to NVRAM. This also requires changing the jumper on the configuration register of the processor. Refer to the appropriate hardware manual.

Configuring Flash Memory as a TFTP Server

In the description that follows, one router with a Flash Memory card installed will be referred to as the Flash server; all other routers will be referred to as client routers. The configurations for the Flash server and client routers will be given through example configurations of each, with commands included 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.

The syntax for 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. In response to this command, `!!!!!` indicates a connection, while `... [timed out]` or `[failed]` indicates none. If the connection fails, reconfigure the interface, check the physical connection between the Flash server and client router, and ping again.

After this connection is verified, ensure that a TFTP-bootable image is present in Flash memory. This is the system software image that the client router will boot. Note the name of this software image so that it can be verified after the first client boot (the filename gs3-bfx.90.1 will be used in this example).

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: The type of software (bfx, and so forth) residing in the Flash memory *must* be of the same type as the ROM software installed on the client router. For example, if the client router has gs3-bfx.90.1 (capable of X.25 bridging) in ROM and gs3-bf.90.1 is booted from the Flash server, the client router will operate under the control of the gs3-bf.90.1 software and will not have X.25 bridging capability.

Once you have verified the presence of a bootable image in Flash memory, you can configure the Flash server.

Configuring the Flash Server

Configure the Flash server by adding both the **tftp-server system** command and the **access-list** command to the configuration memory. Use the **configure terminal** command to do so.

Sample output of these commands follows:

```
Server# configure terminal
Enter configuration commands, one per line.
Edit with DELETE, CRTL/W, and CRTL/U;end with CTRL/Z
tftp-server system gs3-bfx.90.1 1
access-list 1 permit 131.131.101.0 0.0.0.255
^Z
Server# write memory <Return>
```

```
[ok]
```

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 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 filenamed gs3-bfx.90.1. For this exercise, the IP address of the Flash server is 131.131.111.111.

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 router you will use as the client should first be saved (uploaded) to a TFTP file server. Refer to the chapter "FIrst-Time Startup and Basic Configuration" for instructions on uploading and downloading system configuration files.

Configuring the Client Router

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 configuration memory. Using these commands requires changing the jumper on the processor's configuration register to the pattern 0-0-1-0 (Position 1).

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 gs3-bfx.90.1 131.131.111.111
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 gs3-bfx.90.1 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.

Caution: The system software (gs3-bfx.90.1) 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:

```
Client# show version
GS Software (GS3-BFX), Version 9.0(1), CISCO SYSTEMS SOFTWARE
Copyright (c) 1986-1992 by cisco Systems, Inc.
Compiled Mon 30-Mar-92 17:16

System Bootstrap, Version 4.5(0.5), CISCO SYSTEMS SOFTWARE

Client uptime is 5 minutes
System restarted by reload
System image file is "gs3-bfx.90.1", booted via tftp from
131.131.111.111
```

The important information in this example is contained in the first line (GS Software...), and the last full line (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 above 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.

Additional TFTP Information

In the event of a Flash memory load failure, the following error message is displayed:

```
Error programming flash memory
```

If you see this message, contact customer service immediately and inform them of the situation.

Note: Images that run from ROM, which includes IGS images and images for the CSC-2 processor, cannot be loaded over the network.

Additionally, you can perform the following steps, which might recover from the error. These steps can be repeated.

Step 1: If possible, inspect the cable connections between the Flash card and the host card. Also, make sure that the host card is securely seated in its Multibus slot.

Step 2: Erase the Flash memory and try to download the file again. To erase the Flash memory, press Return or type y at the `Erase flash before writing? [confirm]` prompt.

Fault-Tolerant 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 a more fault-tolerant alternative in the netbooting environment. Use the following three commands in your configuration to allow you to boot first from Flash, then from a system file, and finally from ROM.

boot system flash *filename*

boot system *filename address*

boot system rom

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

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

In addition to these commands, the router must be configured to boot from ROM (involves a jumper setting change).

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 affects of a failure with the TFTP file server, and the network in general.

Establishing Passwords and System Security

This section describes how to configure password protection and terminal access security.

You may set passwords to control access to the privileged command level and to individual lines. The Terminal Access Controller Access Control System (TACACS) protocol controls terminal use by means of a user-ID-and-password pair. The Defense Data Network developed TACACS to control access to its TAC terminal servers; Cisco patterned its TACACS support after the DDN application.

These system security measures may not provide the level of protection needed for some environments; individual routing protocols and bridging support may have additional security procedures. You may also need to use access lists for additional protection. For procedures for configuring access lists, refer to the sections describing configuration of a particular routing protocol or bridging support.

Establishing the Privileged-Level Password

To assign a password for the privileged command level, use the **enable password** global configuration command:

enable password *password*

The argument *password* is case sensitive and specifies the password prompted for in response to the EXEC command **enable**. The *password* argument may contain any alphanumeric characters, including spaces, up to 80 characters. Password checking is also case sensitive. The password *Secret* is different than the password *secret*, for example, and the password *two words* is an acceptable password.

To enter the privileged command level, type the following EXEC command and then press Return:

enable

Next, type the password for the privileged command level at the `Password:` prompt.

When you use the **enable** command at the console terminal, the EXEC does not prompt you for a password if the privileged mode password is not set. Additionally, if the enable password is not set and the line 0 (console line) password is not set, it is only possible to enter privileged mode on the console terminal. This feature allows you to use physical security rather than passwords to protect privileged mode if that is what you prefer to do.

If the enable password is not set and the line 0 (console) password is set, it is possible to enter privileged command mode either without entering a password at the console terminal or by entering the console line password when prompted while using any other line.

Example:

The following example sets the password *secretword* for the privileged command level on all lines, including the console:

```
enable password secretword
```

Specifying a Password

When an EXEC is started on a line with password protection, the EXEC prompts for the password. If you enter the correct password, the EXEC prints its normal nonprivileged prompt. You may try three times to enter a password before the EXEC exits and returns the terminal to the idle state.

To specify a password, use the **password** line subcommand. The full command syntax follows:

password *text*
no password

The *text* argument may contain any alphanumeric character, including spaces, up to 80 characters. The password checking is also case sensitive. The password *Secret* is different than the password *secret*, for example, and the password *two words* is an acceptable password.

To enable checking for the password specified by the **password** command, use the line subcommand **login**:

login

Alternatively, to use the TACACS user ID and password-checking mechanism instead, use the following subcommand:

login tacacs

To disable all password checking, use the command:

no login

The server prints the message-of-the-day banner before prompting for a password, so you immediately see messages such as no trespassing notifications. By default, virtual terminals require a password. If you do not set a password for a virtual terminal, it will respond to attempted connections by displaying an error message and closing the connection. Use the **no login** subcommand to disable this behavior and allow connections without a password.

Example:

The following example sets the password *letmein* on line 5:

```
line 5
password letmein
login
```

Recovering from a Lost Password

If your network server has the nonvolatile memory option, you can lock yourself out if you enable password checking on the console terminal line and then forget the line password.

To recover from this, force the network server into factory diagnostic mode by turning off the network server, inserting a jumper in bit 15 of the processor configuration register, (or bit 7 of the processor configuration register in the IGS or CRM), and turning on the network server. Follow these steps.

Step 1: You will be asked if you want to set the manufacturers' addresses. Respond by typing "yes." You then see the following prompt:

```
TEST-SYSTEM>
```

Step 2: Type the **enable** command to get the privileged prompt, as in:

```
TEST-SYSTEM#enable
```

Step 3: Type the **show configuration** command to review the system configuration and find the password.

Step 4: To resume normal operation, turn off the network server, remove the jumper from bit 15 (or bit 7) of the configuration register, and turn on the network server again.

Step 5: Log in to the network server with the password that was shown in the configuration file.

The processor configuration registers are described in the Cisco hardware installation and reference publications.

When the network server restarts in factory diagnostic mode, it does not read the nonvolatile memory, thus avoiding the command to set a password for the console terminal. Do not change anything in the factory diagnostic mode.

Note: All debugging capabilities are turned on during diagnostic mode.

Establishing Terminal Access Control

Cisco Systems provides unsupported versions of both a standard and an extended TACACS server. The servers run on most UNIX systems and are available via FTP from the host *ftp.cisco.com*. You may use the servers to create UNIX accounting applications that monitor use of a system and user logins.

The configuration commands in the following sections tailor the behavior of the standard TACACS server.

Setting the Server Host Name

The **tacacs-server host** global configuration command specifies a TACACS host. The full syntax of this command follows.

tacacs-server host *name*
no tacacs-server host *name*

The argument *name* is the name or Internet address of the host. You can use multiple **tacacs-server host** subcommands to specify multiple hosts. The server will search for the hosts in the order you specify them. The **no tacacs-server host** global configuration command deletes the specified name or address.

Limiting Login Attempts

The **tacacs-server attempts** global configuration command controls the number of login attempts that may be made on a line set up for TACACS verification.

tacacs-server attempts *count*
no tacacs-server attempts

The argument *count* is the number of attempts. The default is three attempts.

The **no tacacs-server attempts** global configuration command restores the default.

Example:

This command changes the login attempt to just one try:

```
tacacs-server attempts 1
```

Controlling Retries

The **tacacs-server retransmit** global configuration command specifies the number of times the server will search the list of TACACS server hosts before giving up. The server will try all servers, allowing each one to time-out before increasing the retransmit count.

tacacs-server retransmit *retries*
no tacacs-server retransmit

The argument *retries* is the retransmit count. The default is two retries.

The **no tacacs-server retransmit** global configuration command restores the default.

Example:

This command specifies a retransmit counter value of five times:

```
tacacs-server retransmit 5
```

Setting the Timeout Intervals

The **tacacs-server timeout** global configuration command sets the interval the server waits for a server host to reply.

tacacs-server timeout *seconds*
no tacacs-server timeout

The argument *seconds* specifies the number of seconds. The default interval is five seconds. The **no tacacs-server timeout** global configuration command restores the default.

Example:

This command changes the interval timer to ten seconds:

```
tacacs-server timeout 10
```

Setting the Last Resort Login Feature

If, when running the TACACS server, the TACACS server does not respond, the default action is to deny the request. Use the **tacacs-server last-resort** global configuration command to change that default.

tacacs-server last-resort {**password**|**succeed**}
no tacacs-server last-resort {**password**|**succeed**}

The command causes the network server to request the privileged password as verification, or forces successful login without further input from the user, depending upon the keyword specified, as follows:

- **password**—Allows the user to access the privileged-level command mode by entering the password set by the **enable** command.
- **succeed**—Allows the user to access the privileged-level command mode without further question.

Note: The last resort login feature can be useful when it is important to be able to ensure that login can occur. An example of such a condition is when a systems administrator needs to login in order to troubleshoot TACACS servers which are down.

The **no tacacs-server last-resort** global configuration command restores the system to the default behavior.

Establishing Privileged-Level TACACS

The following variations of the **enable** command may be used to configure privileged-level command access using the TACACS protocol.

Enabling the Privileged Mode

The **enable use-tacacs** global configuration command is used for setting the TACACS protocol for determining whether a user can access the privileged command level.

enable use-tacacs
no enable use-tacacs

If you use this command, the EXEC **enable** command will ask for both a new user name and password. This is then passed to the TACACS server for authentication. If you are using the extended TACACS, it will also pass any existing UNIX user identification code to the server.

Note: When used without extended TACACS, this command allows anyone with a valid user name and password to access the privileged command level, creating a potential security problem. This is because the TACACS query resulting from entering the **enable** command is indistinguishable from an attempt to log in without extended TACACS.

Enabling the Privileged Mode Last Resort Login Feature

The **enable last-resort** global configuration command allows you to specify what happens if the TACACS servers used by the **enable** command do not respond.

enable last-resort {**password**|**succeed**}
no enable last-resort {**password**|**succeed**}

The default action is to fail. Use of the keyword changes the action, as follows:

- **password**—Allows you to enable by entering the privileged command level password.
- **succeed**—Allows you to enable without further question.

The **no enable last-resort** global configuration command restores the default.

Configuring TACACS Accounting

What follows are the configuration commands that tailor the behavior of the extended TACACS client.

Enabling Extended TACACS Mode

The **tacacs-server extended** global configuration command enables an extended TACACS mode.

tacacs-server extended
no tacacs-server extended

This mode provides information about the terminal requests for use in setting up host auditing trails and accounting files for tracking use of terminal servers and routers. Information includes responses from terminal servers and routers, and validation of user requests. An unsupported, extended TACACS server is available from Cisco Systems via anonymous FTP for UNIX users who want to create the auditing programs.

The **no tacacs-server extended** command disables this mode.

Login Notification

The **tacacs-server notify** global configuration command causes a message to be transmitted to the TACACS server, with retransmission being performed by a background process for up to five minutes. The terminal user, however, receives an immediate response allowing access to the terminal. The full syntax of this command follows.

tacacs-server notify {**connect**|**slip**|**enable**|**logout**}
no tacacs-server notify {**connect**|**slip**|**enable**|**logout**}

The keywords specify notification of the TACACS server whenever a user does one of the following:

- **connect**—User makes TCP connections.
- **slip**—User turns SLIP on or off.
- **enable**–User enters the **enable** command.
- **logout**—User logs out.

The **no tacacs-server notify** command with the appropriate keyword disables notification.

Note: When used with extended TACACS, the command **tacacs-server notify enable** allows anyone with a valid user name and password to access the privileged-level command mode.

Login Authentication

The **tacacs-server authenticate** command requires a response from the network or communications server to indicate whether the user may perform the indicated action.

tacacs-server authenticate {**connect**|**slip**|**enable**}
no tacacs-server authenticate {**connect**|**slip**|**enable**}

Actions that require a response include the following, specified as optional keywords:

- **connect**—User makes TCP connections.
- **slip**—User turns SLIP on or off.
- **enable**–User enters the **enable** command.

The **no tacacs-server authenticate** command with the appropriate keyword disables the action.

Optional Password Verification

You can specify that the first TACACS request to a TACACS server is made *without* password verification. This option is configured with the **tacacs-server optional-passwords** global configuration command:

tacacs-server optional-passwords

When the user types in the login name, the login request is transmitted with the name and a zero-length password. If accepted, the login procedure completes. If the TACACS server refuses this request, the terminal server prompts for a password, and tries again when the user supplies a password. The TACACS server must support authentication for users without passwords to make use of this feature. This feature supports all TACACS requests—login, SLIP, enable, and so on.

Authenticating User Names

Networks that cannot support a TACACS service may still wish to use a user name-based authentication system. In addition, it may be useful to define special user names that get special treatment (for example, an "info" user name that does not require a password, but connects the user to a general purpose information service).

The network server software supports these needs by implementing a local **username** configuration command. The format for the command is:

username *name* [**nopassword**|**password** *encryptiontype password*]
username *name* [**accesslist** *number*]
username *name* [**autocommand** *command*]
username *name* [**noescape**] [**nohangup**]

Multiple **username** commands can be used to specify options for a single user.

The **nopassword** keyword means that no password is required for this user to log in. This is usually most useful in combination with the **autocommand** keyword.

The **password** keyword specifies a possibly encrypted password for this user name.

The *encryptiontype* argument is a single-digit number that defines whether the text immediately following is encrypted and, if so, what type of encryption is used. Currently defined encryption types are 0, which means that the text immediately following is not encrypted, and 7, which means that the text is encrypted using a Cisco-defined encryption algorithm. A *password* can contain embedded spaces and must be the last option specified in the **username** command.

When you specify an encryption type of 0 to enter an unencrypted password, the system displays the encrypted version of the password. For example, suppose you enter the following command:

```
username bill password westward
```

The system would display this command like this:

```
username bill password 7 21398211
```

The encrypted version of the password is 21398211. The password was encrypted by the Cisco-defined encryption algorithm, as indicated by the "7."

If you were to enter the following command, the system would assume that the password is already encrypted and would do no encryption. It would display the command exactly as you typed it:

```
username bill password 7 21398211
username bill password 7 21398211
```

The **accesslist** keyword specifies an outgoing access list that overrides the access list specified in the **access class** line configuration subcommand. It is used for the duration of the user's session. The access list number is specified by the *number* argument.

The **autocommand** keyword causes the command specified by the *command* argument to be issued automatically after the user logs in. When the command is complete, the session is terminated. As the command can be any length and contain imbedded spaces, commands using the **autocommand** keyword must be the last option on the line.

The **noescape** keyword prevents a user from using an escape character on the host to which he is connected.

The **nohangup** keyword prevents the network server from disconnecting the user after an automatic command (set up with the **autocommand** keyword) has completed. Instead, the user gets another login prompt.

Examples:

To implement a service similar to the UNIX **who** command, which can be given at the login prompt and lists the current users of the network server, the command takes the following form:

```
username who nopassword nohangup autocommand show users
```

To implement an ID that will work even if all the TACACS servers break, the command is as follows:

```
username superuser password superpassword
```

Configuring the Simple Network Management Protocol (SNMP)

The Simple Network Management Protocol (SNMP) provides a way to access and set configuration and run-time parameters for the network server. Cisco System's implementation of SNMP is compatible with RFCs 1155, 1156, and 1157. The Cisco Management Information Base (MIB) supports RFCs 1155 to 1213, and provides Cisco-specific variables.

A separate document, available in RFC 1213-type (MIB II) format, describes all the Cisco-specific SNMP variables in the Cisco portion of the MIB. It also describes what is required to establish minimum configuration. Contact Cisco Systems to obtain a copy of this document, which includes instructions for accessing the variables using SNMP.

Cisco also provides support for some of the FDDI MIB variables as described in RFC 1285, "FDDI Management Information Base," January 1992 by Jeffrey D. Case of the University of Tennessee and SNMP Research, Inc. One such variable that Cisco supports is `snmpFddiSMTCFState`.

Enabling and Disabling the SNMP Server

To be able to configure the SNMP server, you need to be in the configuration command collection mode. You enter this mode using the EXEC command **configure** at the EXEC prompt. See the section "Entering Configuration Mode" in the chapter "First-Time Startup and Basic Configuration" for a description of the procedure.

You begin SNMP operation by entering the configuration commands that define the desired operation. To disable SNMP server operations on the network server after it has been started, use the **no snmp-server** global configuration command:

no snmp-server

Defining the SNMP Server Access List

To set up an access list that determines which hosts can send requests to the network server, use the **snmp-server access-list** global configuration command. This access list applies only to the global read-only SNMP agent configured with the command **snmp-server community**. The network server ignores packets from hosts that the access list denies.

The full command syntax follows.

snmp-server access-list *list*
no snmp-server access-list *list*

The argument *list* is an integer from 1 through 99 that specifies an IP access list number.

The **no snmp-server access-list** global configuration command removes the specified access list.

Example:

This command causes the router to ignore packets from hosts that do not match access list 21:

```
snmp-server access-list 21
```

Setting the Community String

To set up the community access string, use the **snmp-server community** global configuration command. The full command syntax follows:

snmp-server community *string* [**RO**|**RW**] [*list*]
no snmp-server community [*string*] [**RO**|**RW**] [*list*]

This command enables SNMP server operation on the network server. The argument *string* specifies a community string that acts like a password and permits access to the SNMP protocol.

By default, an SNMP community string permits read-only access (keyword **RO**); use the keyword **RW** to allow read-write access. The optional argument *list* is an integer from 1 through 99 that specifies an access list of Internet addresses that may use the community string.

The **no snmp-server community** global configuration command removes the specified community string or access list.

Example:

This command assigns the string *comaccess* to the SNMP server, allows read-only access, and specifies that addresses that match the criteria in access list 4 may use the community string. (Notice that the string is entered *without* quotes or any other parsing characters.)

```
snmp-server community comaccess RO 4
```

Note: Neither the console password nor the enabled-mode password can be used as the community string.

Setting the System Contact String

To set the system contact string (syscontact), use the **snmp-server contact** command. The command syntax follows:

snmp-server contact *text*

The *text* argument is a string that specifies the system contact information.

Setting the System Location String

To set the system location string, use the **snmp-server location** command. The command syntax follows:

snmp-server location *text*

The *text* argument is a string that specifies the system location information.

Establishing the Message Queue Length

To establish the message queue length for each TRAP host, use the **snmp-server queue-length** global configuration command:

snmp-server queue-length *length*
no snmp-server queue-length

This command defines the length of the message queue for each TRAP host.

The argument *length* is the number of TRAP events that can be held before the queue must be emptied; the default is 10. Once a TRAP message is successfully transmitted, software will continue to empty the queue, but never faster than at a rate of four TRAP messages per second.

The **no snmp-server queue-length** command resets the queue length to its default value of 10.

Example:

This command establishes a message queue that traps four events before it must be emptied:

```
snmp-server queue-length 4
```

Establishing Packet Filtering

To establish the packet filtering size, use the **snmp-server packetsize** global configuration command. The full command syntax follows:

snmp-server packetsize *bytes*
no snmp-server packetsize

This command allows control over the largest SNMP packet size permitted when the SNMP server is receiving a request or generating a reply.

The argument *bytes* is a byte count from 484 through 8192. The default is 484. The **no snmp-server packetsize** command resets this default.

Example:

This command establishes a packet filtering maximum size of 1024 bytes:

```
snmp-server packetsize 1024
```

Establishing the TRAP Message Recipient

To specify the recipients of TRAP messages, use the **snmp-server host** global configuration command. The full syntax follows:

snmp-server host *address community-string* [**snmp**|**tty**]
no snmp-server host *address community-string*

This command specifies which host or hosts should receive TRAP messages. You need to issue the **snmp-server host** command once for each host acting as a TRAP recipient.

The argument *address* is the name or Internet address of the host. The argument *community-string* is the password-like community string set with the **snmp-server community** command.

The optional keywords define whether the TRAPS be included, as follows:

- **snmp**—Causes all SNMP-type TRAP messages to be sent and starts the Cisco-specific RELOAD TRAP message.
- **tty**—Causes TCP connection TRAP messages to be included.

With the **snmp-server host** command, you get all the SNMP TRAP messages about TTY events by default. The **no snmp-server host** command removes the specified host.

Examples:

This command sends all possible SNMP TRAPS to 131.108.2.160, including TTY TRAPS.

```
snmp-server host 131.108.2.160
```

In order to turn these TRAP messages off, use the **no snmp-server host** command. For example, the following sequence of commands only sends the seven SNMP TRAP messages to 131.108.2.160, not all the others.

```
snmp-server host 131.108.2.160
no snmp-server host 131.108.2.160 tty
```

This example causes all the SNMP-type messages to be sent to the host specified by the name *cisco.com*. The command uses the community string *comaccess* as the password:

```
snmp-server host cisco.com comaccess snmp
```

Establishing TRAP Message Authentication

To establish the TRAP message authentication, use the **snmp-server trap-authentication** global configuration command:

snmp-server trap-authentication
no snmp-server trap-authentication

This command enables the network server to send a TRAP message when it receives a packet with an incorrect community string.

The SNMP specification requires that a TRAP message be generated for each packet with an incorrect community string. However, because this action can result in a security breach, the network server by default does not return a TRAP message when it receives an incorrect community string.

Establishing the TRAP Message Timeout

To define how often to try resending TRAP messages on the retransmission queue, use these global configuration commands:

snmp-server trap-timeout *seconds*
no snmp-server trap-timeout

The argument *seconds* sets the interval for resending the messages. The default is set to 30 seconds. The **no snmp-server trap-timeout** command restores this default.

Example:

This command sets an interval of 20 seconds to try resending TRAP messages on the retransmission queue:

```
snmp-server trap-timeout 20
```

Enabling SNMP System Shutdown Feature

Using SNMP packets, a network management tool can send messages to users on virtual terminals and the network server's console. This facility operates in a similar fashion to the SNMP **send** command; however, the SNMP request that causes the message to be issued to the users, also specifies the action to be taken after the message is delivered. One possible action is a shutdown request.

Requesting a *shutdown-after-message* is similar to issuing a **send** command followed by a **reload** command. Because the ability to cause a reload from the network is a powerful feature, it is protected by this configuration command. To use this SNMP message reload feature the device configuration must include the **snmp-server system-shutdown** global configuration command. The full command syntax follows:

snmp-server system-shutdown
no snmp-server system-shutdown

The **no snmp-server system-shutdown** option prevents a SNMP system-shutdown request (from an SNMP manager) from resetting the Cisco agent.

To understand how to use this command with SNMP requests, read the document *mib.txt* available by anonymous FTP from *ftp.cisco.com*. This document is available in RFC 1213-type format. It describes all the Cisco-specific SNMP variables in the Cisco portion of the MIB. It also describes what is required to establish minimum configuration. Contact Cisco Systems to obtain a copy of this document, which includes instructions for accessing the variables using SNMP.

Configuring the Trivial File Transfer Protocol (TFTP) Server

You can configure the network server to act as a limited Trivial File Transfer Protocol (TFTP) server from which other Cisco servers can boot their software. As a TFTP server host, the network server responds to TFTP read request messages by sending a copy of its ROM software to the requesting host. The TFTP read request message must use the file name that you specified in the network server configuration.

To specify TFTP server operation for a communications server, use the **tftp-server system** global configuration command. The full syntax follows.

tftp-server system *filename ip-access-list*
no tftp-server system *filename ip-access-list*

This command has two arguments: *filename* and *list*. The argument *filename* is the name you give the communications server ROM file, and the argument *ip-access-list* is an IP access-list number.

The system sends a copy of the ROM software to any host which issues a TFTP read request with this file name. To learn how to specify an access list, see the "Configuring IP Access Lists" section in the chapter "Routing IP."

You can specify multiple file names by repeating the **tftp-server system** command. To remove a previously defined file name, use the **no tftp-server system** command and append the appropriate file name and an access-list number.

Example:

This command causes the router to send, via TFTP, a copy of the ROM software when it receives a TFTP read request for the file *goodimage*. The requesting host is checked against access list 22.

```
tftp-server system goodimage 22
```

Tailoring Use of Network Services

The **service** global configuration command tailors use by the network server of network-based services. Some **service** commands also configure system defaults; see **decimal-tty** for an example. The full command syntax follows:

service *keyword*
no service *keyword*

The argument *keyword* is one of the following:

- **config**—Specifies TFTP autoloading of configuration files; disabled by default on system with nonvolatile memory.

- **decimal-tty**—Specifies that line numbers be displayed and interpreted as decimal numbers rather than octal numbers; disabled by default.
- **finger**—Allows Finger protocol requests (defined in RFC 742) to be made of the network server; enabled by default. This service is equivalent to issuing a remote **show users** command.
- **tcp-keepalives-**{**in**|**out**}—Generates keepalive packets on idle network connections. The **in** keyword generates them on incoming connections (initiated by remote host); the **out** keyword generates them on outgoing connections (initiated by a user). There is a column in the EXEC **show tcp** display showing the keepalive statistics. The `wakeups` row shows how many keepalives have been transmitted without receiving any response (this is reset to 0 when a response is received).

The **no service** command disables the specified service or function.

Example:

The following command enables TFTP autoloading of configuration files:

```
service config
```

Redirecting System Error Messages

By default, the network server sends the output from the EXEC command **debug** and system error messages to the console terminal.

To redirect these messages, as well as output from asynchronous events such as interface transition, to other destinations, use the **logging** configuration command options.

These destinations include the console terminal, virtual terminals, and UNIX hosts running a syslog server; the syslog format is compatible with 4.3 BSD UNIX.

To configure the logging of messages, you need to be in the configuration command collection mode. To enter this mode, use the EXEC command **configure** at the EXEC prompt (see the section "Entering Configuration Mode" in the chapter "First-Time Startup and Basic Configuration" for the procedure). The following sections describe how to implement these redirection options.

Enabling Message Logging

To enable or disable message logging, use the following global configuration commands:

logging on
no logging on

The **logging on** command enables message logging to all supported destinations other than the console. This behavior is the default.

The **no logging on** command enables logging to the console terminal only.

Logging Messages to an Internal Buffer

The default logging device is the console; all messages are displayed on the console unless otherwise specified.

To log messages to an internal buffer, use the logging buffered global configuration command. The full command syntax follows.

logging buffered
no logging buffered

The **logging buffered** command copies logging messages to an internal buffer instead of writing them to the console terminal. The buffer is circular in nature, so newer messages overwrite older messages. To display the messages that are logged in the buffer, use the EXEC command **show logging**. The first message displayed is the oldest message in the buffer.

The **no logging buffered** command cancels the use of the buffer and writes messages to the console terminal, which is the default.

Logging Messages to the Console

To limit how many messages are logged to the console, use the **logging console** global configuration command. The full syntax of this command follows:

logging console *level*
no logging console

The **logging console** command limits the logging messages displayed on the console terminal to messages with a level number at or below the specified severity level, which is specified by the *level* argument.

The argument *level* can be one of the keywords listed in Table 1-1. They are listed in order from the most severe to the least severe level.

Table 1-1 Logging Message Keywords and Levels

Level	Keyword	Description	Syslog Definition
0	**emergencies**	System is unusable	LOG_EMERG
1	**alerts**	Immediate action is needed	LOG_ALERT
2	**critical**	Critical conditions exist	LOG_CRIT
3	**errors**	Error conditions exist	LOG_ERR
4	**warnings**	Warning conditions exist	LOG_WARNING
5	**notification**	Normal, but significant, conditions exist	LOG_NOTICE
6	**informational**	Informational messages	LOG_INFO
7	**debugging**	Debugging messages	LOG_DEBUG

The default is to log messages at the **warnings** level to the console.

The **no logging console** command disables logging to the console terminal.

Example:

This command sets console logging of messages at the debug level:

```
logging console debug
```

Logging Messages to Another Monitor

To limit the level of messages to log to the terminal lines (monitors), use the **logging monitor** command. The full syntax of this command follows.

logging monitor *level*
no logging monitor

The **logging monitor** command limits the logging messages displayed on terminal lines other than the console line to messages with a level at or above *level*. The argument *level* is one of the keywords described for the **logging console** command in the previous section, "Logging Messages to the Console." To display logging messages on a terminal, use the privileged EXEC command **terminal monitor.**

The **no logging monitor** command disables logging to terminal lines other than the console line.

This command sets the level of messages displayed on monitors other than the console to notifications:

```
logging monitor notifications
```

Logging Messages to a UNIX Syslog Server

To log messages to the syslog server host, use the **logging** global configuration command. The full syntax is as follows:

logging *internet-address*
no logging *internet-address*

The **logging** command identifies a syslog server host to receive logging messages. The argument *internet-address* is the Internet address of the host. By issuing this command more than once, you build a list of syslog servers that receive logging messages.

The **no logging** command deletes the syslog server with the specified address from the list of syslogs.

Limiting Messages to a Syslog Server

To limit how many messages are sent to the syslog servers, use the **logging trap** global configuration command. Its full syntax follows:

logging trap *level*
no logging trap

The **logging trap** command limits the logging messages sent to syslog servers to messages with a level at or above *level*. The argument *level* is one of the keywords described for the **logging console** command in Table 1-1.

To send logging messages to a syslog server, specify its host address with the **logging** command.

The default trap level is **informational.**

The **no logging trap** command disables logging to syslog servers.

The current software generates four categories of the syslog messages:

- Error messages about software or hardware malfunctions, displayed at the **errors** level.
- Output from the **debug** commands, displayed at the **debugging** level.
- Interface up/down transitions and system restart messages, displayed at the **notifications** level.
- Reload requests and low-process stack messages, displayed at the **informational** level.

The EXEC command **show logging** displays the addresses and levels associated with the current logging setup. The command output also includes ancillary statistics.

Example:

To set up the syslog daemon on a 4.3 BSD UNIX system, include a line such as the following in the file */etc/syslog.conf*:

```
local7.debug /usr/adm/logs/tiplog
```

The `local7` keyword specifies the logging facility to be used.

The `debug` argument specifies the syslog level. See the previous *level* arguments list for other arguments that can be listed.

The UNIX system sends messages at or below this level to the file specified in the next field. The file must already exist, and the syslog daemon must have permission to write to it.

Configuring Console and Virtual Terminal Lines

To configure your console and virtual terminal lines you need to be in the configuration command collection mode. To enter this mode, use the EXEC command **configure** at the EXEC prompt (see the section “Entering Configuration Mode” in the chapter “First-Time Startup and Basic Configuration” for the procedure).

Starting Line Configuration

To start configuring a terminal line, use the **line** command. This command identifies a specific line for configuration and starts line configuration command collection.

The **line** command has the following syntax:

line [*type-keyword*] *first-line* [*last-line*]

This command can take up to three arguments: a keyword, a line number, or a range of lines numbers.

The optional argument *type-keyword* specifies the type of line to be configured; it is one of the following keywords:

- **console**—Console terminal line
- **aux**—Auxiliary line (described in the following section)
- **vty**—Virtual terminal for remote console access

When the line type is specified, the argument *first-line* is the relative number of the terminal line (or the first line in a contiguous group) you want to configure. Numbering begins with zero.

The optional argument *last-line* then is the relative number of the last line in a contiguous group you want to configure.

If you omit *type*, then *first-line* and *last-line* are absolute rather than relative line numbers. To display absolute line numbers, use the EXEC command **show users all.**

The network server displays an error message if you do not specify a line number.

Note: Line numbers, by default, are octal on the network servers.

The **line** command enables you to easily configure a large group of lines all at once. After you set the defaults for the group, you can use additional **line** commands and subcommands to set special characteristics, such as location, for individual terminal lines.

Example:

The following command starts configuration for the first five virtual terminal lines:

```
line vty 0 4
```

Configuring the CPU Auxiliary Port

The **line** command keyword **aux** allows use of an auxiliary RS-232 DTE port available on all processor cards. Use this port to attach to an RS-232 port of a CSU/DSU, protocol analyzer, or modem. You can monitor that port remotely by connecting to the TCP port whose number is 2000 decimal plus the line number of the auxiliary port. For example, if the auxiliary port was line 1 (obtained from the EXEC command **show users all**), then the TCP port would be 2001. You must order a special cable from Cisco Systems for use with this auxiliary port.

To configure the auxiliary port, use this variation of the **line** command:

line aux *port-address*

When configuring the auxiliary port, address it as line 0, as in this sample:

```
line aux 0
```

The auxiliary port asserts DTR when a Telnet connection is established or when an EXEC becomes active. Flow control is not supported on either the auxiliary port or the console port.

By default, the auxiliary port supports an EXEC process. This default can be re-enabled using the **exec line** subcommand.

These commands configure the auxiliary port with a line speed of 2400 baud, and enable the EXEC.

```
line aux 0
exec
speed 2400
```

No modem control signals are supported on this line. If an auto-answer modem is configured on the line, you must dial up, log in, then hang up. The DTR signal will be active whenever an EXEC is configured on the auxiliary port.

Note: The EXEC is still present and may be used by the next person that dials into the number. This could cause security problems.

Disabling Automatic Connections

To disable automatic connections, use the following line subcommand:

transport preferred none

This command prevents the system from assuming that any unrecognized command is a host name. No connection will be attempted if the command is not recognized (by misspelling a command, for example).

Establishing Line Passwords

When you start an EXEC on a line with password protection, the EXEC prompts for the password. If you enters the correct password, the EXEC prints its normal prompt. You may try three times to enter a password before the EXEC exits and returns the terminal to the idle state.

To specify a password, use the **password** line subcommand. Its full syntax follows.

password *text*
no password

The *text* argument may contain any alphanumeric characters, including spaces, up to 80 characters. The password checking is case sensitive. The password *Secret* is different than the password *secret*, for example, and the password *two words* is an acceptable password.

Example:

This command sets the words "Big Easy" as the password on line 1:

```
line 1
password Big Easy
```

Note: Many UNIX systems require a tab character to be used as the "white space" separator in the /etc/syslog.conf file. Use of a space character rather than a tab may cause the entry in /etc/syslog.conf to be ignored.

Establishing Connection Restrictions

To establish connection restrictions on the lines to some Internet addresses, use the **access-class** line subcommand. The full command syntax follows.

access-class *list* {**in**|**out**}
no access-class *list* {**in**|**out**}

The **access-class** subcommand restricts connections on a line or group of lines to certain Internet addresses. The argument *list* is an integer from 1 through 99 that identifies a specific access list of Internet addresses. The keyword **in** applies to incoming connections, such as virtual terminals. The keyword **out** applies to outgoing Telnet connections. The **no access-class** command removes access restrictions on the line for the specified connections.

Example:

This example subcommand sets restrictions on access list 1 for outgoing Telnet connections:

```
access-class 1 out
```

See the section "Configuring IP Access Lists" in the chapter "Routing IP" for information about configuring access lists.

Suppressing Banner Messages

By default, messages defined by the **banner motd** and **banner exec** commands are always displayed. This condition is defined by the **exec-banner** line subcommand. Its full syntax follows:

exec-banner
no exec-banner

To suppress display of a banner, enter the **no exec-banner** command.

Example:

These commands suppresses the banner on virtual terminal lines 0 through 4:

```
line vty 0 4
no exec-banner
```

Turning On and Off the Vacant Banner

The router will display a message on the console when there is no active EXEC. This message, called the vacant message, is different from the banner message displayed when an EXEC process is activated.

To turn the vacant message banner on or off, use the **vacant-message** line configuration subcommands. The **vacant-message** command enables the banner to be displayed on the screen of an idle terminal. The full syntax of this command follows.

vacant-message [*c message c*]
no vacant-message

The **vacant-message** subcommand without any arguments causes the default message to be displayed. If you desire a banner, follow **vacant-message** with one or more blank spaces and a delimiting character *(c*) you choose. Then type one or more lines of text (*message*), terminating the text with the second occurrence of the delimiting character.

The **no vacant-message** line configuration subcommand suppresses a banner message.

Example:

This example will turn on the system banner and display a message.

```
line 0
vacant-message #
                Welcome to Cisco Systems, Inc.
        This is the console terminal of the router Dross.
#
```

Note: You cannot use the delimiting character in the banner message.

Setting the Escape Character

The **escape-character** line subcommand defines the escape character. The following illustrates the full syntax of this command:

escape-character *decimal-number*
no escape-character

The argument *decimal-number* is the ASCII decimal representation of the desired escape character or an escape character (Ctrl-P, for example). Typing the escape character followed by the X key returns you to the EXEC when you are connected to another computer. The default escape character is Ctrl-^. (See the appendix "ASCII Character Set" for a list of ASCII characters.)

The operating software interprets Break on the console as an attempt to halt the system.

Note: Depending upon the configuration register setting, console breaks will either be ignored or cause the server to shut down. The Break key *cannot* be used as the escape character on the Cisco router.

The **no escape-character** line configuration subcommand reinstates the default escape character.

Example:

This command changes the escape characters to Ctrl-P (ASCII character 17):

```
line 5
escape-character 17
```

Setting the Terminal Location

To set the location of the terminal, use the **location** line subcommand. The full syntax of this command follows:

location *text*
no location

This subcommand is for informational purposes only; it is not used by any aspects of the system software. The argument *text* is the desired description. The description appears in the output of the EXEC command **systat.** A maximum of 80 characters can be entered.

The **no location** subcommand removes the information.

Example:

This command describes the location of the terminal on line 2 as being Andrea's terminal:

```
line 2
```

```
location Andrea's terminal
```

Setting the EXEC Timeout Intervals

The EXEC command interpreter waits for a specified interval of time until the user starts input. If no input is detected, the EXEC resumes the current connection. If no connections exist, the EXEC returns the terminal to the idle state and disconnects the incoming session.

To set this interval, use the **exec-timeout** line configuration subcommand. The full syntax of the command follows.

exec-timeout *minutes* [*seconds*]
no exec-timeout

The argument *minutes* is the number of minutes, and the optional argument *seconds* specifies additional interval time in seconds. The default interval is ten minutes; an interval of zero specifies no time-outs.

The **no exec-timeout** subcommand removes the time-out definition. It is the same as entering **exec-timeout 0**.

Examples:

This command sets an interval of 2 minutes, 30 seconds.

```
exec-timeout 2 30
```

This command sets an interval of 10 seconds:

```
exec-timeout 0 10
```

Setting the Screen Length

To set the terminal screen length, use the **length** line configuration subcommand. The full syntax of the command follows.

length *screen-length*
no length

The argument *screen-length* is the number of lines on the screen. The network server uses this value to determine when to pause during multiple-screen output. The default length is 24 lines. A value of zero disables pausing between screens of output.

The **no length** command is the same as entering the command **length 0**.

Note: Not all commands pay attention to the configured screen length. For example, the **show terminal** command assumes a screen length of 24 lines or more.

Setting Notification

To enable the terminal to notify the user about pending output, use the **notify** line subcommand. The full syntax of the command follows.

notify
no notify

The **notify** subcommand sets a line to inform a user who has multiple, concurrent Telnet connections when output is pending on a connection other than the current connection.

The **no notify** line configuration subcommand ends notification and is the default.

Setting Character Padding

To set the padding on characters, use the **padding** line configuration subcommand. The full syntax of the command follows.

padding *decimal-number count*
no padding *decimal-number*

The **padding** subcommand sets padding for a specified output character. The argument *decimal-number* is the ASCII decimal representation of the character, and the argument *count* is the number of NUL bytes sent after that character.

The **no padding** line configuration subcommand removes padding for the specified output character.

The following command pads Return (ASCII character 13) with 25 NUL bytes:

```
padding 13 25
```

Global System Configuration Command Summary

This section lists all of the global system configuration commands in alphabetical order.

[**no**] **banner** {**motd**|**exec**|**incoming**} *c text c*

Displays the message that the EXEC command interpreter displays whenever a user starts any EXEC process or activates a line. The **motd**, **exec**, and **incoming** keywords control when the banner message is displayed. The argument *c* specifies a delimiting character of your choice. The argument *text* specifies the message to be shown on the screen whenever an interface line is activated.

[**no**] **boot buffersize** *bytes*

Specifies the size of the buffer to be used for netbooting a host or a network configuration file. The argument *bytes* by default is the size of your nonvolatile memory, or 32 kilobytes if you do not have nonvolatile memory. There is no minimum or maximum size that may be specified.

[**no**] **boot host** *filename* [*address*]

Specifies the host configuration file name. The argument *filename* is the new name for the host configuration file. If you omit the argument *address,* the network server uses the default broadcast address of 255.255.255.255. The optional argument *address* allows you to specify a specific network host or a subnet broadcast address.

[**no**] **boot network** *filename* [*address*]

Specifies the network configuration file name. The argument *filename* is the new name for the network configuration file. If you omit the optional argument *address,* the network server uses the default broadcast address of 255.255.255.255. The optional argument *address* allows you to specify a specific network host or a subnet broadcast address.

[**no**] **boot system** *filename* [*address*]

Specifies a second operating software image from a file that is not in nonvolatile memory. The argument *filename* is the file name of the operating software to load, and the optional argument *address* is the address of the network host holding that file.

[**no**] **boot system flash** *filename*

Same as **boot system** *filename* [*address*] but instead the system is booted from the Flash memories. Specify a name to associate with the image from Flash memories with the *filename* parameter.

[**no**] **boot system rom**

Automatically boots the system from the ROM system image. This command is usually used as a backup to other **boot system** commands that specify system images that exist either on the network or in Flash memories.

[**no**] **buffers** {**small**|**middle**|**big**|**large**|**huge**} {**permanent**|**max-free**|**min-free**|**initial**} *number*

Allows a network administrator to adjust initial buffer pool settings and set limits at which temporary buffers are created and destroyed.The first keyword denotes the size of buffers in the pool; the default number of the buffers in a pool is determined by the hardware configuration. The second keyword specifies the buffer management parameter to be changed, as follows:

- **permanent**—The number of permanent buffers that the system tries to allocate.
- **max-free**—The maximum number of free or unallocated buffers in a buffer pool.
- **min-free**—The minimum number of free or unallocated buffers in a buffer pool.
- **initial**—The number of additional temporary buffers which should be allocated when the system is reloaded.

The argument *number* specifies the number of buffers to be allocated.

The **no buffers** command with appropriate keywords and arguments restores the default buffer values.

[**no**] **buffers huge size** *number*

Dynamically resizes all huge buffers to the value that you supply. The buffer size cannot be lowered below the default. The argument *number* specifies the number of buffers to be allocated. The **no** version of the command with the keyword and argument restores the default buffer values.

copy flash tftp

Copies a Flash TFTP image back to a TFTP server.

copy tftp flash

Copies a TFTP image into the current Flash configuration.

enable

Allows you to enter the privileged command level.

enable password *password*

Assigns a password for the privileged command level. The argument *password* is case sensitive and specifies the password prompted for in response to the EXEC command **enable.**

[**no**] **enable last-resort** {**succeed**|**password**}

Allows you to specify what happens if the TACACS servers used by the **enable** command do not respond. The default action is to fail. The keywords change this default action:

- **succeed**—Allows you to enable without further question.
- **password**—Allows you to enable by entering the privileged command level.

The **no** version of the command restores the default.

[**no**] **enable use-tacacs**

Enables or disables use of TACACS to check the user ID and password supplied to the EXEC **enable** command.

hostname *name*

Specifies the name for the network server. The default is *Router*.

[**no**] **logging** *internet-address*

Identifies a syslog server host to receive logging messages. The argument *internet-address* is the Internet address of the host.The **no logging** command deletes the syslog server with the specified address from the list of syslogs.

[**no**] **logging buffered**

Copies logging messages to an internal buffer instead of writing them to the console. The **no** version of the command cancels this behavior and writes messages to the console terminal; this is the default.

[**no**] **logging console** *level*

Limits the logging of messages displayed on the console terminal to messages with a level at or above the specified severity, which is specified by the *level* argument. The argument *level* can be one of the following keywords, listed here in order from the most severe to the least severe level.

- **emergencies**—System unusable
- **alerts**—Immediate action needed
- **critical**—Critical conditions
- **errors**—Error conditions
- **warnings**—Warning conditions (default)
- **notification**—Normal but significant conditions

- **informational**—Informational messages only
- **debugging**—Debugging messages

The **no logging console** command disables logging to the console terminal.

[**no**] **logging monitor** *level*

Limits the logging messages displayed on terminal lines other than the console line to messages with a level at or above *level*. The argument *level* is one of the keywords described for the **logging console** command. The **no logging monitor** command disables logging to terminal lines other than the console line.

[**no**] **logging on**

Enables or disables message logging to all supported destinations except the console. Enabled message logging is the default.

[**no**] **logging trap** *level*

Limits the logging messages sent to syslog servers to messages with a level at or above *level*. The argument *level* is one of the keywords described for the **logging console** command.

[**no**] **service** *keyword*

Tailors use by the network server of network-based services. The argument *keyword* is one of the following:

- **config**—Specifies TFTP autoloading of configuration files; disabled by default on system with nonvolatile memory.
- **decimal-tty**—Specifies that line numbers be displayed and interpreted as decimal numbers rather than octal numbers; disabled by default.
- **finger**—Allows Finger protocol requests (defined in RFC 742) to be made of the network server; enabled by default.
- **tcp-keepalives-{in|out}**—Generates keepalive packets on idle network connections. The **in** keyword generates them on incoming connections; the **out** keyword generates them on outgoing connections.

The **no service** command disables the specified service or function.

no snmp-server

Disables the SNMP operations.

[**no**] **snmp-server access-list** *list*

Sets up an access list that determines which hosts can send requests to the network server. The argument *list* is an integer from 1 through 99 that specifies an IP access list.

[**no**] **snmp-server community** *string* [**RO**|**RW**] [*list*]

Enables or disables SNMP server operation on the network server. The argument *string* specifies a community string that acts like a password and permits access to the SNMP protocol.

[**no**] **snmp-server host** *address community-string* [**snmp**|**tty**]

Specifies which host or hosts should receive TRAP messages. Issue the **snmp-server host** command once for each host acting as a TRAP recipient. The argument *address* is the name or Internet address of the host. The argument *community-string* is the password-like community string set with the **snmp-server community** command. These optional keywords direct the TRAP:

- **snmp**—Causes all SNMP-type TRAP messages to be sent and starts the Cisco-specific RELOAD TRAP message.
- **tty**—Causes TCP connection TRAP messages to be included.

[**no**] **snmp-server packetsize** *bytes*

Sets or removes control over the largest SNMP packet size permitted when the SNMP server is receiving a request or generating a reply. The argument *bytes* is a byte count from 484 through 8192. The default is 484.

[**no**] **snmp-server queue-length** *length*

Defines the length of the message queue for each TRAP host. The argument *length* is the number of TRAP events that can be held before the queue must be emptied; the default is ten. The **no** version of the command resets the queue length to the default.

[**no**] **snmp-server system-shutdown**

Allows or restricts use of the SNMP message reload feature, and prevents an SNMP system-shutdown request from resetting the Cisco agent.

[**no**] **snmp-server trap-authentication**

Allows or restricts the network server from sending a TRAP message when it receives a packet with an incorrect community string.

[**no**] **snmp-server trap-timeout** *seconds*

Defines how often to try resending TRAP messages on the retransmission queue. The argument *seconds* sets the interval for resending the messages. The default is set to 30 seconds. The **no** form of the command restores the default.

[**no**] **tacacs-server attempts** *count*

Controls the number of login attempts that may be made on a line set up for TACACS verification. The argument *count* is the number of attempts. The default is three attempts.

[**no**] **tacacs-server authenticate** {**connect**|**slip**|**enable**}

Specifies that a response is required from the network or communications server to indicate whether the user may perform the indicated action. Actions which require a response include the following:

- **connect**—User makes TCP connections.
- **slip**—User turns SLIP on or off.
- **enable**–User enters the **enable** command.

The **no** form of the command disables the action.

[**no**] **tacacs-server extended**

Enables or disables an extended TACACS mode. This mode provides information about the terminal requests for use in setting up UNIX auditing trails and accounting files for tracking use of terminal servers and routers.

[**no**] **tacacs-server host** *name*

Specifies a TACACS host. The argument *name* is the name or Internet address of the host. You can use multiple commands to specify multiple hosts.

[**no**] **tacacs-server last-resort** {**password**|**succeed**}

Causes the network server to request the privileged password as verification, or forces successful login without further input from the user, depending upon the keyword specified, as follows.

- **password**—Allows the user to access the privileged-level command mode by entering the password set by the **enable** command.
- **succeed**—Allows the user to access the privileged-level command mode without further question.

The **no** form of the command disables the action.

[**no**] **tacacs-server notify** {**connect**|**slip**|**enable**|**logout**}

Causes a message to be transmitted to the TACACS server, with retransmission being performed by a background process for up to five minutes. The keywords specify notification of the TACACS server whenever a user does one of the following:

- **connect**—Makes TCP connections.
- **slip**—Turns SLIP on or off.
- **enable**–Enters the **enable** command.
- **logout**—Logs out.

The **no** form of the command disables the messages.

tacacs-server optional-passwords

Specifies that the first TACACS request to a TACACS server is made *without* password verification. Supports all TACACS requests—login, SLIP, enable, and so on.

[**no**] **tacacs-server retransmit** *retries*

Specifies the number of times the server will search the list of TACACS server hosts before giving up. The argument *retries* is the retransmit count. The default is two retries. The **no** form of the command restores the default.

[**no**] **tacacs-server timeout** *seconds*

Sets the interval the server waits for a server host to reply. The argument *seconds* specifies the number of seconds. The default interval is five seconds. The **no** form of the command restores the default.

[**no**] **tftp-server system** *filename ip-access-list*

Specifies or removes TFTP server operation for a communications server. The argument *filename* is the name given to the network server ROM file, and the argument *access-list* is an IP access list number.

username *name* [**nopassword**|**password** *encryptiontype password*]
username *name* [**accesslist** *number*]
username *name* [**autocommand** *command*]
username *name* [**noescape**] [**nohangup**]

Create special case user name-based authentications. Multiple **username** commands can be used to specify options for a single user.

The **nopassword** keyword means that no password is required for this user to log in. This is usually most useful in combination with the **autocommand** keyword.

The **password** keyword specifies a possibly encrypted password for this user name.

The *encryptiontype* argument is a single digit number. Currently defined encryption types are 0, which means no encryption, and 7, which specifies a Cisco-defined encryption algorithm. Passwords entered unencrypted are written out with the Cisco encryption. Passwords can contain imbedded spaces and must be the last option specified in the **username** command.

The **accesslist** keyword specifies an outgoing access list that overrides the access list specified in the **access class** line configuration subcommand. It is used for the duration of the user's session. The access list number is specified by the *number* argument.

The **autocommand** keyword causes the command specified by the *command* argument to be issued automatically after the user logs in. When the command is complete, the session is terminated. As the command can be any length and contain imbedded spaces, commands using the **autocommand** keyword must be the last option on the line.

The **noescape** keyword prevents a user from using an escape character on the host to which he is connected.

The **nohangup** keyword prevents the network server from disconnecting the user after an automatic command (set up with the **autocommand** keyword) has completed. Another login prompt is provided to the user.

Line Configuration Subcommand Summary

This section contains a summary of all the line configuration subcommands in alphabetical order.

[**no**] **access-class** *list* {**in**|**out**}

Restricts or permits connections on a line or group of lines to certain Internet addresses. The argument *list* is an integer from 1 through 99 that identifies a specific access list of Internet addresses. The keyword **in** applies to incoming connections; the keyword **out** applies to outgoing Telnet connections.

[**no**] **escape-character** *decimal-number*

Sets or removes the escape character on the specified line. The argument *decimal-number* is either the ASCII decimal representation of the character or a control sequence (Ctrl-E, for example). The default escape character is Ctrl-^.

[**no**] **exec-banner**

Enables or disables a banner. By default, a banner is displayed on the console. To suppress display of a banner, enter the **no** variation of the command.

[**no**] **exec-timeout** *minutes* [*seconds*]

Sets the interval the EXEC waits for user input before resuming the current connection, or if no connections exist, before returning the terminal to the idle state and disconnecting the incoming session. The argument *minutes* is the number of minutes, and the optional argument *seconds* specifies additional interval time in seconds. The default interval is ten minutes; an interval of zero specifies no time-outs. The **no** form of the command restores the default.

[**no**] **length** *screen-length*

Sets the terminal screen length. The argument *screen-length* is the number of lines on the screen. The network server uses this value to determine when to pause during multiple-screen output. The default length is 24 lines. A value of 0 (zero) or the **no** form of the command disables pausing between screens of output.

line [*type-keyword*] *first-line* [*last-line*]

Identifies a specific line for configuration and starts line configuration command collection. The optional argument *type-keyword* specifies the type of line to be configured, it is **console**, **aux**, or **vty**. When the line type is specified, the argument *first-line* is the relative number of the terminal line (or the first line in a contiguous group) you want to configure. The optional argument *last-line* then is the relative number of the last line in a contiguous group you want to configure. If you omit *type*, then *first-line* and *last-line* are absolute rather than relative line numbers.

line aux *port-address*

This variation of the **line** command configures the auxiliary port. When configuring the auxiliary port, address it as line 0.

[**no**] **location** *text*

Enters or removes informational only information about the terminal location and/or status. The argument *text* is the desired description.

[**no**] **login**

Enables or disables password checking for the password specified by the **password** command.

[**no**] **login tacacs**

Causes the TACACS user ID and password checking mechanism to be used instead of the regular password checking. The **no** form of the command disables this mechanism.

[**no**] **notify**

Enables or disables line notification of when a user who has multiple, concurrent Telnet connections has output pending on a connection other than the current line.

[**no**] **padding** *decimal-number count*

Sets or unsets padding for a specified output character. The argument *decimal-number* is the ASCII decimal representation of the character, and the argument *count* is the number of NUL bytes sent after that character.

[**no**] **password** *text*

Specifies a password. The *text* argument specifies a password. It may contain any alpha-numeric characters, including spaces, up to 80 characters. The **no** version of the command removes the specified password.

[**no**] **vacant-message** [*c message c*]

Controls whether or not a banner is displayed on the screen of an idle terminal. The command without any arguments causes the default message to be displayed. The **no vacant-message** command suppresses a banner message. To display a banner, follow **vacant-message** with one or more blank spaces and a delimiting character (*c*) you choose, then type one or more lines of text (*message*), and terminate the text with the second occurrence of the delimiting character.