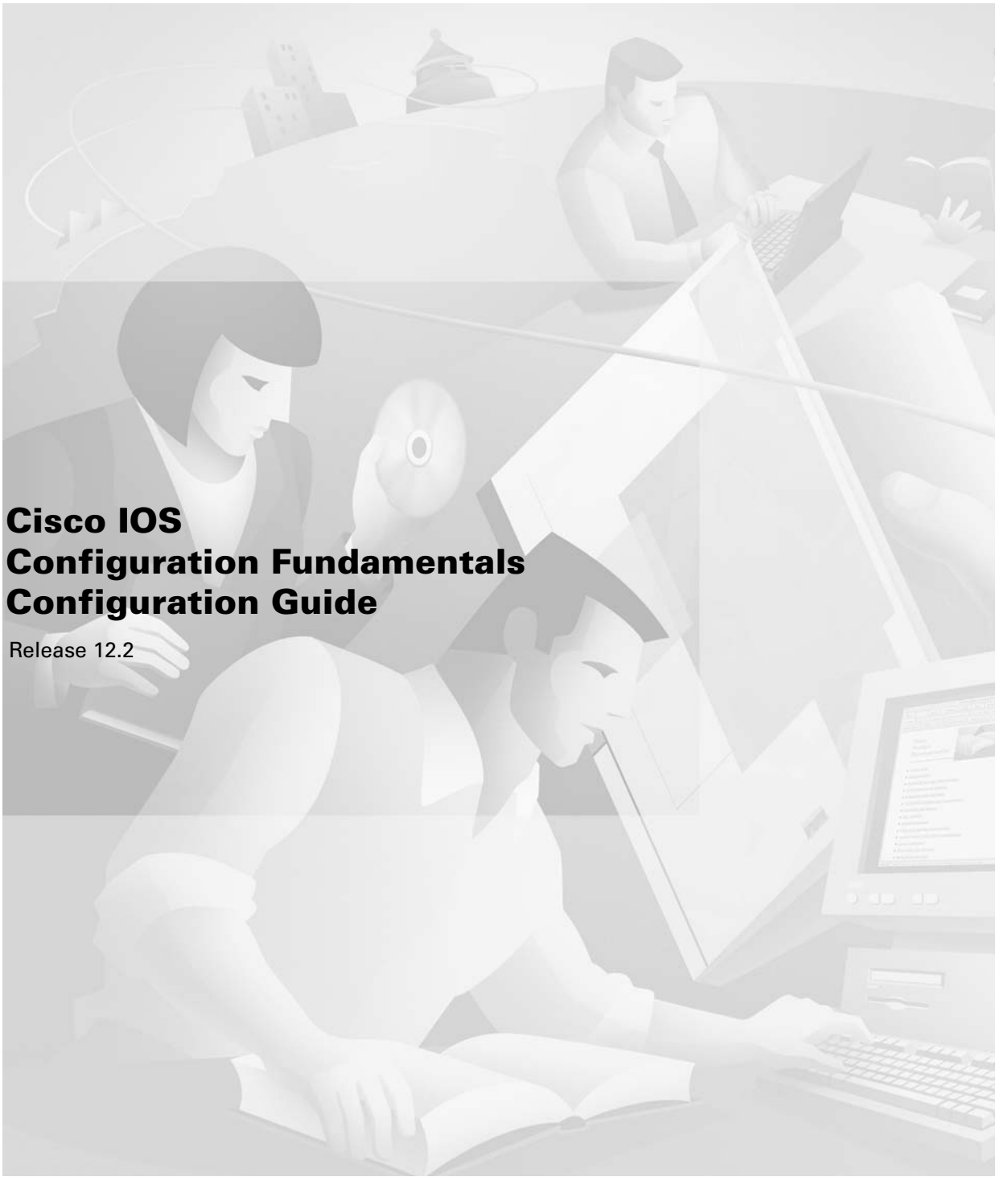


Cisco IOS Configuration Fundamentals Configuration Guide

Release 12.2

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number: DOC-7811739=
Text Part Number: 78-11739-02

CONTENTS

About Cisco IOS Software Documentation

This chapter discusses the objectives, audience, organization, and conventions of Cisco IOS software documentation. It also provides sources for obtaining documentation from Cisco Systems.

Documentation Objectives

Cisco IOS software documentation describes the tasks and commands necessary to configure and maintain Cisco networking devices.

Audience

The Cisco IOS software documentation set is intended primarily for users who configure and maintain Cisco networking devices (such as routers and switches) but who may not be familiar with the tasks, the relationship between tasks, or the Cisco IOS software commands necessary to perform particular tasks. The Cisco IOS software documentation set is also intended for those users experienced with Cisco IOS software who need to know about new features, new configuration options, and new software characteristics in the current Cisco IOS software release.

Documentation Organization

The Cisco IOS software documentation set consists of documentation modules and master indexes. In addition to the main documentation set, there are supporting documents and resources.

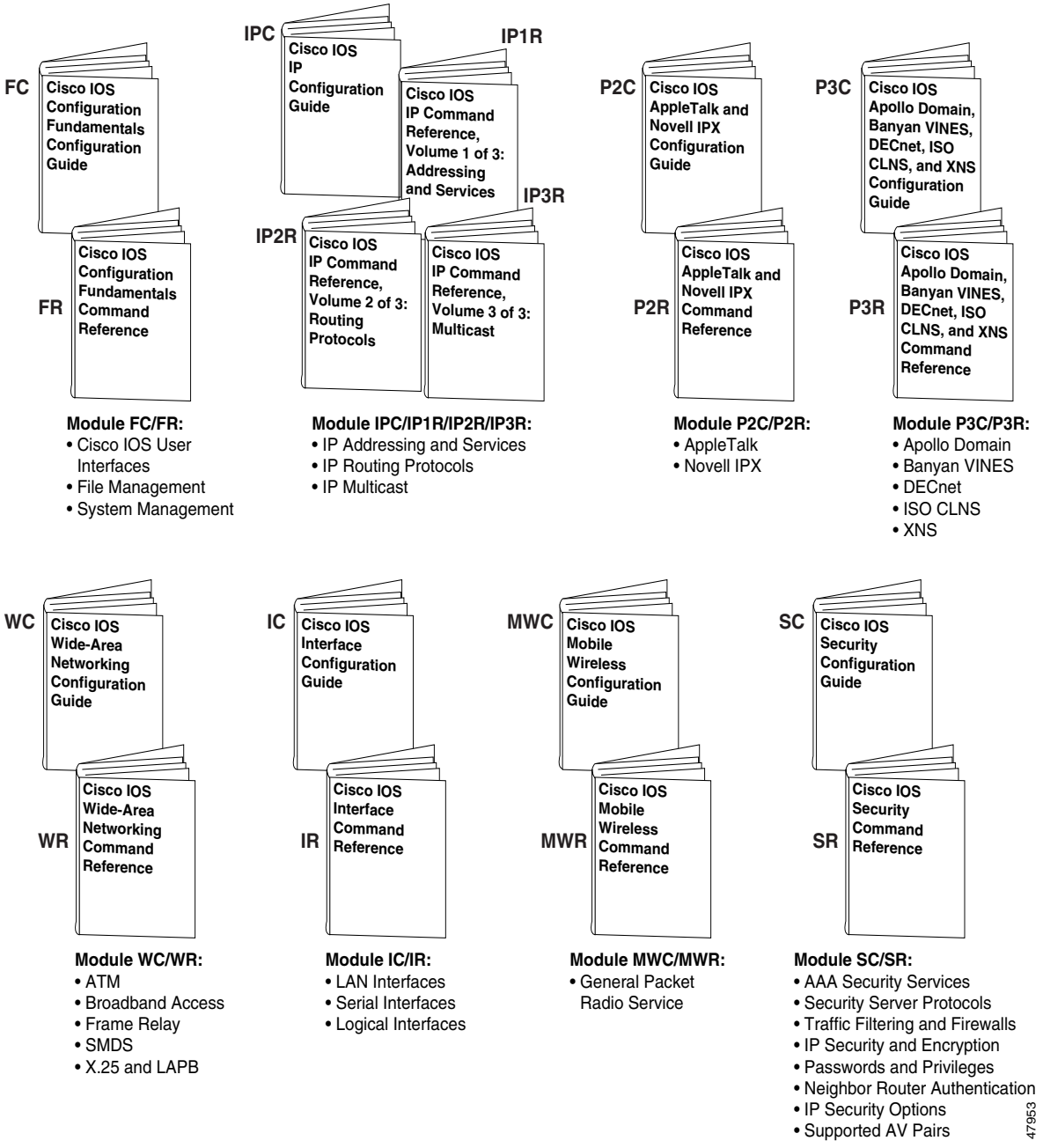
Documentation Modules

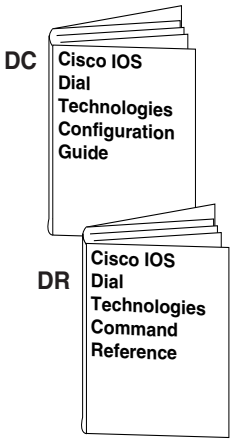
The Cisco IOS documentation modules consist of configuration guides and corresponding command reference publications. Chapters in a configuration guide describe protocols, configuration tasks, and Cisco IOS software functionality and contain comprehensive configuration examples. Chapters in a command reference publication provide complete Cisco IOS command syntax information. Use each configuration guide in conjunction with its corresponding command reference publication.

Figure 1 shows the Cisco IOS software documentation modules.

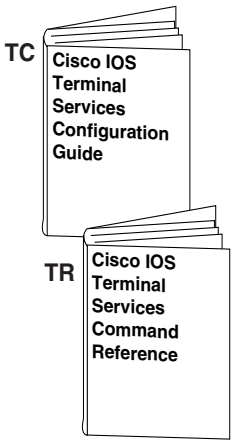
Note The abbreviations (for example, FC and FR) next to the book icons are page designators, which are defined in a key in the index of each document to help you with navigation. The bullets under each module list the major technology areas discussed in the corresponding books.

Figure 1 Cisco IOS Software Documentation Modules

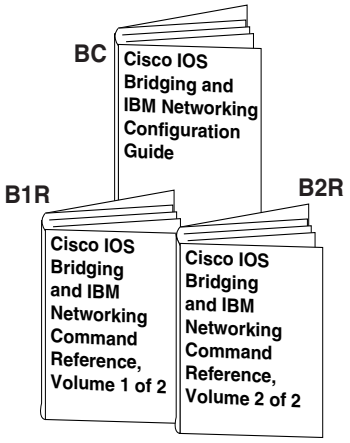




Module DC/DR:
- Preparing for Dial Access
- Modem and Dial Shelf Configuration and Management
- ISDN Configuration
- Signalling Configuration
- Dial-on-Demand Routing Configuration
- Dial-Backup Configuration
- Dial-Related Addressing Services
- Virtual Templates, Profiles, and Networks
- PPP Configuration
- Callback and Bandwidth Allocation Configuration
- Dial Access Specialized Features
- Dial Access Scenarios

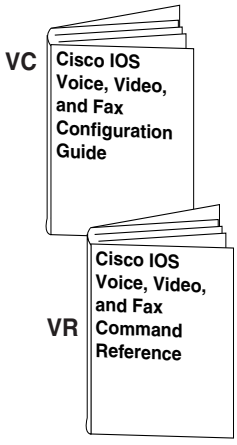


Module TC/TR:
- ARA
- LAT
- NASI
- Telnet
- TN3270
- XRemote
- X.28 PAD
- Protocol Translation

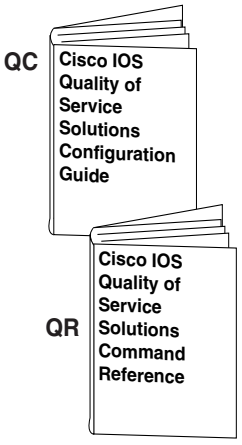


Module BC/B1R:
- Transparent Bridging
- SRB
- Token Ring Inter-Switch Link
- Token Ring Route Switch Module
- RSRB
- DLSw+
- Serial Tunnel and Block Serial Tunnel
- LLC2 and SDLC
- IBM Network Media Translation
- SNA Frame Relay Access
- NCIA Client/Server
- Airline Product Set

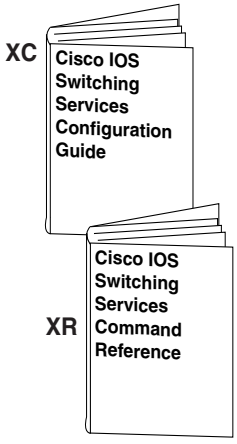
Module BC/B2R:
- DSPU and SNA Service Point
- SNA Switching Services
- Cisco Transaction Connection
- Cisco Mainframe Channel Connection
- CLAW and TCP/IP Offload
- CSNA, CMPC, and CMPC+
- TN3270 Server



Module VC/VR:
- Voice over IP
- Call Control Signalling
- Voice over Frame Relay
- Voice over ATM
- Telephony Applications
- Trunk Management
- Fax, Video, and Modem Support



Module QC/QR:
- Packet Classification
- Congestion Management
- Congestion Avoidance
- Policing and Shaping
- Signalling
- Link Efficiency Mechanisms



Module XC/XR:
- Cisco IOS Switching Paths
- NetFlow Switching
- Multiprotocol Label Switching
- Multilayer Switching
- Multicast Distributed Switching
- Virtual LANs
- LAN Emulation

47954

Master Indexes

Two master indexes provide indexing information for the Cisco IOS software documentation set: an index for the configuration guides and an index for the command references. Individual books also contain a book-specific index.

The master indexes provide a quick way for you to find a command when you know the command name but not which module contains the command. When you use the online master indexes, you can click the page number for an index entry and go to that page in the online document.

Supporting Documents and Resources

The following documents and resources support the Cisco IOS software documentation set:

- *Cisco IOS Command Summary* (two volumes)—This publication explains the function and syntax of the Cisco IOS software commands. For more information about defaults and usage guidelines, refer to the Cisco IOS command reference publications.
- *Cisco IOS System Error Messages*—This publication lists and describes Cisco IOS system error messages. Not all system error messages indicate problems with your system. Some are purely informational, and others may help diagnose problems with communications lines, internal hardware, or the system software.
- *Cisco IOS Debug Command Reference*—This publication contains an alphabetical listing of the **debug** commands and their descriptions. Documentation for each command includes a brief description of its use, command syntax, usage guidelines, and sample output.
- *Dictionary of Internetworking Terms and Acronyms*—This Cisco publication compiles and defines the terms and acronyms used in the internetworking industry.
- New feature documentation—The Cisco IOS software documentation set documents the mainline release of Cisco IOS software (for example, Cisco IOS Release 12.2). New software features are introduced in early deployment releases (for example, the Cisco IOS "T" release train for 12.2, 12.2(x)T). Documentation for these new features can be found in standalone documents called "feature modules." Feature module documentation describes new Cisco IOS software and hardware networking functionality and is available on Cisco.com and the Documentation CD-ROM.
- Release notes—This documentation describes system requirements, provides information about new and changed features, and includes other useful information about specific software releases. See the "Using Software Release Notes" section for more information.
- Caveats documentation—This documentation provides information about Cisco IOS software defects in specific software releases.
- RFCs—RFCs are standards documents maintained by the Internet Engineering Task Force (IETF). Cisco IOS software documentation references supported RFCs when applicable. The full text of referenced RFCs may be obtained on the World Wide Web at http://www.rfc-editor.org/.
- MIBs—MIBs are used for network monitoring. For lists of supported MIBs by platform and release, and to download MIB files, see the Cisco MIB website on Cisco.com at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

New and Changed Information

The following organizational changes have been made since the 12.1 release of the *Cisco IOS Configuration Fundamentals Configuration Guide*:

- The material found in the "Monitoring the Router and Network" chapter of the previous release can now be found in the following chapters:
  - "Configuring SNMP Support"
  - "Configuring RMON Support"
  - "Configuring Cisco Discovery Protocol"
  - "Network Monitoring Using Cisco Service Assurance Agent"
- The chapters titled "System Management Using System Controllers" and "Managing Dial Shelves" have been removed; information on system controllers and dial shelves is now found in the *Cisco IOS Dial Technologies Configuration Guide.*

New Features in Cisco IOS Release 12.2

Cisco IOS Release 12.2 software incorporates the enhancements available in Cisco IOS Release 12.1(1) through 12.1(5) and combines them with the new features introduced in Cisco IOS Release 12.1(1)T through 12.1(5)T.

For a complete list of new features in Cisco IOS Release 12.2, see the "New Features in Cisco IOS Release 12.2" index or the "New Features in Release 12.1 T" online index, available on Cisco.com and the Documentation CD-ROM. The *Cisco IOS Configuration Fundamentals Configuration Guide* for Release 12.2 includes information about the following new features in the Cisco IOS software:

In the "Configuring SNMP Support" chapter:

- Call Tracker plus ISDN and AAA Enhancements for the Cisco AS5300 and Cisco AS5800
- Circuit Interface Identification MIB
- Cisco AAA Server MIB and Additional Enhancements for the Cisco AS5300 and Cisco AS5800
- Cisco AAA Session MIB
- Ethernet-like Interfaces MIB
- Event MIB
- Individual SNMP Trap Support
- Interface Index Persistence
- Interfaces Group MIB Enhancement
- Monitoring Resource Availability on Cisco AS5300 Universal Access Servers
- MSDP MIB
- NTP MIB

In the "Managing Configuration Files" chapter:

- Parser Cache

In the "Network Monitoring Using Cisco Service Assurance Agent" chapter:

- Service Assurance Agent Enhancements

In the "Performing Basic System Management" chapter:

- Trimble Palisade NTP Synchronization Driver for the Cisco 7200 Series

In the "Configuring Web Cache Services Using WCCP" chapter:

- WCCP Redirection on Inbound Interfaces

In the "Configuring Line Cards on the Cisco 7500 Series" Appendix:

- Single Line Card Reload for the Cisco 7500 Series

Identifying Platform Support for Cisco IOS Software Features

Cisco IOS software is packaged in feature sets consisting of software images that support specific platforms. The feature sets available for a specific platform depend on which Cisco IOS software images are included in a release. To identify the set of software images available in a specific release or to find out if a feature is available in a given Cisco IOS software image, see the following sections:

- Using Feature Navigator
- Using Software Release Notes

Using Feature Navigator

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a particular set of features and which features are supported in a particular Cisco IOS image.

Feature Navigator is available 24 hours a day, 7 days a week. To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, e-mail the Contact Database Administration group at cdbadmin@cisco.com. If you do not have an account on Cisco.com, go to http://www.cisco.com/register and follow the directions to establish an account.

To use Feature Navigator, you must have a JavaScript-enabled web browser such as Netscape 3.0 or later, or Internet Explorer 4.0 or later. Internet Explorer 4.0 always has JavaScript enabled. To enable JavaScript for Netscape 3.x or Netscape 4.x, follow the instructions provided with the web browser. For JavaScript support and enabling instructions for other browsers, check with the browser vendor.

Feature Navigator is updated when major Cisco IOS software releases and technology releases occur. You can access Feature Navigator at the following URL:

http://www.cisco.com/go/fn

Using Software Release Notes

Cisco IOS software releases include release notes that provide the following information:

- Platform support information
- Memory recommendations
- Microcode support information
- Feature set tables
- Feature descriptions
- Open and resolved severity 1 and 2 caveats for all platforms

Release notes are intended to be release-specific for the most current release, and the information provided in these documents may not be cumulative in providing information about features that first appeared in previous releases.

Document Conventions

Within Cisco IOS software documentation, the term *router* is generally used to refer to a variety of Cisco products (for example, routers, access servers, and switches). Routers, access servers, and other networking devices that support Cisco IOS software are shown interchangeably within examples. These products are used only for illustrative purposes; that is, an example that shows one product does not necessarily indicate that other products are not supported.

The Cisco IOS documentation set uses the following conventions:

Convention	Description
^ or Ctrl	The ^ and Ctrl symbols represent the Control key. For example, the key combination ^D or Ctrl-D means hold down the Control key while you press the D key. Keys are indicated in capital letters but are not case sensitive.
string	A string is a nonquoted set of characters shown in italics. For example, when setting an SNMP community string to public, do not use quotation marks around the string or the string will include the quotation marks.

Command syntax descriptions use the following conventions:

Convention	Description
boldface	Boldface text indicates commands and keywords that you enter literally as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical line indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical line indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical line indicate a required choice.

Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. For example:

Convention	Description
[x {y | z}]	Braces and a vertical line within square brackets indicate a required choice within an optional element.

Examples use the following conventions:

Convention	Description
`screen`	Examples of information displayed on the screen are set in Courier font.
`boldface screen`	Examples of text that you must enter are set in Courier bold font.
< >	Angle brackets enclose text that is not printed to the screen, such as passwords.
!	An exclamation point at the beginning of a line indicates a comment line. (Exclamation points are also displayed by the Cisco IOS software for certain processes.)
[ ]	Square brackets enclose default responses to system prompts.

The following conventions are used to attract the attention of the reader:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials not contained in this manual.

Timesaver Means the *described action saves time*. You can save time by performing the action described in the paragraph.

Obtaining Documentation

The following sections provide sources for obtaining documentation from Cisco Systems.

World Wide Web

The most current Cisco documentation is available on the World Wide Web at the following website:

http://www.cisco.com

Translated documentation is available at the following website:

http://www.cisco.com/public/countries_languages.html

Documentation CD-ROM

Cisco documentation and additional literature are available in a CD-ROM package, which ships with your product. The Documentation CD-ROM is updated monthly and may be more current than printed documentation. The CD-ROM package is available as a single unit or through an annual subscription.

Ordering Documentation

Cisco documentation can be ordered in the following ways:

- Registered Cisco Direct Customers can order Cisco product documentation from the Networking Products MarketPlace:

  http://www.cisco.com/cgi-bin/order/order_root.pl

- Registered Cisco.com users can order the Documentation CD-ROM through the online Subscription Store:

  http://www.cisco.com/go/subscription

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco corporate headquarters (California, USA) at 408 526-7208 or, in North America, by calling 800 553-NETS(6387).

Documentation Feedback

If you are reading Cisco product documentation on the World Wide Web, you can submit technical comments electronically. Click **Feedback** in the toolbar and select **Documentation**. After you complete the form, click **Submit** to send it to Cisco.

You can e-mail your comments to bug-doc@cisco.com.

To submit your comments by mail, use the response card behind the front cover of your document, or write to the following address:

Cisco Systems, Inc.
Document Resource Connection
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Obtaining Technical Assistance

Cisco provides Cisco.com as a starting point for all technical assistance. Customers and partners can obtain documentation, troubleshooting tips, and sample configurations from online tools. For Cisco.com registered users, additional troubleshooting tools are available from the TAC website.

Cisco.com

Cisco.com is the foundation of a suite of interactive, networked services that provides immediate, open access to Cisco information and resources at anytime, from anywhere in the world. This highly integrated Internet application is a powerful, easy-to-use tool for doing business with Cisco.

Cisco.com provides a broad range of features and services to help customers and partners streamline business processes and improve productivity. Through Cisco.com, you can find information about Cisco and our networking solutions, services, and programs. In addition, you can resolve technical issues with online technical support, download and test software packages, and order Cisco learning materials and merchandise. Valuable online skill assessment, training, and certification programs are also available.

Customers and partners can self-register on Cisco.com to obtain additional personalized information and services. Registered users can order products, check on the status of an order, access technical support, and view benefits specific to their relationships with Cisco.

To access Cisco.com, go to the following website:

http://www.cisco.com

Technical Assistance Center

The Cisco TAC website is available to all customers who need technical assistance with a Cisco product or technology that is under warranty or covered by a maintenance contract.

Contacting TAC by Using the Cisco TAC Website

If you have a priority level 3 (P3) or priority level 4 (P4) problem, contact TAC by going to the TAC website:

http://www.cisco.com/tac

P3 and P4 level problems are defined as follows:

- P3—Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.
- P4—You need information or assistance on Cisco product capabilities, product installation, or basic product configuration.

In each of the above cases, use the Cisco TAC website to quickly find answers to your questions.

To register for Cisco.com, go to the following website:

http://www.cisco.com/register/

If you cannot resolve your technical issue by using the TAC online resources, Cisco.com registered users can open a case online by using the TAC Case Open tool at the following website:

http://www.cisco.com/tac/caseopen

Contacting TAC by Telephone

If you have a priority level 1 (P1) or priority level 2 (P2) problem, contact TAC by telephone and immediately open a case. To obtain a directory of toll-free numbers for your country, go to the following website:

http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml

P1 and P2 level problems are defined as follows:

- P1—Your production network is down, causing a critical impact to business operations if service is not restored quickly. No workaround is available.
- P2—Your production network is severely degraded, affecting significant aspects of your business operations. No workaround is available.

Using Cisco IOS Software

This chapter provides helpful tips for understanding and configuring Cisco IOS software using the command-line interface (CLI). It contains the following sections:

- Understanding Command Modes
- Getting Help
- Using the no and default Forms of Commands
- Saving Configuration Changes
- Filtering Output from the show and more Commands
- Identifying Supported Platforms

For an overview of Cisco IOS software configuration, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*.

For information on the conventions used in the Cisco IOS software documentation set, see the chapter "About Cisco IOS Software Documentation" located at the beginning of this book.

Understanding Command Modes

You use the CLI to access Cisco IOS software. Because the CLI is divided into many different modes, the commands available to you at any given time depend on the mode you are currently in. Entering a question mark (**?**) at the CLI prompt allows you to obtain a list of commands available for each command mode.

When you log in to the CLI, you are in user EXEC mode. User EXEC mode contains only a limited subset of commands. To have access to all commands, you must enter privileged EXEC mode, normally by using a password. From privileged EXEC mode you can issue any EXEC command—user or privileged mode—or you can enter global configuration mode. Most EXEC commands are one-time commands. For example, **show** commands show important status information, and **clear** commands clear counters or interfaces. The EXEC commands are not saved when the software reboots.

Configuration modes allow you to make changes to the running configuration. If you later save the running configuration to the startup configuration, these changed commands are stored when the software is rebooted. To enter specific configuration modes, you must start at global configuration mode. From global configuration mode, you can enter interface configuration mode and a variety of other modes, such as protocol-specific modes.

ROM monitor mode is a separate mode used when the Cisco IOS software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode.

Table 1 describes how to access and exit various common command modes of the Cisco IOS software. It also shows examples of the prompts displayed for each mode.

Table 1 Accessing and Exiting Command Modes

Command Mode	Access Method	Prompt	Exit Method
User EXEC	Log in.	`Router>`	Use the **logout** command.
Privileged EXEC	From user EXEC mode, use the **enable** EXEC command.	`Router#`	To return to user EXEC mode, use the **disable** command.
Global configuration	From privileged EXEC mode, use the **configure terminal** privileged EXEC command.	`Router(config)#`	To return to privileged EXEC mode from global configuration mode, use the **exit** or **end** command, or press **Ctrl-Z**.
Interface configuration	From global configuration mode, specify an interface using an **interface** command.	`Router(config-if)#`	To return to global configuration mode, use the **exit** command. To return to privileged EXEC mode, use the **end** command, or press **Ctrl-Z**.
ROM monitor	From privileged EXEC mode, use the **reload** EXEC command. Press the **Break** key during the first 60 seconds while the system is booting.	`>`	To exit ROM monitor mode, use the **continue** command.

For more information on command modes, refer to the "Using the Command-Line Interface" chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Getting Help

Entering a question mark (**?**) at the CLI prompt displays a list of commands available for each command mode. You can also get a list of keywords and arguments associated with any command by using the context-sensitive help feature.

To get help specific to a command mode, a command, a keyword, or an argument, use one of the following commands:

Command	Purpose
help	Provides a brief description of the help system in any command mode.
*abbreviated-command-entry***?**	Provides a list of commands that begin with a particular character string. (No space between command and question mark.)
abbreviated-command-entry<**Tab**>	Completes a partial command name.
?	Lists all commands available for a particular command mode.
command **?**	Lists the keywords or arguments that you must enter next on the command line. (Space between command and question mark.)

Example: How to Find Command Options

This section provides an example of how to display syntax for a command. The syntax can consist of optional or required keywords and arguments. To display keywords and arguments for a command, enter a question mark (**?**) at the configuration prompt or after entering part of a command followed by a space. The Cisco IOS software displays a list and brief description of available keywords and arguments. For example, if you were in global configuration mode and wanted to see all the keywords or arguments for the **arap** command, you would type **arap ?**.

The <cr> symbol in command help output stands for "carriage return." On older keyboards, the carriage return key is the Return key. On most modern keyboards, the carriage return key is the Enter key. The <cr> symbol at the end of command help output indicates that you have the option to press **Enter** to complete the command and that the arguments and keywords in the list preceding the <cr> symbol are optional. The <cr> symbol by itself indicates that no more arguments or keywords are available and that you must press **Enter** to complete the command.

Table 2 shows examples of how you can use the question mark (**?**) to assist you in entering commands. The table steps you through configuring an IP address on a serial interface on a Cisco 7206 router that is running Cisco IOS Release 12.0(3).

Table 2 How to Find Command Options

Command	Comment

`Router> enable`
`Password: <password>`
`Router#`
	Enter the **enable** command and password to access privileged EXEC commands. You are in privileged EXEC mode when the prompt changes to `Router#`.

`Router# configure terminal`
`Enter configuration commands, one per line. End with CNTL/Z.`
`Router(config)#`
	Enter the **configure terminal** privileged EXEC command to enter global configuration mode. You are in global configuration mode when the prompt changes to `Router(config)#`.

`Router(config)# interface serial ?`
`  <0-6>  Serial interface number`
`Router(config)# interface serial 4 ?`
`  /`
`Router(config)# interface serial 4/ ?`
`  <0-3>  Serial interface number`
`Router(config)# interface serial 4/0`
`Router(config-if)#`
	Enter interface configuration mode by specifying the serial interface that you want to configure using the **interface serial** global configuration command.
	Enter **?** to display what you must enter next on the command line. In this example, you must enter the serial interface slot number and port number, separated by a forward slash.
	You are in interface configuration mode when the prompt changes to `Router(config-if)#`.

Table 2 How to Find Command Options (continued)

Command	Comment

`Router(config-if)# ?`
`Interface configuration commands:`
`  .`
`  .`
`  .`
`  ip  Interface Internet Protocol config commands`
`  keepalive  Enable keepalive`
`  lan-name  LAN Name command`
`  llc2  LLC2 Interface Subcommands`
`  load-interval  Specify interval for load calculation for an interface`
`  locaddr-priority  Assign a priority group`
`  logging  Configure logging for interface`
`  loopback  Configure internal loopback on an interface`
`  mac-address  Manually set interface MAC address`
`  mls  mls router sub/interface commands`
`  mpoa  MPOA interface configuration commands`
`  mtu  Set the interface Maximum Transmission Unit (MTU)`
`  netbios  Use a defined NETBIOS access list or enable name-caching`
`  no  Negate a command or set its defaults`
`  nrzi-encoding  Enable use of NRZI encoding`
`  ntp  Configure NTP`
`  .`
`  .`
`  .`
`Router(config-if)#`
	Enter **?** to display a list of all the interface configuration commands available for the serial interface. This example shows only some of the available interface configuration commands.

`Router(config-if)# ip ?`
`Interface IP configuration subcommands:`
`  access-group  Specify access control for packets`
`  accounting  Enable IP accounting on this interface`
`  address  Set the IP address of an interface`
`  authentication  authentication subcommands`
`  bandwidth-percent  Set EIGRP bandwidth limit`
`  broadcast-address  Set the broadcast address of an interface`
`  cgmp  Enable/disable CGMP`
`  directed-broadcast  Enable forwarding of directed broadcasts`
`  dvmrp  DVMRP interface commands`
`  hello-interval  Configures IP-EIGRP hello interval`
`  helper-address  Specify a destination address for UDP broadcasts`
`  hold-time  Configures IP-EIGRP hold time`
`  .`
`  .`
`  .`
`Router(config-if)# ip`
	Enter the command that you want to configure for the interface. This example uses the **ip** command.
	Enter **?** to display what you must enter next on the command line. This example shows only some of the available interface IP configuration commands.

Table 2 How to Find Command Options (continued)

Command	Comment

`Router(config-if)# ip address ?`
`  A.B.C.D  IP address`
`  negotiated  IP Address negotiated over PPP`
`Router(config-if)# ip address`
	Enter the command that you want to configure for the interface. This example uses the **ip address** command.
	Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP address or the **negotiated** keyword.
	A carriage return (<cr>) is not displayed; therefore, you must enter additional keywords or arguments to complete the command.

`Router(config-if)# ip address 172.16.0.1 ?`
`  A.B.C.D  IP subnet mask`
`Router(config-if)# ip address 172.16.0.1`
	Enter the keyword or argument you want to use. This example uses the 172.16.0.1 IP address.
	Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP subnet mask.
	A <cr> is not displayed; therefore, you must enter additional keywords or arguments to complete the command.

`Router(config-if)# ip address 172.16.0.1 255.255.255.0 ?`
`  secondary  Make this IP address a secondary address`
`  <cr>`
`Router(config-if)# ip address 172.16.0.1 255.255.255.0`
	Enter the IP subnet mask. This example uses the 255.255.255.0 IP subnet mask.
	Enter **?** to display what you must enter next on the command line. In this example, you can enter the **secondary** keyword, or you can press **Enter**.
	A <cr> is displayed; you can press **Enter** to complete the command, or you can enter another keyword.

`Router(config-if)# ip address 172.16.0.1 255.255.255.0` `Router(config-if)#`	In this example, Enter is pressed to complete the command.

Using the no and default Forms of Commands

Almost every configuration command has a **no** form. In general, use the **no** form to disable a function. Use the command without the **no** keyword to reenable a disabled function or to enable a function that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, use the **no ip routing** command; to reenable IP routing, use the **ip routing** command. The Cisco IOS software command reference publications provide the complete syntax for the configuration commands and describe what the **no** form of a command does.

Configuration commands also can have a **default** form, which returns the command settings to the default values. Most commands are disabled by default, so in such cases using the **default** form has the same result as using the **no** form of the command. However, some commands are enabled by default and

have variables set to certain default values. In these cases, the **default** form of the command enables the command and sets the variables to their default values. The Cisco IOS software command reference publications describe the effect of the **default** form of a command if the command functions differently than the **no** form.

Saving Configuration Changes

Use the **copy system:running-config nvram:startup-config** command to save your configuration changes to the startup configuration so that the changes will not be lost if the software reloads or a power outage occurs. For example:

```
Router# copy system:running-config nvram:startup-config
Building configuration...
```

It might take a minute or two to save the configuration. After the configuration has been saved, the following output appears:

```
[OK]
Router#
```

On most platforms, this task saves the configuration to NVRAM. On the Class A Flash file system platforms, this task saves the configuration to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Filtering Output from the show and more Commands

In Cisco IOS Release 12.0(1)T and later releases, you can search and filter the output of **show** and **more** commands. This functionality is useful if you need to sort through large amounts of output or if you want to exclude output that you need not see.

To use this functionality, enter a **show** or **more** command followed by the "pipe" character (|); one of the keywords **begin**, **include**, or **exclude**; and a regular expression on which you want to search or filter (the expression is case-sensitive):

command | {**begin** | **include** | **exclude**} *regular-expression*

The output matches certain lines of information in the configuration file. The following example illustrates how to use output modifiers with the **show interface** command when you want the output to include only lines in which the expression "protocol" appears:

```
Router# show interface | include protocol

FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

For more information on the search and filter functionality, refer to the "Using the Command-Line Interface" chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.2.

Identifying Supported Platforms

Cisco IOS software is packaged in feature sets consisting of software images that support specific platforms. The feature sets available for a specific platform depend on which Cisco IOS software images are included in a release. To identify the set of software images available in a specific release or to find out if a feature is available in a given Cisco IOS software image, see the following sections:

- Using Feature Navigator
- Using Software Release Notes

Using Feature Navigator

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a particular set of features and which features are supported in a particular Cisco IOS image.

Feature Navigator is available 24 hours a day, 7 days a week. To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, e-mail the Contact Database Administration group at cdbadmin@cisco.com. If you do not have an account on Cisco.com, go to http://www.cisco.com/register and follow the directions to establish an account.

To use Feature Navigator, you must have a JavaScript-enabled web browser such as Netscape 3.0 or later, or Internet Explorer 4.0 or later. Internet Explorer 4.0 always has JavaScript enabled. To enable JavaScript for Netscape 3.x or Netscape 4.x, follow the instructions provided with the web browser. For JavaScript support and enabling instructions for other browsers, check with the browser vendor.

Feature Navigator is updated when major Cisco IOS software releases and technology releases occur. You can access Feature Navigator at the following URL:

http://www.cisco.com/go/fn

Using Software Release Notes

Cisco IOS software releases include release notes that provide the following information:

- Platform support information
- Memory recommendations
- Microcode support information
- Feature set tables
- Feature descriptions
- Open and resolved severity 1 and 2 caveats for all platforms

Release notes are intended to be release-specific for the most current release, and the information provided in these documents may not be cumulative in providing information about features that first appeared in previous releases.

Configuration Fundamentals Overview

This chapter provides an overview of the *Cisco IOS Configuration Fundamentals Configuration Guide* for Cisco IOS Release 12.2. It includes descriptions of the parts and chapters of this document, and suggestions on which parts of the documentation to read to perform common tasks.

Organization of This Guide

The *Cisco IOS Configuration Fundamentals Configuration Guide* is divided into three main parts:

- Cisco IOS User Interfaces
- File Management
- System Management

This section provides a description of the chapters within each part.

Cisco IOS User Interfaces

The user interface chapters describe the different methods of entering commands into a router and altering the user environment:

- "Using the Command-Line Interface"

  The command-line interface (CLI) is the primary means of configuring Cisco IOS software-based devices. This chapter provides an overview of the CLI, and discusses its editing features, context-sensitvie help, and other features.

- "Using AutoInstall and Setup"

  The Cisco IOS software includes two features that simplify or automate the configuration of Cisco devices: AutoInstall and Setup. AutoInstall allows a network manager to load configuration files onto new Cisco devices automatically. Setup guides a user throught the initial configuration of a Cisco device. This chapter describes how to set up your network for AutoInstall, and how to use Setup.

- "Configuring Operating Characteristics for Terminals"

  A basic method of accessing the CLI is to connect a terminal to the router through the console port or one of the tty lines. This terminal connection uses default settings, which should work for most terminal sessions. However, you may want to alter the terminal settings. This chapter provides details on how to perform these alterations.

- "Managing Connections, Menus, and System Banners"

  This chapter provides details on managing connections you make to other hosts, displaying messages to users connecting to your router, and setting up user menus.

- "Using the Cisco Web Browser User Interface"

  This chapter provides detailed information on using the Cisco IOS web browser user interface (UI) to configure and monitor your router, as an alternative to using the CLI. It also explains how to configure the Web Browser interface for other users.

File Management

The file management chapters describe the tasks associated with copying, saving, moving, and loading different types of files, such as configuration files, images, and microcode:

- "Using the Cisco IOS File System"
  This chapter descibes how to manage files using the Cisco IOS File System (IFS), which provides a common syntax for managing all file systems on Cisco devices, including Flash memory file systems and network file systems, as well as for any other endpoints used for reading or writing data.
- "Managing Configuration Files"
  This chapter describes how to modify configuration files, as well as how to upload, store, and download configuration files. This chapter also explains how to specify which configuration file the system should use at startup.
- "Loading and Maintaining System Images"
  This chapter describes how to download images from servers, store images on servers, and specify which image is loaded at system startup. If you are not upgrading your system image and you do not want to change image booting procedures, you do not need to read this chapter.
- "Maintaining System Memory"
  This chapter describes the different types of memory your router may have and how to use this memory to manage files.
- "Rebooting"
  This chapter focuses on tasks related to the rebooting procedure. Read this chapter if you want to change which image or configuration file is loaded at system startup. This chapter also discusses ROM Monitor mode, which allows you to boot the router manually.
- "Configuring Basic File Transfer Services"
  This chapter describes how to configure your router to function as a server, or use the remote shell (rsh) and remote copy (rcp) functions. As a TFTP server, your router can provide other routers with images and configuration files over the network. The rsh and rcp functions allow users to remotely execute commands or copy files to or from another host. This chapter also addresses optional configuration of Maintenance Operation Protocol (MOP) and Boot Operation Protocol (BOOTP) services.

System Management

The system management chapters discuss tasks that allow you to maintain your router after it is configured with the network, routing, and WAN protocols. These chapters discuss ways you can fine-tune the router and maintain it over time. These chapters also discuss router and network monitoring tools used for gathering information about connected devices and network performance.

- "Performing Basic System Management"
  Discusses basic optional tasks. For example, you can change the name of the router, create command aliases, enable minor services, and set time and calendar services.
- "Troubleshooting and Fault Management"
  Provides an introduction to troubleshooting techniques (including use of **show** commands), error message logging, and debugging commands. If you are troubleshooting a particular protocol, read this chapter to learn how to log system error messages and use debugging commands. Then, refer to the chapter in the documentation set that documents your protocol. For detailed troubleshooting information, see the *Internetwork Troubleshooting Guide*.
- "Configuring SNMP Support"
  Describes the steps for configuring Simple Network Management Protocol (SNMP) on your router.
- "Configuring Cisco Discovery Protocol"
  Describes the Cisco Discovery Protocol (CDP), and how to use CDP to discover other local devices.
- "Configuring RMON Support"
  Describes the Remote Monitoring (RMON) features available on Cisco routers to supplement SNMP use.
- "Network Monitoring Using Cisco Service Assurance Agent"
  Describes the Cisco Service Assurance Agent (SA Agent), and how to use SA Agent operations to monitor network performance and ensure levels of service.
- "Configuring Web Cache Services Using WCCP"
  Describes the Web Cache Control Protocol, a Cisco-developed content-routing technology that allows you to utilize cache engines (such as the Cisco Cache Engine 550) and web-caches in your network.

Task-Oriented Documentation Approaches

The above parts and chapters of the *Cisco IOS Configuration Fundamentals Configuration Guide* suggest a framework for learning configuration and maintenance tasks. This section provides some suggestions on alternate paths you can take through the documentation to learn about particular topics or tasks, focusing on common configuration topics that span multiple chapters of this book.

For complete descriptions of the configuration commands introduced in this guide, see the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*, which is the second book of this documentation module.

Overview of Router Configuration Tasks

To configure your router or access server, you must perform several tasks. Initially, you must determine the following:

- Which network protocols you are supporting (for example, AppleTalk, IP, Novell IPX, and so on)
- The addressing plan for each network protocol
- Which routing protocol you will use for each network protocol
- Which WAN protocols you will run on each interface (for example, Frame Relay, HDLC, SMDS, X.25, and so on)

Then, refer to the *Cisco Product Catalog* and the platform-specific release notes for a list of Cisco-supported protocols, interfaces, and platforms. Set up the hardware as described in the documentation shipped with your product. Configure any user interface, file management, or interface management tasks as described in this book. Configure protocol-specific features on your router or access server as described in the appropriate chapters of the other Cisco IOS software configuration guides.

Understanding the Cisco IOS Command-Line Interface

If you are not familiar with the Cisco IOS command-line interface, read the following sections to gain a basic understanding of the user interface and basic configuration tasks:

In the "Using the Command-Line Interface" chapter:

- Understanding Cisco IOS Command Modes
- Using the No and Default Forms of Commands
- Getting Context-Sensitive Help Within a Command Mode
- Checking Command Syntax
- Using CLI Command History
- Using Command-Line Editing Features and Shortcuts

In the "Modifying, Downloading, and Maintaining Configuration Files" chapter:

- Displaying Configuration File Information
- Understanding Configuration Files
- Entering Configuration Mode and Selecting a Configuration Source
- Configuring Cisco IOS from the Terminal
- Reexecuting the Configuration Commands in Startup Configuration
- Clearing the Configuration Information

In the "Performing Basic System Management" chapter:

- Setting the Router Name

You may also wish to review the Appendix of this book, "Cisco IOS Command Modes," for a summary description of modes available in the command-line interface.

Storing or Obtaining Configuration Files or Images from a Server

You might want to save a configuration or image on a server or upgrade your image to a different software release. If you will be storing or obtaining configuration files or images from a server, read the following sections:

In the "Managing Configuration Files"chapter:

- Copying Configuration Files from the Router to a Network Server
- Copying Configuration Files from a Network Server to the Router
- Maintaining Configuration Files Larger than NVRAM
- Copying Configuration Files Between Different Locations

In the "Maintaining System Memory" chapter:

- Partitioning Flash Memory
- Using Flash Load Helper to Upgrade Software on Run-from-Flash Systems

Changing the Image or Configuration File Loaded by the Router

If you want to change the image or configuration file used when the system reloads, read the following sections:

In the "Managing Configuration Files" chapter:

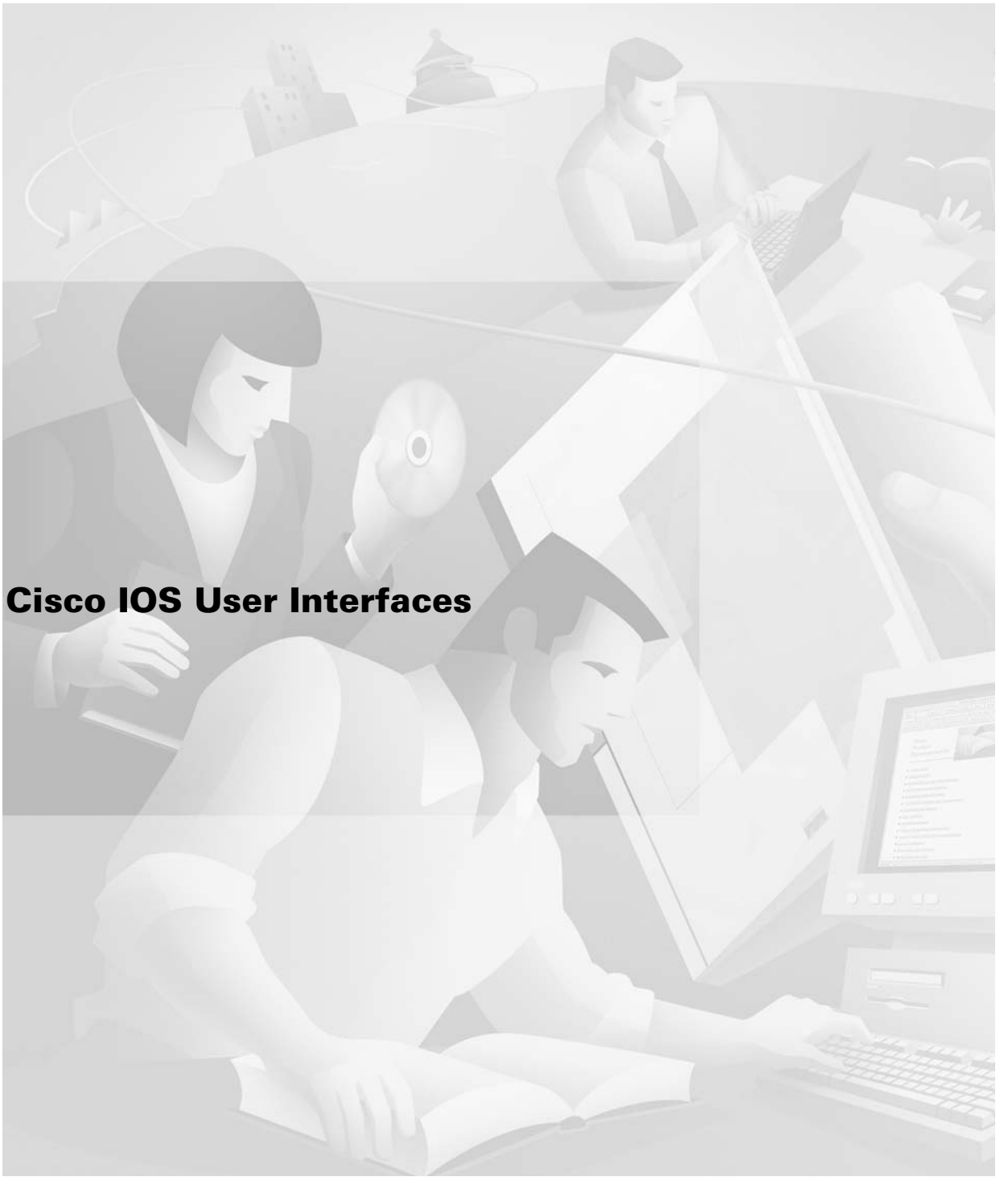
- Specifying the Startup Configuration File

In the"Loading and Maintaining System Images" chapter:

- Specifying the Startup System Image in the Configuration File

In the "Rebooting" chapter:

- Displaying Booting Information
- Rebooting Procedures
- Modifying the Configuration Register Boot Field
- Setting Environment Variables



Cisco IOS User Interfaces

Using the Command-Line Interface

The Cisco IOS command-line interface (CLI) is the primary user interface used for configuring, monitoring, and maintaining Cisco devices. This user interface allows you to directly and simply execute Cisco IOS commands, whether using a router console or terminal, or using remote access methods.

This chapter describes the basic features of the Cisco IOS CLI and how to use them. Topics covered include an introduction to Cisco IOS command modes, navigation and editing features, help features, and command history features.

Additional user interfaces include Setup mode (used for first-time startup), the Cisco Web Browser, and user menus configured by a system administrator. For information about Setup mode, see the "Using AutoInstall and Setup" chapter of this book. For information on issuing commands using the Cisco Web Browser, see the "Using the Cisco Web Browser User Interface" chapter of this book. For information on user menus, see the "Managing Connections, Menus, and System Banners" chapter of this book.

For a complete description of the user interface commands in this chapter, refer to the "Basic Command-Line Interface Commands" chapter of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

This chapter contains the following sections:

- Cisco IOS CLI Command Modes Overview
- Cisco IOS CLI Task List
- Using the Cisco IOS CLI Examples

Cisco IOS CLI Command Modes Overview

To aid in the configuration of Cisco devices, the Cisco IOS command-line interface is divided into different command modes. Each command mode has its own set of commands available for the configuration, maintenance, and monitoring of router and network operations. The commands available to you at any given time depend on the mode you are in. Entering a question mark (**?**) at the system prompt (router prompt) allows you to obtain a list of commands available for each command mode.

The use of specific commands allows you to navigate from one command mode to another. The standard order that a user would access the modes is as follows: user EXEC mode; privileged EXEC mode; global configuration mode; specific configuration modes; configuration submodes; and configuration subsubmodes.

When you start a session on a router, you generally begin in *user EXEC mode*, which is one of two access levels of the EXEC mode. For security purposes, only a limited subset of EXEC commands are available in user EXEC mode. This level of access is reserved for tasks that do not change the configuration of the router, such as determining the router status.

In order to have access to all commands, you must enter *privileged EXEC mode*, which is the second level of access for the EXEC mode. Normally, you must enter a password to enter privileged EXEC mode. In privileged EXEC mode, you can enter any EXEC command, as the privileged EXEC mode is a superset of the user EXEC mode commands.

Most EXEC mode commands are one-time commands, such as **show** or **more** commands, which show the current configuration status, and **clear** commands, which clear counters or interfaces. EXEC mode commands are not saved across reboots of the router.

From privileged EXEC mode, you can enter *global configuration mode*. In this mode, you can enter commands that configure general system characteristics. You also can use global configuration mode to enter specific configuration modes. Configuration modes, including global configuration mode, allow you to make changes to the running configuration. If you later save the configuration, these commands are stored across router reboots.

From global configuration mode you can enter a variety of protocol-specific or feature-specific configuration modes. The CLI hierarchy requires that you enter these specific configuration modes only through global configuration mode. As an example, this chapter describes *interface configuration mode*, a commonly used configuration mode.

From configuration modes, you can enter configuration submodes. Configuration submodes are used for the configuration of specific features within the scope of a given configuration mode. As an example, this chapter describes the *subinterface configuration mode*, a submode of the interface configuration mode.

ROM monitor mode is a separate mode used when the router cannot boot properly. If your system (router, switch, or access server) does not find a valid system image to load when it is booting, the system will enter ROM monitor mode. ROM monitor (ROMMON) mode can also be accessed by interrupting the boot sequence during startup.

The following sections contain detailed information on these command modes:

- User EXEC Mode
- Privileged EXEC Mode
- Global Configuration Mode
- Interface Configuration Mode
- Subinterface Configuration Mode
- ROM Monitor Mode

Table 3 follows these sections and summarizes the main Cisco IOS command modes.

User EXEC Mode

Logging in to the router places you in user EXEC command mode (unless the system is configured to take you immediately to privileged EXEC mode). Typically, log-in will require a user name and a password. You may try three times to enter a password before the connection attempt is refused.

Note For information on setting the password, see the "Configuring Passwords and Privileges" chapter in the Release 12.2 *Cisco IOS Security Configuration Guide* and the "Using AutoInstall and Setup" chapter in this document.

The EXEC commands available at the user level are a subset of those available at the privileged level. In general, the user EXEC commands allow you to connect to remote devices, change terminal line settings on a temporary basis, perform basic tests, and list system information.

To list the available user EXEC commands, use the following command:

Command	Purpose
`Router> ?`	Lists the user EXEC commands.

The user EXEC mode prompt consists of the host name of the device followed by an angle bracket (>), as shown in the following example:

```
Router>
```

The default host name is generally `Router,` unless it has been changed during initial configuration using the **setup** EXEC command. You also change the host name using the **hostname** global configuration command.

Note Examples in Cisco IOS documentation assume the use of the default name of "Router." Different devices (for example, access servers) may use a different default name. If the routing device (router, access server, or switch) has been named with the **hostname** command, that name will appear as the prompt instead of the default name.

To list the commands available in user EXEC mode, enter a question mark (**?**) as shown in the following example:

```
Router> ?
Exec commands:
 <1-99>           Session number to resume
 connect          Open a terminal connection
 disconnect       Disconnect an existing telnet session
 enable           Turn on privileged commands
 exit             Exit from Exec mode
 help             Description of the interactive help system
 lat              Open a lat connection
 lock             Lock the terminal
 login            Log in as a particular user
 logout           Exit from Exec mode and log out
 menu             Start a menu-based user interface
 mbranch          Trace multicast route for branch of tree
 mrbranch         Trace reverse multicast route to branch of tree
 mtrace           Trace multicast route to group
 name-connection  Name an existing telnet connection
 pad              Open a X.29 PAD connection
 ping             Send echo messages
 resume           Resume an active telnet connection
 show             Show running system information
 systat           Display information about terminal lines
 telnet           Open a telnet connection
 terminal         Set terminal line parameters
 tn3270           Open a tn3270 connection
```

```
trace           Trace route to destination
where           List active telnet connections
x3              Set X.3 parameters on PAD
```

The list of commands will vary depending on the software feature set and router platform you are using.

Note You can enter commands in uppercase, lowercase, or mixed case. Only passwords are case sensitive. However, Cisco IOS documentation convention is to always present commands in lowercase.

Privileged EXEC Mode

Because many privileged EXEC mode commands set operating parameters, privileged-level access should be password protected to prevent unauthorized use. The privileged EXEC command set includes those commands contained in user EXEC mode. Privilege EXEC mode also provides access to configuration modes through the **configure** command, and includes advanced testing commands, such as **debug**.

The privileged EXEC mode prompt consists of the host name of the device followed by a pound sign (#), as shown in the following example:

```
Router#
```

To access privileged EXEC mode, use the following command:

Command	Purpose
`Router> enable`	Enables privileged EXEC mode. After issuing the enable command, the system will prompt you for a password.

Note that privileged EXEC mode is sometimes referred to as "enable mode," because the **enable** command is used to enter the mode.

If a password has been configured on the system, you will be prompted to enter it before being allowed access to privileged EXEC mode. The password is not displayed on the screen and is case sensitive. If an enable password has not been set, privileged EXEC mode can be accessed only from the router console (terminal connected to the console port). The system administrator uses the **enable secret** or **enable password** global configuration commands to set the password that restricts access to privileged mode. For information on setting the passwords, see the "Configuring Passwords and Privileges" chapter in the Release 12.2 *Cisco IOS Security Configuration Guide.*

To return to user EXEC mode, use the following command:

Command	Purpose
`Router# disable`	Exits from privileged EXEC mode to user EXEC mode.

The following example shows the process of accessing privileged EXEC mode:

```
Router> enable
Password:<letmein>
Router#
```

Note that the password will not be displayed as you type, but is shown here for illustrational purposes.To list the commands available in privileged EXEC mode, issue the **?** command at the prompt. From privileged EXEC mode you can access global configuration mode, which is described in the following section.

Note Because the privileged EXEC command set contains all of the commands available in user EXEC mode, some commands can be entered in either mode. In Cisco IOS documentation, commands that can be entered in either user EXEC mode or privileged EXEC mode are referred to as EXEC mode commands. If user or privileged is not specified in the documentation, assume that you can enter the referenced commands in either mode.

Global Configuration Mode

The term "global" is used to indicate characteristics or features that affect the system as a whole. Global configuration mode is used to configure your system globally, or to enter specific configuration modes to configure specific elements such as interfaces or protocols. Use the **configure terminal** privileged EXEC command to enter global configuration mode.

To access global configuration mode, use the following command in privileged EXEC mode:

Command	Purpose
`Router# configure terminal`	From privileged EXEC mode, enters global configuration mode.

The following example shows the process of entering global configuration mode from privileged EXEC mode:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

Note that the system prompt changes to indicate that you are now in global configuration mode. The prompt for global configuration mode consists of the host-name of the device followed by `(config)` and the pound sign (`#`). To list the commands available in privileged EXEC mode, issue the **?** command at the prompt.

Commands entered in global configuration mode update the running configuration file as soon as they are entered. In other words, changes to the configuration take effect each time you press the Enter or Return key at the end of a valid command. However, these changes are not saved into the startup configuration file until you issue the **copy running-config startup-config** EXEC mode command. This behavior is explained in more detail later in this document.

As shown in the example above, the system dialogue prompts you to end your configuration session (exit configuration mode) by pressing the Control (Ctrl) and "z" keys simultaneously; when you press these keys, **^Z** is printed to the screen. You can actually end your configuration session by entering the Ctrl-Z key combination, using the **end** command, using the Ctrl-C key combination. The **end** command is the recommended way to indicate to the system that you are done with the current configuration session.

Warning **If you use Ctrl-Z at the end of a command line in which a valid command has been typed, that command will be added to the running configuration file. In other words, using Ctrl-Z is equivalent to hitting the Enter (Carriage Return) key before exiting. For this reason, it is safer to end your configuration session using the end command. Alternatively, you can use the Ctrl-C key combination to end your configuration session without sending a Carriage Return signal.**

You can also use the **exit** command to return from global configuration mode to EXEC mode, but this only works in global configuration mode. Pressing Ctrl-Z or entering the **end** command will always take you back to EXEC mode regardless of which configuration mode or configuration submode you are in.

To exit global configuration command mode and return to privileged EXEC mode, use one of the following commands:

Command	Purpose

`Router(config)# end`
or
`Router(config)# ^Z`
	Ends the current configuration session and returns to privileged EXEC mode.

`Router(config)# exit`	Exits the current command mode and returns to the preceding mode. For example, exits from global configuration mode to privileged EXEC mode.

From global configuration mode, you can enter a number of protocol-specific, platform-specific, and feature-specific configuration modes. For a complete list of configuration modes, see the "Cisco IOS Command Modes" appendix in this book. This appendix provides references to the appropriate documentation module for information about specific configuration modes.

Interface configuration mode, described in the following section, is an example of a configuration mode you can enter from global configuration mode.

Interface Configuration Mode

One example of a specific configuration mode you enter from global configuration mode is interface configuration mode.

Many features are enabled on a per-interface basis. Interface configuration commands modify the operation of an interface such as an Ethernet, FDDI, or serial port. Interface configuration commands always follow an **interface** global configuration command, which defines the interface type.

For details on interface configuration commands that affect general interface parameters, such as bandwidth or clock rate, refer to the Release 12.2 *Cisco IOS Interface Configuration Guide*. For protocol-specific commands, refer to the appropriate Cisco IOS software command reference.

To access and list the interface configuration commands, use the following command:

Command	Purpose
`Router(config)# interface` *`type number`*	Specifies the interface to be configured, and enters interface configuration mode.

In the following example, the user enter interface configuration mode for serial interface 0. The new prompt, *`hostname`*`(config-if)#,` indicates interface configuration mode.

```
Router(config)# interface serial 0
Router(config-if)#
```

To exit interface configuration mode and return to global configuration mode, enter the **exit** command.

Configuration submodes are configuration modes entered from other configuration modes (besides global configuration mode). Configuration submodes are for the configuration of specific elements within the configuration mode. For a complete list of configuration submodes, see the "Cisco IOS Command Modes" appendix in this book. One example of a configuration submode is subinterface configuration mode, described in the following section.

Subinterface Configuration Mode

From interface configuration mode, you can enter subinterface configuration mode. Subinterface configuration mode is a submode of interface configuration mode. In subinterface configuration mode you can configure multiple virtual interfaces (called subinterfaces) on a single physical interface. Subinterfaces appear to be distinct physical interfaces to the various protocols. For example, Frame Relay networks provide multiple point-to-point links called permanent virtual circuits (PVCs). PVCs can be grouped under separate subinterfaces that in turn are configured on a single physical interface. From a bridging spanning-tree viewpoint, each subinterface is a separate bridge port, and a frame arriving on one subinterface can be sent out on another subinterface.

Subinterfaces also allow multiple encapsulations for a protocol on a single interface. For example, a router or access server can receive an ARPA-framed IPX packet and forward the packet back out the same physical interface as a SNAP-framed IPX packet.

For detailed information on how to configure subinterfaces, refer to the appropriate documentation module for a specific protocol in the Cisco IOS software documentation set.

To access subinterface configuration mode, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# interface` *type number*	Specifies the virtual interface to be configured and enters subinterface configuration mode.

In the following example, a subinterface is configured for serial line 2, which is configured for Frame Relay encapsulation. The subinterface is identified as "2.1" to indicate that it is subinterface 1 of serial interface 2. The new prompt *hostname*`(config-subif)#` indicates subinterface configuration mode. The subinterface can be configured to support one or more Frame Relay PVCs.

```
Router(config)# interface serial 2
Router(config-if)# encapsulation frame-relay
Router(config-if)# interface serial 2.1
Router(config-subif)#
```

To exit subinterface configuration mode and return to interface configuration mode, use the **exit** command. To end your configuration session and return to privileged EXEC mode, press Ctrl-Z or enter the **end** command.

ROM Monitor Mode

ROM monitor mode (ROMMON) runs from a specialized software image, and is used to manually locate a valid system software image from which to boot the system (ROM monitor mode is also sometimes called "boot mode").

If your system (router, switch, or access server) does not find a valid system image to load, the system will enter ROM monitor mode. ROM monitor mode can also be accessed by interrupting the boot sequence during startup. From ROM monitor mode, you can boot the device or perform diagnostic tests.

On most systems you can enter ROM monitor mode by entering the **reload** EXEC command and then issuing the Break command during the first 60 seconds of startup. The Break command is issued by pressing the Break key on your keyboard or by using the Break key-combination (the default Break key-combination is Ctrl-C).

Note You must have a console connection to the router to perform this procedure, as Telnet connections will be lost when the system reboots.

To access ROM monitor mode from EXEC mode, perform the following steps:

Step 1 Enter the **reload** command in EXEC mode. After issuing this command and responding to the system prompts as necessary, the system will begin reloading the system software image.

Step 2 Issue the Break command during the first 60 seconds of system startup. The break command is issued using the Break key or Break key-combination. (The default Break key combination is Ctrl-C, but this may be configured differently on your system.) Issuing the break command interrups the boot sequence and brings you into ROM monitor mode.

Another method for entering ROM monitor mode is to set the configuration register so that the router automatically enters ROM monitor mode when it boots. For information about setting the configuration register value, see the "Rebooting" chapter in this book.

ROM monitor mode uses an angle bracket (>) as the command line prompt. On some Cisco devices the default ROM monitor prompt is `rommon >`. A list of ROM monitor commands is displayed when you enter the **?** command or **help** command. The following example shows how this list of commands may appear:

```
User break detected at location 0x8162ac6\C
rommon 1 > ?
alias               set and display aliases command
boot                boot up an external process
break               set/show/clear the breakpoint
confreg             configuration register utility
cont                continue executing a downloaded image
context             display the context of a loaded image
cpu_card_type       display CPU card type
dev                 list the device table
dir                 list files in file system
dis                 disassemble instruction stream
frame               print out a selected stack frame
help                monitor builtin command help
history             monitor command history
meminfo             main memory information
repeat              repeat a monitor command
reset               system reset
set                 show all monitor variables
stack               produce a stack trace
```

```
sync                write monitor environment to NVRAM
sysret              print out info from last system return
unalias             unset an alias
unset               unset a monitor variable
rommon 2>
```

The list of available commands will vary depending on the software image and platform you are using. Some versions of ROMMON will display a list of commands in a pre-aliased format such as the following:

```
> ?
$ state      Toggle cache state (? for help)
B [filename] [TFTP Server IP address | TFTP Server Name]
             Load and execute system image from ROM or from TFTP server
C [address]  Continue execution [optional address]
D /S M L V   Deposit value V of size S into location L with modifier M
E /S M L     Examine location L with size S with modifier M
G [address]  Begin execution
H            Help for commands
I            Initialize
K            Stack trace
L [filename] [TFTP Server IP address | TFTP Server Name]
             Load system image from ROM or from TFTP server, but do not
             begin execution
O            Show configuration register option settings
P            Set the break point
S            Single step next instruction
T function   Test device (? for help)
Deposit and Examine sizes may be B (byte), L (long) or S (short).
Modifiers may be R (register) or S (byte swap).
Register names are: D0-D7, A0-A6, SS, US, SR, and PC
```

To exit ROM monitor mode, use the **continue** command or **C** command alias; this will restart the booting process.

For more information on ROM monitor mode characteristics (including using aliases for commands) and using ROM monitor mode, see the "Rebooting" chapter in this document.

Summary of Main Cisco IOS Command Modes

Table 3 summarizes the main command modes used in the Cisco IOS CLI. For a complete list of configuration modes, see the "Cisco IOS Command Modes" appendix in this book.

Table 3 Summary of the Main Cisco IOS Command Modes

Command Mode	Access Method	Prompt	Exit Method
User EXEC	Log in.	`Router>`	Use the **logout** command.
Privileged EXEC	From user EXEC mode, use the **enable** EXEC command.	`Router#`	To exit to user EXEC mode, use the **disable** command. To enter global configuration mode, use the **configure terminal** privileged EXEC command.

Table 3 Summary of the Main Cisco IOS Command Modes (continued)

Command Mode	Access Method	Prompt	Exit Method
Global configuration	From privileged EXEC mode, use the **configure terminal** privileged EXEC command.	`Router(config)#`	To exit to privileged EXEC mode, use the **end** command or press **Ctrl-Z**. To enter interface configuration mode, use the **interface** configuration command.
Interface configuration	From global configuration mode, enter by specifying an interface with an **interface** command.	`Router(config-if)#`	To exit to global configuration mode, use the **exit** command. To exit to privileged EXEC mode, use the **end** command or press **Ctrl-Z**. To enter subinterface configuration mode, specify a subinterface with the **interface** command.
Subinterface configuration	From interface configuration mode, specify a subinterface with an **interface** command. (The availability of this mode is dependent on your platform.)	`Router(config-subif)#`	To exit to global configuration mode, use the **exit** command. To exit to privileged EXEC mode, use the **end** command or press **Ctrl-Z**.

ROM monitor	From privileged EXEC mode, use the **reload** EXEC command. Press the Break key during the first 60 seconds while the system is booting.	`>`
		or
		`boot>`
		or
		`rommon >`
			If you entered ROM monitor mode by interrupting the loading process, you can exit ROM monitor and resume loading by using the **continue** command or the **C** command alias.

Cisco IOS CLI Task List

To familiarize yourself with the features of the Cisco IOS CLI, perform any of the tasks described in the following sections:

- Getting Context-Sensitive Help
- Using the no and default Forms of Commands
- Using Command History
- Using CLI Editing Features and Shortcuts
- Searching and Filtering CLI Output

Getting Context-Sensitive Help

Entering a question mark (**?**) at the system prompt displays a list of commands available for each command mode. You also can get a list of the arguments and keywords available for any command with the context-sensitive help feature.

To get help specific to a command mode, a command name, a keyword, or an argument, use any of the following commands:

Command	Purpose
(*prompt*)# **help**	Displays a brief description of the help system.
(*prompt*)# *abbreviated-command-entry*<**?**>	Lists commands in the current mode that begin with a particular character string.
(*prompt*)# *abbreviated-command-entry*<**Tab**>	Completes a partial command name.
(*prompt*)# **?**	Lists all commands available in the command mode.
(*prompt*)# *command* **?**	Lists the available syntax options (arguments and keywords) for the command.
(*prompt*)# *command keyword* **?**	Lists the next available syntax option for the command.

Note that the system prompt will vary depending on which configuration mode you are in.

When using context-sensitive help, the space (or lack of a space) before the question mark (**?**) is significant. To obtain a list of commands that begin with a particular character sequence, type in those characters followed immediately by the question mark (**?**). Do not include a space. This form of help is called *word help,* because it completes a word for you. For more information, see the "Completing a Partial Command Name" section later in this chapter.

To list keywords or arguments, enter a question mark (**?**) in place of a keyword or argument. Include a space before the **?**. This form of help is called *command syntax help,* because it shows you which keywords or arguments are available based on the command, keywords, and arguments you already have entered.

You can abbreviate commands and keywords to the number of characters that allow a unique abbreviation. For example, you can abbreviate the **configure terminal** command to **config t**. Because the abbreviated form of the command is unique, the router will accept the abbreviated form and execute the command.

Entering the **help** command (available in any command mode) will provide the following description of the help system:

```
Router# help
Help may be requested at any point in a command by entering
a question mark '?'. If nothing matches, the help list will
be empty and you must back up until entering a '?' shows the
available options.
Two styles of help are provided:
1. Full help is available when you are ready to enter a
   command argument (e.g. 'show ?') and describes each possible
   argument.
2. Partial help is provided when an abbreviated argument is entered
   and you want to know what arguments match the input
   (e.g. 'show pr?'.)
```

As described in the **help** command output, you can use the question mark (**?**) to complete a partial command name (partial help), or to obtain a list of arguments or keywords that will complete the current command.

The following example illustrates how the context-sensitive help feature enables you to create an access list from configuration mode.

Enter the letters **co** at the system prompt followed by a question mark (**?**). Do not leave a space between the last letter and the question mark. The system provides the commands that begin with **co**.

```
Router# co?
configure  connect  copy
```

Enter the **configure** command followed by a space and a question mark to list the keywords for the command and a brief explanation:

```
Router# configure ?
  memory    Configure from NV memory
  network   Configure from a TFTP network host
  overwrite-network  Overwrite NV memory from TFTP network host
  terminal  Configure from the terminal
  <cr>
```

The <cr> symbol ("cr" stands for carriage return) appears in the list to indicate that one of your options is to press the Return or Enter key to execute the command, without adding any additional keywords. In this example, the output indicates that your options for the configure command are **configure memory** (configure from NVRAM), **configure network** (configure from a file on the network), **configure overwrite-network** (configure from a file on the network and replace the file in NVRAM), or **configure terminal** (configure manually from the terminal connection). For most commands, the <cr> symbol is used to indicate that you can execute the command with the syntax you have already entered. However, the configure command is a special case, as the CLI will prompt you for the missing syntax:

```
Router# configure
Configuring from terminal, memory, or network [terminal]? terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

The default response for the ? prompt is indicated in the CLI output by a bracketed option at the end of the line. In the preceding example, pressing the Enter (or Return) key is equivalent to typing in the word "terminal."

To skip the prompting, enter the **configure terminal** command to enter global configuration mode:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

The CLI provides error isolation in the form of an error indicator, a caret symbol (^). The ^ symbol appears at the point in the command string where the user has entered incorrect or unrecognized command syntax. For example, the caret symbol in the following output shows the letter that was mistyped in the command:

```
Router# configure termiMal
                    ^
% Invalid input detected at '^' marker.

Router#
```

Note that an error message (indicated by the % symbol) is printed to the screen to alert you to the error marker.

Enter the **access-list** command followed by a space and a question mark to list the available options for the command:

```
c3660-2(config)#access-list ?
  <1-99>            IP standard access list
  <100-199>         IP extended access list
  <1100-1199>       Extended 48-bit MAC address access list
  <1300-1999>       IP standard access list (expanded range)
  <200-299>         Protocol type-code access list
```

```
  <2000-2699>       IP extended access list (expanded range)
  <700-799>         48-bit MAC address access list
  dynamic-extended  Extend the dynamic ACL abolute timer
  rate-limit        Simple rate-limit specific access list
```

The two numbers within the angle brackets represent an inclusive range. Enter the access list number **99** and then enter another question mark to see the arguments that apply to the keyword and brief explanations:

```
Router(config)# access-list 99 ?
  deny    Specify packets to reject
  permit  Specify packets to forward
```

Enter the **deny** argument followed by a question mark (**?**) to list additional options:

```
Router(config)# access-list 99 deny ?
  A.B.C.D  Address to match
```

Generally, uppercase letters represent variables (arguments). Enter the IP address followed by a question mark (**?**) to list additional options:

```
Router(config)# access-list 99 deny 172.31.134.0 ?
  A.B.C.D  Mask of bits to ignore
  <cr>
```

In this output, A.B.C.D indicates that use of a wildcard mask is allowed. The wildcard mask is a method for matching IP addresses or ranges of IP addresses. For example, a wildcard mask of 0.0.0.255 matches any number in the range from 0 to 255 that appears in the fourth octet of an IP address.

Enter the wildcard mask followed by a question mark (**?**) to list further options.

```
Router(config)# access-list 99 deny 172.31.134.0 0.0.0.255 ?
<cr>
```

The <cr> symbol by itself indicates there are no more keywords or arguments. Press Enter (or Return) to execute the command.

```
Router(config)# access-list 99 deny 172.31.134.0 0.0.0.255
```

The system adds an entry to access list 99 that denies access to all hosts on subnet 172.31.134.0, while ignoring bits for IP addresses that end in 0 to 255.

Displaying All User EXEC Commands

To configure the current session to display the full set of user EXEC commands, use the following command in EXEC mode (user EXEC or privileged EXEC):

Command	Purpose
`Router# terminal full-help`	Configures this session to provide help for the full set of user-level commands.

The system administrator can also configure the system to always display full help for connections made to a particular line using the **full-help** line configuration command.

The **full-help** and **terminal full-help** commands enable the displaying of all help messages available in user EXEC mode when the **show ?** command is executed.

The following example is output for the **show ?** command with **terminal full-help** disabled and then enabled:

```
Router> terminal no full-help
Router> show ?

  bootflash  Boot Flash information
  calendar   Display the hardware calendar
  clock      Display the system clock
  context    Show context information
  dialer     Dialer parameters and statistics
  history    Display the session command history
  hosts      IP domain-name, lookup style, nameservers, and host table
  isdn       ISDN information
  kerberos   Show Kerberos Values
  modemcap   Show Modem Capabilities database
  ppp        PPP parameters and statistics
  rmon       rmon statistics
  sessions   Information about Telnet connections
  snmp       snmp statistics
  terminal   Display terminal configuration parameters
  users      Display information about terminal lines
  version    System hardware and software status

Router> terminal full-help
Router> show ?

  access-expression  List access expression
  access-lists       List access lists
  aliases            Display alias commands
  apollo             Apollo network information
  appletalk          AppleTalk information
  arp                ARP table
  async              Information on terminal lines used as router interfaces
  bootflash          Boot Flash information
  bridge             Bridge Forwarding/Filtering Database [verbose]
  bsc                BSC interface information
  bstun              BSTUN interface information
  buffers            Buffer pool statistics
  calendar           Display the hardware calendar
  cdp                CDP information
  clns               CLNS network information
  clock              Display the system clock
  cls                DLC user information
  cmns               Connection-Mode networking services (CMNS) information
  .
  .
  .
  x25                X.25 information
```

Using the no and default Forms of Commands

Almost every configuration command has a **no** form. In general, use the **no** form to disable a feature or function. Use the command without the **no** keyword to reenable a disabled feature or to enable a feature that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, use the **no ip routing** form of the **ip routing** command. To reenable it, use the plain **ip routing** form. The Cisco IOS software command reference publications describe the function of the **no** form of the command whenever a **no** form is available.

Many CLI commands also have a **default** form. By issuing the command **default** *command-name*, you can configure the command to its default setting. The Cisco IOS software command reference publications describe the function of the **default** form of the command when the **default** form performs a different function than the plain and **no** forms of the command. To see what default commands are available on your system, enter **default ?** in the appropriate command mode.

Using Command History

The Cisco IOS CLI provides a history or record of commands that you have entered. This feature is particularly useful for recalling long or complex commands or entries, including access lists. To use the command history feature, perform any of the tasks described in the following sections:

- Setting the Command History Buffer Size
- Recalling Commands
- Disabling the Command History Feature

Setting the Command History Buffer Size

By default, the system records ten command lines in its history buffer. To set the number of command lines that the system will record during the current terminal session, use the following command in EXEC mode:

Command	Purpose
`Router# terminal history [size number-of-lines]`	Enables the command history feature for the current terminal session.

The **terminal no history size** command resets the number of lines saved in the history buffer to the default of ten lines.

To configure the number of command lines the system will record for all sessions on a particular line, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# history [size number-of-lines]`	Enables the command history feature.

Recalling Commands

To recall commands from the history buffer, use one of the following commands or key combinations:

Command or Key Combination	Purpose
Ctrl-P or the **Up Arrow** key.[1]	Recalls commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.
Ctrl-N or the **Down Arrow** key.[1]	Returns to more recent commands in the history buffer after recalling commands with Ctrl-P or the Up Arrow key. Repeat the key sequence to recall successively more recent commands.
`Router> show history`	While in EXEC mode, lists the last several commands entered.

1. The arrow keys function only on ANSI-compatible terminals .

Disabling the Command History Feature

The command history feature is automatically enabled. To disable it during the current terminal session, use the following command in EXEC mode:

Command	Purpose
`Router> terminal no history`	Disables command history for the current session.

To configure a specific line so that the command history feature is disabled, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# no history`	Disables command history for the line.

Using CLI Editing Features and Shortcuts

A variety of shortcuts and editing features are enabled for the Cisco IOS CLI. The following subsections describe these features:

- Moving the Cursor on the Command Line
- Completing a Partial Command Name
- Recalling Deleted Entries
- Editing Command Lines that Wrap
- Deleting Entries
- Continuing Output at the --More-- Prompt
- Redisplaying the Current Command Line
- Transposing Mistyped Characters
- Controlling Capitalization

- Designating a Keystroke as a Command Entry
- Disabling and Reenabling Editing Features

Moving the Cursor on the Command Line

Table 4 shows the key combinations or sequences you can use to move the cursor around on the command line to make corrections or changes. Ctrl indicates the Control key, which must be pressed simultaneously with its associated letter key. Esc indicates the Escape key, which must be pressed first, followed by its associated letter key. Keys are not case sensitive. Many letters used for CLI navigation and editing were chosen to provide an easy way of remembering their functions. In Table 4 characters are bolded in the "Function Summary" column to indicate the relation between the letter used and the function.

Table 4 Key Combinations Used to Move the Cursor

Keystrokes	Function Summary	Function Details
Left Arrow or **Ctrl-B**	**B**ack character	Moves the cursor one character to the left. When you enter a command that extends beyond a single line, you can press the Left Arrow or Ctrl-B keys repeatedly to scroll back toward the system prompt and verify the beginning of the command entry, or you can press the Ctrl-A key combination.
Right Arrow or **Ctrl-F**	**F**orward character	Moves the cursor one character to the right.
Esc, **B**	**B**ack word	Moves the cursor back one word.
Esc, **F**	**F**orward word	Moves the cursor forward one word.
Ctrl-A	Beginning of line	Moves the cursor to the beginning of the line.
Ctrl-E	**E**nd of line	Moves the cursor to the end of the command line.

Completing a Partial Command Name

If you cannot remember a complete command name, or if you want to reduce the amount of typing you have to perform, enter the first few letters of the command, then press the Tab key. The command line parser will complete the command if the string entered is unique to the command mode. If your keyboard does not have a Tab key, press **Ctrl-I** instead.

The CLI will recognize a command once you have entered enough characters to make the command unique. For example, if you enter **conf** in privileged EXEC mode, the CLI will be able to associate your entry with the **configure** command, because only the **configure** command begins with **conf**.

In the following example the CLI recognizes the unique string for privileged EXEC mode of **conf** when the Tab key is pressed:

```
Router# conf<Tab>
Router# configure
```

When you use the command completion feature the CLI displays the full command name. The command is not executed until you use the Return or Enter key. This way you can modify the command if the full command was not what you intended by the abbreviation. If you enter a set of characters that could indicate more than one command, the system beeps to indicate that the text string is not unique.

If the CLI can not complete the command, enter a question mark (**?**) to obtain a list of commands that begin with that set of characters. Do not leave a space between the last letter you enter and the question mark (**?**).

For example, entering **co?** will list all commands available in the current command mode:

```
Router# co?
configure  connect  copy
Router# co
```

Note that the characters you enter before the question mark are reprinted to the screen to allow you to complete the command entry.

Deleting Entries

Use any of the following keys or key combinations to delete command entries if you make a mistake or change your mind:

Keystrokes	Purpose
Delete or **Backspace**	Deletes the character to the left of the cursor.
Ctrl-D	Deletes the character at the cursor.
Ctrl-K	Deletes all characters from the cursor to the end of the command line.
Ctrl-U or **Ctrl-X**	Deletes all characters from the cursor to the beginning of the command line.
Ctrl-W	Deletes the word to the left of the cursor.
Esc, **D**	Deletes from the cursor to the end of the word.

Recalling Deleted Entries

The CLI stores commands or keywords that you delete in a history buffer. Only character strings that begin or end with a space are stored in the buffer; individual characters that you delete (using Backspace or Ctrl-D) are not stored. The buffer stores the last ten items that have been deleted using Ctrl-K, Ctrl-U, or Ctrl-X. To recall these items and paste them in the command line, use the following key combinations:

Keystrokes	Purpose
Ctrl-Y	Recalls the most recent entry in the buffer (press keys simultaneously).
Esc, **Y**	Recalls the previous entry in the history buffer (press keys sequentially).

Note that the Esc, Y key sequence will not function unless you press the Ctrl-Y key combination first. If you press Esc, Y more than ten times, you will cycle back to the most recent entry in the buffer.

Editing Command Lines that Wrap

The CLI provides a wrap-around feature for commands that extend beyond a single line on the screen. When the cursor reaches the right margin, the command line shifts ten spaces to the left. You cannot see the first ten characters of the line, but you can scroll back and check the syntax at the beginning of the command. To scroll back, press Ctrl-B or the left arrow key repeatedly until you scroll back to the beginning of the command entry, or press Ctrl-A to return directly to the beginning of the line.

In the following example, the **access-list** command entry extends beyond one line. When the cursor first reaches the end of the line, the line is shifted ten spaces to the left and redisplayed. The dollar sign ($) indicates that the line has been scrolled to the left. Each time the cursor reaches the end of the line, the line is again shifted ten spaces to the left.

```
Router(config)# access-list 101 permit tcp 172.31.134.5 255.255.255.0 172.31.1
Router(config)# $ 101 permit tcp 172.31.134.5 255.255.255.0 172.31.135.0 255.25
Router(config)# $t tcp 172.31.134.5 255.255.255.0 172.31.135.0 255.255.255.0 eq
Router(config)# $31.134.5 255.255.255.0 172.31.135.0 255.255.255.0 eq 45
```

When you have completed the entry, press **Ctrl-A** to check the complete syntax before pressing the Return key to execute the command. The dollar sign ($) appears at the end of the line to indicate that the line has been scrolled to the right:

```
Router(config)# access-list 101 permit tcp 172.31.134.5 255.255.255.0 172.31.1$
```

The Cisco IOS software assumes you have a terminal screen that is 80 columns wide. If you have a different screen-width, use the **terminal width** EXEC command to set the width of your terminal.

Use line wrapping in conjunction with the command history feature to recall and modify previous complex command entries. See the "Recalling Commands" section in this chapter for information about recalling previous command entries.

Continuing Output at the --More-- Prompt

When working with the Cisco IOS CLI, output often extends beyond the visible screen length. For cases where output continues beyond the bottom of the screen, such as with the output of many **?**, **show**, or **more** commands, the output is paused and a --More-- prompt is displayed at the bottom of the screen. To resume output, press the Return key to scroll down one line, or press the Spacebar to display the next full screen of output.

Tips If output is pausing on your screen, but you do not see the --More-- prompt, try entering a smaller value for the screen length using the **length** line configuration command or the **terminal length** EXEC command. Command output will not be paused if the **length** value is set to zero.

For information about filtering output from the --More-- prompt, see the "Searching and Filtering CLI Output" section in this chapter.

Redisplaying the Current Command Line

If you are entering a command and the system suddenly sends a message to your screen, you can easily recall your current command line entry. To redisplay the current command line (refresh the screen), use either of the following key combinations:

Keystrokes	Purpose
Ctrl-**L** or **Ctrl**-**R**	Redisplays the current command line.

Transposing Mistyped Characters

If you have mistyped a command entry, you can transpose the mistyped characters. To transpose characters, use the following key combination:

Keystrokes	Purpose
Ctrl-**T**	Transposes the character to the left of the cursor with the character located at the cursor.

Controlling Capitalization

You can capitalize or lowercase words or capitalize a set of letters with simple key sequences. Note, however, that Cisco IOS commands are generally case-insensitive, and are typically all in lowercase. To change the capitalization of commands, use any of the following key sequences:

Keystrokes	Purpose
Esc, **C**	Capitalizes the letter at the cursor.
Esc, **L**	Changes the word at the cursor to lowercase.
Esc, **U**	Capitalizes letters from the cursor to the end of the word.

Designating a Keystroke as a Command Entry

You can configure the system to recognize particular keystroke (key combination or sequence) as command aliases. In other words, you can set a keystroke as a shortcut for executing a command. To enable the system to interpret a keystroke as a command, use the either of the following key combinations before entering the command sequence:

Keystrokes	Purpose
Ctrl-**V** or **Esc**, **Q**	Configures the system to accept the following keystroke as a user-configured command entry (rather than as an editing command).

Disabling and Reenabling Editing Features

The editing features described in the previous sections were introduced in Cisco IOS Release 9.21, and are automatically enabled on your system. However, there may be some unique situations that could warrant disabling these editing features. For example, you may have scripts that conflict with editing functionality. To globally disable editing features, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# no editing`	Disables CLI editing features for a particular line.

To disable the editing features for the current terminal session, use the following command in EXEC mode:

Command	Purpose
`Router# terminal no editing`	Disables CLI editing features for the local line.

To reenable the editing features for the current terminal session, use the following command in EXEC mode:

Command	Purpose
`Router# terminal editing`	Enables the CLI editing features for the current terminal session.

To reenable the editing features for a specific line, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# editing`	Enables the CLI editing features.

Searching and Filtering CLI Output

The Cisco IOS CLI provides ways of searching through large amounts of command output and filtering output to exclude information you do not need. These features are enabled for **show** and **more** commands, which generally display large amounts of data.

Note **Show** and **more** commands are always entered in EXEC mode.

When output continues beyond what is displayed on your screen, the Cisco IOS CLI displays a --More-- prompt. Pressing Return displays the next line; pressing the Spacebar displays the next screen of output. The CLI String Search feature allows you to search or filter output from --More-- prompts.

Understanding Regular Expressions

A regular expression is a pattern (a phrase, number, or more complex pattern) the CLI String Search feature matches against **show** or **more** command output. Regular expressions are case sensitive and allow for complex matching requirements. Simple regular expressions include entries like `Serial`, `misses`, or `138`. Complex regular expressions include entries like `00210...`, `( is )`, or `[Oo]utput`.

A regular expression can be a single-character pattern or a multiple-character pattern. That is, a regular expression can be a single character that matches the same single character in the command output or multiple characters that match the same multiple characters in the command output. The pattern in the command output is referred to as a string. This section describes creating both single-character patterns and multiple-character patterns. It also discusses creating more complex regular expressions using multipliers, alternation, anchoring, and parentheses.

Single-Character Patterns

The simplest regular expression is a single character that matches the same single character in the command output. You can use any letter (A–Z, a–z) or digit (0–9) as a single-character pattern. You can also use other keyboard characters (such as ! or ~) as single-character patterns, but certain keyboard characters have special meaning when used in regular expressions. Table 5 lists the keyboard characters that have special meaning.

Table 5 Characters with Special Meaning

Character	Special Meaning
.	Matches any single character, including white space.
*	Matchers 0 or more sequences of the pattern.
+	Matches 1 or more sequences of the pattern.
?	Matches 0 or 1 occurrences of the pattern.
^	Matches the beginning of the string.
$	Matches the end of the string.
_ (underscore)	Matches a comma (,), left brace ({), right brace (}), left parenthesis (( ), right parenthesis ( )), the beginning of the string, the end of the string, or a space.

To use these special characters as single-character patterns, remove the special meaning by preceding each character with a backslash (\). The following examples are single-character patterns matching a dollar sign, an underscore, and a plus sign, respectively.

\$ _ \+

You can specify a range of single-character patterns to match against command output. For example, you can create a regular expression that matches a string containing one of the following letters: a, e, i, o, or u. Only one of these characters must exist in the string for pattern matching to succeed. To specify a range of single-character patterns, enclose the single-character patterns in square brackets (**[ ]**). For example, **[aeiou]** matches any one of the five vowels of the lowercase alphabet, while **[abcdABCD]** matches any one of the first four letters of the lower- or uppercase alphabet.

You can simplify ranges by entering only the endpoints of the range separated by a dash (-). Simplify the previous range as follows:

[a-dA-D]

To add a dash as a single-character pattern in your range, include another dash and precede it with a backslash:

[a-dA-D\-]

You can also include a right square bracket (]) as a single-character pattern in your range, as shown here:

[a-dA-D\-\]]

The previous example matches any one of the first four letters of the lower- or uppercase alphabet, a dash, or a right square bracket.

You can reverse the matching of the range by including a caret (^) at the start of the range. The following example matches any letter except the ones listed.

[^a-dqsv]

The following example matches anything except a right square bracket (]) or the letter d:

[^\]d]

Multiple-Character Patterns

When creating regular expressions, you can also specify a pattern containing multiple characters. You create multiple-character regular expressions by joining letters, digits, or keyboard characters that do not have special meaning. For example, a4% is a multiple-character regular expression. Put a backslash before the keyboard characters that have special meaning when you want to indicate that the character should be interpreted literally.

With multiple-character patterns, order is important. The regular expression **a4%** matches the character a followed by a 4 followed by a % sign. If the string does not have a4%, in that order, pattern matching fails. The multiple-character regular expression **a.** uses the special meaning of the period character to match the letter a followed by any single character. With this example, the strings ab, a!, or a2 are all valid matches for the regular expression.

You can remove the special meaning of the period character by putting a backslash in front of it. For example, when the expression **a\.** is used in the command syntax, only the string a. will be matched.

You can create a multiple-character regular expression containing all letters, all digits, all keyboard characters, or a combination of letters, digits, and other keyboard characters. For example, **telebit 3107 v32bis** is a valid regular expression.

Multipliers

You can create more complex regular expressions that instruct Cisco IOS software to match multiple occurrences of a specified regular expression. To do so, you use some special characters with your single-character and multiple-character patterns. Table 6 lists the special characters that specify "multiples" of a regular expression.

Table 6 Special Characters Used as Multipliers

Character	Description
*	Matches 0 or more single-character or multiple-character patterns.
+	Matches 1 or more single-character or multiple-character patterns.
?	Matches 0 or 1 occurrences of a single-character or multiple-character pattern.

The following example matches any number of occurrences of the letter a, including none:

a*

The following pattern requires that at least one letter a be in the string to be matched:

a+

The following pattern matches the string bb or bab:

ba?b

The following string matches any number of asterisks (*):

******

To use multipliers with multiple-character patterns, you enclose the pattern in parentheses. In the following example, the pattern matches any number of the multiple-character string ab:

(ab)*

As a more complex example, the following pattern matches one or more instances of alphanumeric pairs, but not none (that is, an empty string is not a match):

([A-Za-z][0-9])+

The order for matches using multipliers (*, +, or ?) is to put the longest construct first. Nested constructs are matched from outside to inside. Concatenated constructs are matched beginning at the left side of the construct. Thus, the regular expression matches A9b3, but not 9Ab3 because the letters are specified before the numbers.

Alternation

Alternation allows you to specify alternative patterns to match against a string. You separate the alternative patterns with a vertical bar (|). Exactly one of the alternatives can match the string. For example, the regular expression **codex|telebit** matches the string codex or the string telebit, but not both codex and telebit.

Anchoring

You can instruct Cisco IOS software to match a regular expression pattern against the beginning or the end of the string. That is, you can specify that the beginning or end of a string contain a specific pattern. You "anchor" these regular expressions to a portion of the string using the special characters shown in Table 7.

Table 7 Special Characters Used for Anchoring

Character	Description
^	Matches the beginning of the string.
$	Matches the end of the string.

For example, the regular expression **^con** matches any string that starts with con, and **$sole** matches any string that ends with sole.

In addition to indicating the beginning of a string, the ^ symbol can be used to indicate the logical function "not" when used in a bracketed range. For example, the expression **[^abcd]** indicates a range that matches any single letter, as long as it is not the letters a, b, c, or d.

Contrast these anchoring characters with the special character underscore (_). Underscore matches the beginning of a string (^), the end of a string ($), parentheses (( )) , space ( ), braces ({}), comma (,), or underscore (_). With the underscore character, you can specify that a pattern exist anywhere in the string. For example, **_1300_** matches any string that has 1300 somewhere in the string. The string 1300 can be preceded by or end with a space, brace, comma, or underscore. So, while {1300_ matches the regular expression **_1300_**, 21300 and 13000 do not.

Using the underscore character, you can replace long regular expression lists. For example, instead of specifying **^1300( ) ( )1300$ {1300, ,1300, {1300} ,1300, (1300** you can specify simply **_1300_**.

Parentheses for Recall

As shown in the "Multipliers" section, you use parentheses with multiple-character regular expressions to multiply the occurrence of a pattern. You can also use parentheses around a single- or multiple-character pattern to instruct the Cisco IOS software to remember a pattern for use elsewhere in the regular expression.

To create a regular expression that recalls a previous pattern, you use parentheses to indicate memory of a specific pattern and a backslash (\) followed by a digit to reuse the remembered pattern. The digit specifies the occurrence of a parentheses in the regular expression pattern. If you have more than one remembered pattern in your regular expression, then \1 indicates the first remembered pattern, and \2 indicates the second remembered pattern, and so on.

The following regular expression uses parentheses for recall:

a(.)bc(.)\1\2

This regular expression matches an a followed by any character (call it character no. 1), followed by bc followed by any character (character no. 2), followed by character no. 1 again, followed by character no. 2 again. So, the regular expression can match aZbcTZT. The software remembers that character no. 1 is Z and character no. 2 is T and then uses Z and T again later in the regular expression.

Searching and Filtering show Commands

To search **show** command output, use the following command in EXEC mode:

Command	Purpose
`Router# show` *any-command* `| begin` *regular-expression*	Begins unfiltered output of the **show** command with the first line that contains the regular expression.

Note Cisco IOS documentation generally uses the vertical bar to indicate a choice of syntax. However, to search the output of **show** and **more** commands, you will need to enter the pipe character (the vertical bar). In this section the pipe appears in bold (**|**) to indicate that you should enter this character.

To filter **show** command output, use one of the following commands in EXEC mode:

Command	Purpose
`Router# show` *any-command* `| exclude` *regular-expression*	Displays output lines that do not contain the regular expression.
`Router# show` *any-command* `| include` *regular-expression*	Displays output lines that contain the regular expression.

On most systems you can enter the Ctrl-Z key combination at any time to interrupt the output and return to EXEC mode. For example, you can enter the **show running-config | begin hostname** command to start the display of the running configuration file at the line containing the hostname setting, then use Ctrl-z when you get to the end of the information you are interested in.

Searching and Filtering more Commands

You can search **more** commands the same way you search **show** commands (**more** commands perform the same function as **show** commands). To search **more** command output, use the following command in EXEC mode:

Command	Purpose
`Router# more` *any-command* `| begin` *regular-expression*	Begins unfiltered output of a **more** command with the first line that contains the regular expression.

You can filter **more** commands the same way you filter **show** commands. To filter **more** command output, use one of the following commands in EXEC mode:

Command	Purpose
`Router# more` *any-command* `| exclude` *regular-expression*	Displays output lines that do not contain the regular expression.
`Router# more` *any-command* `| include` *regular-expression*	Displays output lines that contain the regular expression.

Searching and Filtering from the --More-- Prompt

You can search output from --More-- prompts. To search **show** or **more** command output from a --More-- prompt, use the following command in EXEC mode:

Command	Purpose
`-More-` `/`*regular-expression*	Begins unfiltered output with the first line that contains the regular expression.

You can filter output from --More-- prompts. However, you can only specify one filter for each command. The filter remains until the **show** or **more** command output finishes or until you interrupt the output (using Ctrl-Z or Ctrl-6). Therefore, you cannot add a second filter at a --More-- prompt if you already specified a filter at the original command or at a previous --More--prompt.

Note Searching and filtering are different functions. You can search command output using the **begin** keyword and specify a filter at the --More-- prompt for the same command.

To filter **show** or **more** command output at a --More-- prompt, use one of the following commands in EXEC mode:

Command	Purpose
`-More-` `-`*regular-expression*	Displays output lines that do not contain the regular expression.
`-More-` `+`*regular-expression*	Displays output lines that contain the regular expression.

Using the Cisco IOS CLI Examples

The following sections provide examples of using the CLI:

- Determining Command Syntax and Using Command History Example
- Searching and Filtering CLI Output Examples

Determining Command Syntax and Using Command History Example

The CLI provides error isolation in the form of an error indicator, a caret symbol (^). The ^ symbol appears at the point in the command string where you have entered an incorrect command, keyword, or argument.

In the following example, suppose you want to set the clock. Use context-sensitive help to determine the correct command syntax for setting the clock.

```
Router# clock ?
  set  Set the time and date
Router# clock
```

The help output shows that the **set** keyword is required. Determine the syntax for entering the time:

```
Router# clock set ?
hh:mm:ss   Current time
Router# clock set
```

Enter the current time:

```
Router# clock set 13:32:00
% Incomplete command.
```

The system indicates that you need to provide additional arguments to complete the command. Press **Ctrl-P** or the Up Arrow to automatically repeat the previous command entry. Then add a space and question mark (**?**) to reveal the additional arguments:

```
Router# clock set 13:32:00 ?
  <1-31>     Day of the month
  January    Month of the year
  February
  March
  April
  May
  June
  July
  August
  September
  October
  November
  December
```

Now you can complete the command entry:

```
Router# clock set 13:32:00 23 February 01
                                       ^
% Invalid input detected at '^' marker.
```

The caret symbol (^) and help response indicate an error at 01. To list the correct syntax, enter the command up to the point where the error occurred and then enter a question mark (**?**):

```
Router# clock set 13:32:00 23 February ?
  <1993-2035> Year
```

```
Router# clock set 13:32:00 23 February
```

Enter the year using the correct syntax and press Enter or Return to execute the command:

```
Router# clock set 13:32:00 23 February 2001
```

Searching and Filtering CLI Output Examples

The following is partial sample output of the **more nvram:startup-config | begin** EXEC command that begins unfiltered output with the first line that contain the regular expression `ip` . At the --More-- prompt, the user specifies a filter to exclude output lines that contain the regular expression `ip` .

```
Router# more nvram:startup-config | begin ip
ip subnet-zero
ip domain-name cisco.com
ip name-server 198.92.30.32
ip name-server 171.69.2.132
!
isdn switch-type primary-5ess
.
.
.
interface Ethernet1
 ip address 5.5.5.99 255.255.255.0
 --More--
-ip
filtering...
 media-type 10BaseT
!
interface Serial0:23
 encapsulation frame-relay
 no keepalive
 dialer string 4001
 dialer-group 1
 isdn switch-type primary-5ess
 no fair-queue
```

The following is partial sample output of the **more nvram:startup-config | include** command. It only displays lines that contain the regular expression `ip` .

```
Router# more nvram:startup-config | include ip
ip subnet-zero
ip domain-name cisco.com
ip name-server 198.92.30.32
ip name-server 171.69.2.132
```

The following is partial sample output of the **more nvram:startup-config | exclude** command. It excludes lines that contain the regular expression `service`. At the --More-- prompt, the user specifies a filter with the regular expression `Dialer1` . Specifying this filter resumes the output with the first line that contains `Dialer1` .

```
Router# more nvram:startup-config | exclude service
!
version 12.2
!
hostname router
!
boot system flash
no logging buffered
!
```

```
ip subnet-zero
ip domain-name cisco.com
.
.
.
--More--
/Dialer1
filtering...
interface Dialer1
 no ip address
 no ip directed-broadcast
 dialer in-band
 no cdp enable
```

The following is partial sample output of the **show interface** EXEC command with an output search specified. The use of the keywords **begin Ethernet** after the pipe begins unfiltered output with the first line that contains the regular expression `Ethernet` . At the --More-- prompt, the user specifies a filter that displays only the lines that contain the regular expression `Serial` .

```
Router# show interface | begin Ethernet
Ethernet0 is up, line protocol is up
Hardware is Lance, address is 0060.837c.6399 (bia 0060.837c.6399)
  Description: ip address is 172.1.2.14 255.255.255.0
  Internet address is 172.1.2.14/24
.
.
.
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
--More--
+Serial
filtering...
Serial1 is up, line protocol is up
Serial2 is up, line protocol is up
Serial3 is up, line protocol is down
Serial4 is down, line protocol is down
Serial5 is up, line protocol is up
Serial6 is up, line protocol is up
Serial7 is up, line protocol is up
```

The following is partial sample output of the **show buffers | exclude** command. It excludes lines that contain the regular expression `0 misses`. At the --More-- prompt, the user specifies a search that continues the filtered output beginning with the first line that contains `Serial0`.

```
Router# show buffers | exclude 0 misses

Buffer elements:
     398 in free list (500 max allowed)
Public buffer pools:
Small buffers, 104 bytes (total 50, permanent 50):
     50 in free list (20 min, 150 max allowed)
     551 hits, 3 misses, 0 trims, 0 created
Big buffers, 1524 bytes (total 50, permanent 50):
     49 in free list (5 min, 150 max allowed)
Very Big buffers, 4520 bytes (total 10, permanent 10):
.
.
.
Huge buffers, 18024 bytes (total 0 permanent 0):
     0 in free list (0 min, 4 max allowed)
--More--
/Serial0
filtering...
```

```
Serial0 buffers, 1543 bytes (total 64, permanent 64):
     16 in free list (0 min, 64 max allowed)
     48 hits, 0 fallbacks
```

The following is partial sample output of the **show interface | include** command. The use of the **include (is)** keywords after the pipe (|) causes the command to display only lines that contain the regular expression `( is )` . The parenthesis force the inclusion of the spaces before and after `is`. Use of the parenthesis ensures that only lines containing `is` with a space both before and after it will be included in the output (excluding from the search, for example, words like "disconnect").

```
router# show interface | include ( is )
ATM0 is administratively down, line protocol is down
  Hardware is ATMizer BX-50
Dialer1 is up (spoofing), line protocol is up (spoofing)
  Hardware is Unknown
  DTR is pulsed for 1 seconds on reset
Ethernet0 is up, line protocol is up
  Hardware is Lance, address is 0060.837c.6399 (bia 0060.837c.6399)
  Internet address is 172.21.53.199/24
Ethernet1 is up, line protocol is up
  Hardware is Lance, address is 0060.837c.639c (bia 0060.837c.639c)
  Internet address is 5.5.5.99/24
Serial0:0 is down, line protocol is down
  Hardware is DSX1
.
.
.
 --More--
```

At the --More-- prompt, the user specifies a search that continues the filtered output beginning with the first line that contains `Serial0:13` :

```
/Serial0:13
filtering...
Serial0:13 is down, line protocol is down
  Hardware is DSX1
  Internet address is 11.0.0.2/8
     0 output errors, 0 collisions, 2 interface resets
  Timeslot(s) Used:14, Transmitter delay is 0 flag
```

Using AutoInstall and Setup

The Cisco IOS software includes two features that simplify or automate the configuration of Cisco devices. AutoInstall allows a network manager to load configuration files onto new Cisco devices automatically. Setup is a Cisco IOS software feature that guides a user through the first-time configuration of a Cisco device. This chapter describes AutoInstall, Setup, and provides a brief summary of external configuration applications. It include the following sections:

- Using AutoInstall
- Using Setup
- Using Configuration Applications

For a complete description of the **setup** command, refer to the Setup command reference page in the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Note This chapter uses quotation marks around file names to improve readability. Also, all instances of the term "router" in this chapter should be assumed to apply to any Cisco device that supports Cisco IOS Release 12.2.

Using AutoInstall

This section provides the following information about the AutoInstall feature in Cisco IOS software:

- Understanding AutoInstall
- AutoInstall Configuration Task List
- Monitoring and Completing the AutoInstall Process
- AutoInstall Configuration Examples

AutoInstall allows you to connect a new router to the network, turn on the new router, and have it configured automatically from a preexisting configuration file. This process was designed to facilitate the centralized management of router installation.

The AutoInstall process begins any time a Cisco IOS software-based device is turned on and a valid configuration file is not found in nonvolatile random-access memory (NVRAM). A configuration file is typically not available when a router is turned on for the first time, or when the configuration file has been manually deleted from NVRAM.

Note To configure a new router manually, connect directly to the console port and ensure that the router is not connected to the network via any of the interface ports before you turn on the router. It may take several minutes for the router boot software to determine that AutoInstall is not connected to the network. See the "Using Setup" section later in this chapter for information on configuring a new router manually.

The following sections describes the options available to prepare your network for the AutoInstall process. Network set up for AutoInstall can also be performed with network management applications such as the AutoInstall Manager in CiscoWorks software. For details on other ways to set up the AutoInstall process, refer to the documentation for your application, or search for Network Management information on Cisco.com.

Understanding AutoInstall

There are two basic approaches to preparing your network for AutoInstall. One approach is to create a minimal configuration file that provides just enough configuration information to allow you to Telnet to the new router and configure it manually. The other approach is to create a host-specific configuration file for each new router containing all of the necessary configuration information. In each case, the configuration file should be created and stored on a TFTP server on the network prior to connecting the new router.

Before the new router can attempt to download a configuration file, however, it must acquire an IP address. This means that a service must be available on the network to provide an IP address to the new router. Your choice of service will determine which interface port on the new router should be connected to the network.

AutoInstall is supported over Ethernet, Token Ring, and FDDI interfaces for LANs and serial interfaces with High-Level Data Link Control (HDLC) encapsulation or serial interfaces with Frame Relay encapsulation for WANs. If a LAN interface is used, AutoInstall will attempt to acquire an IP address for the attached interface using Dynamic Host Configuration Protocol (DHCP) requests, Bootstrap Protocol (BOOTP) requests, or Reverse Address Resolution Protocol (RARP) requests. If a serial interface with HDLC encapsulation is connected, AutoInstall will attempt to acquire an IP address for the attached interface using Serial Line Address Resolution Protocol (SLARP). Table 8 summarizes this information.

Table 8 Protocols Used for IP Address Acquisition in AutoInstall

Interface Type	Protocol Used for AutoInstall
Ethernet, Token Ring, or FDDI interface	DHCP, BOOTP, or RARP
Serial interface using HDLC	SLARP
Serial interface using Frame Relay	BOOTP

Note Cisco IOS Release 12.2 replaces the use of BOOTP with DHCP for LAN interfaces in AutoInstall. DHCP (defined in RFC 2131) is based on BOOTP, which provides the framework for passing configuration information to hosts on a TCP/IP network. DHCP adds the capability of automatic

allocation of reusable network addresses and additional configuration options. Because AutoInstall supports RFC 1534 (*Interoperation Between DHCP and BOOTP*), BOOTP servers can provide limited responses to DHCP requests sent during the AutoInstall process. Likewise, those routing devices using BOOTP requests can be serviced by DHCP servers. This interoperability maintains backward compatibility for your network, and allows for a seamless transition to the newer DHCP-based AutoInstall process. For further information on BOOTP-based AutoInstall, please see RFC 951, *The Bootstrap Protocol (BOOTP)*, and the *Cisco IOS Configuration Fundamentals Configuration Guide* for Release 12.1 or earlier releases.

When the AutoInstall process begins, the new router will send DHCP, BOOTP, and RARP requests out any attached interfaces. AutoInstall will use the first available method for configuration. If all LAN interface requests fail, AutoInstall will attempt to configure an available serial interface using SLARP.

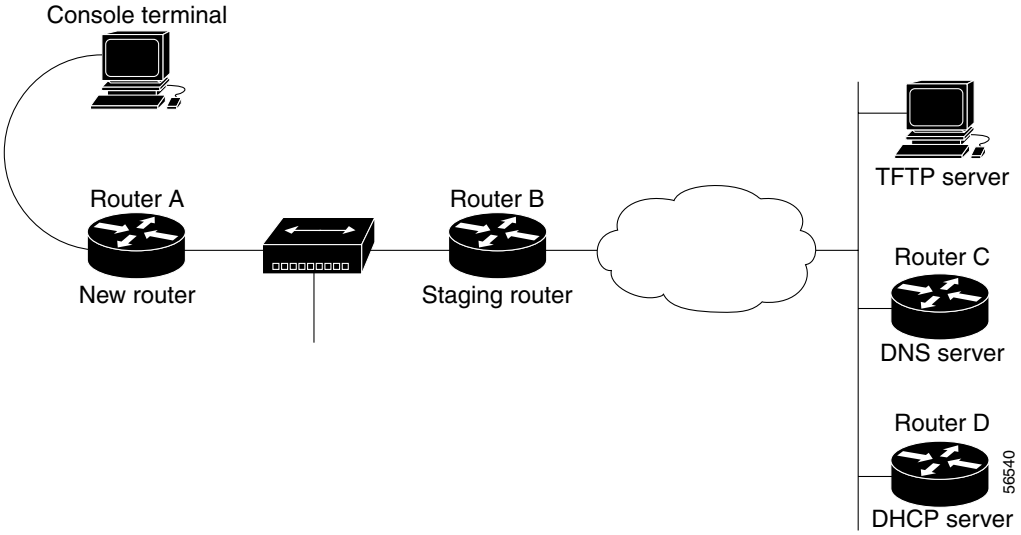
Note Of Token Ring interfaces, only those that set ring speed with physical jumpers support AutoInstall. AutoInstall does not work with Token Ring interfaces for which the ring speed must be set with software configuration commands. If the ring speed is not set, the interface is set to shutdown mode.

In addition to a TFTP server, and a DHCP, BOOTP, RARP, or SLARP server, you may need to configure other elements in your network to enable AutoInstall, as follows:

- If the new router is not directly connected to the device providing the IP address resolution service, you will need to configure the intermediate router to forward requests. In this chapter, we refer to this intermediate router as the *staging router*. For serial interfaces, a directly attached router providing a SLARP service is required.
- If you wish to enable the new router to download a host-specific configuration file, you can configure a Domain Name System (DNS) server on the LAN network to provide the new router with its hostname. In this case, an IP address-to-hostname mapping for the new router must be added to the DNS database file prior to beginning AutoInstall. Note that a DNS server is not necessary if you configure a DHCP server to provide a hostname for the new router and to provide the IP address of the TFTP server to the new router.

Figure 2 shows a hypothetical network topology that utilizes these various elements and shows an example of the AutoInstall process.

Figure 2 Hypothetical Network Topology for AutoInstall over an Ethernet Interface



The following steps outline an example of the AutoInstall process that would be used for the topology in Figure 2:

1. Router A (the new router) sends a DHCP request out of its attached Ethernet 0 (E0) interface.
2. Router B (the staging router) forwards the request to Router D, which is running a DHCP service.
3. The DHCP server in Router D sends a reply back to Router A. The reply contains a temporary IP address for the E0 interface on Router A and the IP address of the TFTP server.
4. Router A sends a request for a network configuration file to the TFTP server using the address acquired in Step 3.
5. The network configuration file downloaded from the TFTP server does not contain an IP address to hostname mapping for Router A's new IP address, so Router A sends out a DNS request (forwarded by Router B) to acquire its new hostname.
6. The DNS server in Router C resolves the IP address of the new router to the hostname "rtr1" and sends this data to Router A.
7. Router A uses its hostname to send a unicast request to the TFTP server for the host-specific configuration file "rtr1-confg", using the address acquired in Step 3.
8. The "rtr1-confg" file is loaded as the running configuration for Router A. The new configuration contains a permanent IP address assignment, so Router A releases the leased IP address from the DHCP server (using a DHCPRELEASE message).

Note that these steps are simplified to give an impression of the process flow (for example, the DHCP request forwarding in steps 1 and 2 actually consist of multiple discover, offer, and request messages).

Note For more information on the Cisco implementation of DHCP, see the "Configuring DHCP" chapter of the Release 12.2 *Cisco IOS IP Configuration Guide*.

In this example, the TFTP, DNS, and DHCP services are provided by different devices. However, more than one service may be enabled on any particular device. For example, the Cisco Server Suite 1000 includes a DHCP/BOOTP server, a DNS server, and a TFTP server all on one platform. Note that you

can enable more than one service (such as a TFTP server and a DNS server) on a Cisco router. The example in Figure 2 also shows Cisco routers acting as DNS and DHCP servers, but these services can be provided by any standard workstation. For information on the configuration of this example, see the "Network Configuration for DHCP-Based AutoInstall Example" section on page 57.

Tip A network management tool can help immensely in implementing features like AutoInstall for complex networks. One such tool is the Cisco Network Registrar (CNR), which automates common tasks such as IP address assignment and maintenance to simplify and streamline network administration. CNR includes robust DNS and DCHP servers and utilities within the program. For more information about CNR, see http://www.cisco.com/go/cnr/.

Specifying a Staging Router

For those network topologies in which the servers used in the AutoInstall process are not on the same LAN segment as the new router, a staging router is typically needed.

For AutoInstall over serial interfaces, a staging router is always required, because the proper encapsulation must be configured on the interface that will be connected to the new router. Because the address of the staging router can not be specified to the new router, you should have a direct connection from the new router to the staging router.

The staging router must have an **ip helper** *address* command configured on the appropriate interface for each server to enable unicast requests from the staging router. For example, you may want to configure a helper address for the TFTP server, the DNS server, and the DHCP server on the staging router for AutoInstall over LAN interfaces. For AutoInstall over a Frame Relay encapsulated interface, the staging router will require a helper address for the TFTP server and a Frame Relay map pointing back to the new router. For more information, see the "Configuring a Staging Router" section on page 50.

For AutoInstall over HDLC-encapsulated serial interfaces using SLARP, the interface on the staging router must be configured with an IP address whose host portion has the value 1 or 2. AutoInstall over Frame Relay does not have this address constraint. Subnet masks of any size are supported. For more information, see the "AutoInstall over Serial Interfaces" section on page 48.

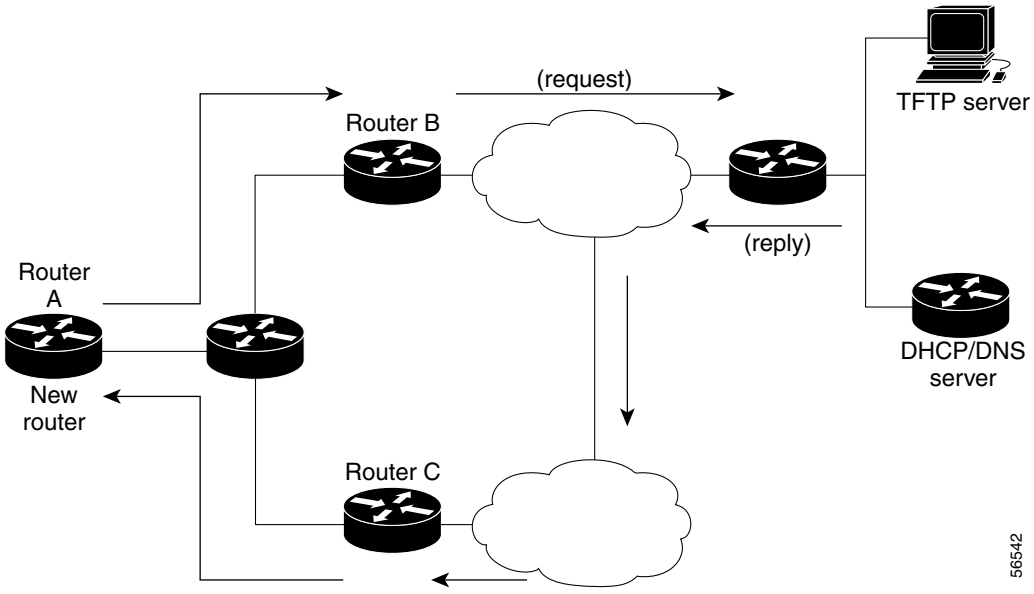
Specifying a Default Router

In some network topologies it may be necessary to specify the "default" router through which AutoInstall TFTP requests should be forwarded. The IP address of a default router can be specified as the value of the option 3 field of the DHCP response to the new router. For information on specifying the values for the option fields of DHCP responses, see the "Configuring a Cisco Router as a DHCP Server" section on page 53. For more information on the BOOTP and DHCP fields and options discussed in this chapter, refer to RFC 2132, *DHCP Option and BOOTP Vendor Extensions*.

Note In this section, the forwarding device is a router. However, any addressable device capable of forwarding IP packets can be specified as the default first hop.

Figure 3 shows an example of a network topology in which a default router needs to be specified.

Figure 3 Specifying a Default Router for AutoInstall Requests



In the example shown in Figure 3, there are multiple routers on the same LAN as the interface performing Autoinstall. Router C is the router forwarding the TFTP packets from the TFTP server to the new router, but only Router B is capable of forwarding the TFTP request from the new router to the TFTP server. In this case, the IP address of Router B should be configured as the default router, using option 3 in the DHCP configuration. The new router will interpret the value of option 3 (specified in the DHCP configuration) as the next-hop toward the TFTP server during Autoinstall. Option 3 will take effect only if information (hostname or IP address) for the TFTP server is also returned in the DHCP/BOOTP reply.

Note that option 3 cannot be used to specify the default router for DNS requests. For configuration details, see the "Specifying the Default Router on a Cisco DHCP Server" section on page 56.

Preparing a Configuration File

Your choice of configuration file will determine many aspects of how you set up the network for AutoInstall. There are three types of configuration files you can make available on a TFTP server:

- A host-specific configuration file, containing a full configuration for each new router ("*hostname*-confg" or "*hostname*.cfg", where *hostname* is a specific router name)
- A default configuration file, containing just enough configuration to allow you to Telnet to the new router and configure it manually ("router-confg", "router.cfg", or "ciscortr.cfg")
- A network configuration file, which allows you to specify IP addresses or hostnames for routers on the network ("network-confg" or "cisconet.cfg")

The syntax of the configuration file-name will depend on the host of the TFTP server. UNIX-based or DOS-based configuration files are saved using the 8.3 naming convention and use ".cfg". Any hostname longer than eight characters is truncated to eight characters. For example, a router with a hostname "australia" will be treated as "australi" and AutoInstall will attempt to download "australi.cfg".

Tip We recommend that configuration files intended for specific hosts be saved with the name "*hostname*-confg" or, if using a DOS-based TFTP server, "*hostname*.cfg", where *hostname* is the name of the intended routing device. The hostname specified in the configuration file should match the filename.

Default network configuration files should have IP address to hostname mappings (using **ip host ip** *address hostname* command line entries).

In general, AutoInstall will attempt to download "-confg" files first, then ".cfg" files. AutoInstall will attempt to download default configuration files in the following order:

- "network-confg"
- "cisconet.cfg"
- "router-confg"
- "router.cfg"
- "ciscortr.cfg"

The request cycle is repeated three times.

AutoInstall over LAN Interfaces

The process used for AutoInstall over an Ethernet, Token Ring, or FDDI interface, and is divided into two phases, which are described in the following sections:

- IP Address Acquisition for LAN Interfaces
- Configuration File Downloading for LAN Interfaces

IP Address Acquisition for LAN Interfaces

When the AutoInstall process begins, the new router will attempt to acquire an IP address for the connected interface.

The new router will send DHCP discover packets out all attached LAN interfaces to determine if a DHCP server is available. If an offer is returned from a DHCP server, the new router will act as a DHCP client and send a DHCP request. If more than one offer is returned, the first is used.

If a response is returned first from a BOOTP server, or a DHCP server is not available, the DHCP client in AutoInstall will use the BOOTP information to continue the AutoInstall process. Because DHCP is an extension to BOOTP, the DHCP client in AutoInstall can interpret BOOTP replies. Prior to Cisco IOS Release 12.1(5)T, the TFTP identifier (specified in the siaddr or sname field) and bootfile name (specified in the file field) in BOOTP server replies were ignored. If you have configured the TFTP identifier and bootfile name to be provided by the BOOTP server, this information can now be used by the DHCP AutoInstall process to perform unicast TFTP uploading of configuration files.

Note The MAC address of the new router must be mapped on a DHCP, BOOTP, or RARP server to an IP address for the new router prior to starting the AutoInstall process over a LAN interface.

If a DHCP server responds, any or all of the following information can be returned to the new router:

- The IP address (yiaddr) and subnet mask (option 1) to be assigned to the interface on the new router (the values in parenthesis in this section represent the field names for the packet as defined in the relevant RFCs). The following lines will be written to the configuration of the new router:

```
interface <type><number>
 ip address dhcp
```

- The address of the TFTP server (siaddr) to be used for AutoInstall requests.
- The name of the configuration file (file or option 67) to be requested from the TFTP server.
- The IP address of the TFTP server (option 150).
- The hostname of the TFTP server (option 66 or sname). Typically either the TFTP address or name is specified, not both. If only the name of the TFTP server is specified, a DNS server must be available to translate the name to an IP address.
- The IP addresses of up to two DNS name servers (option 6). You should configure this option to be returned from the DHCP server only if the DNS server is in the same LAN as the interface performing AutoInstall.
- The IP address of the staging router (option 3). This option is provided for those cases in which the TFTP server is not on the same LAN segment as the new router, or if the network topology requires the use of a specific router. The staging router address is used to specify which router the AutoInstall TFTP requests should be sent through (in other words, the "first hop" router). This staging router is also referred to as the "default" or "helper" router. Only one staging router can be specified.

The TFTP server IP address can be deduced from the following sources, from the highest priority to the lowest: the sname field, option 66, option 150, or the siaddr field. If only the sname or option 66 values are returned to the new router, a DNS server must be available to resolve the IP address.

Tip The most efficient method is to configure the IP address of the TFTP server (option 150) to be available on the DHCP server.

If a DHCP server is not available on the network and the sname or siaddr information is not available from a BOOTP server, the new router will use the BOOTP-based AutoInstall process described in the Release 12.1 *Cisco IOS Configuration Fundamentals Configuration Guide*.

After an IP address is assigned to the interface on the new router, the AutoInstall process will send a DNS request for the corresponding hostname. Likewise, if the new router is assigned a hostname, the AutoInstall process will send a DNS request for the corresponding IP address.

Using RARP for LAN-based AutoInstall

Although DHCP and BOOTP are the recommended protocols for LAN-based AutoInstall, Reverse Address Resolution Protocol (RARP) also remains supported. RARP requests send the local data-link (MAC) address to the RARP server and requests the IP address associated with that physical address. Use of RARP requires a RARP server on the same network segment as the router interface. RARP is typically used for devices (such as new routers or diskless workstations) that do not know their IP addresses when they boot. RARP relies on the presence of a RARP server with table entries of MAC-layer-to-IP address mappings. You should directly connect the new router to the device providing RARP services if using RARP for AutoInstall.

Configuration File Downloading for LAN Interfaces

After acquiring an IP address for the interface on the new router, the AutoInstall process will initiate attempts to download a configuration file. This is the second phase of the AutoInstall process for LAN interfaces.

The new router will automatically attempt to download a configuration file using one of the following methods:

- If the host-specific configuration file (bootfile) name was specified by the DHCP server, that specific filename is used in the TFTP request. The new router will make three unicast TFTP requests for the specified bootfile. If the unicast attempts fail, or if a TFTP server address was not provided, the new router will make three broadcast requests to any available TFTP server for the specified bootfile.
- If the specified bootfile can not be located, or the new router was not provided a specific bootfile name by the DHCP server, AutoInstall will unicast or broadcast TFTP requests for a default network configuration file. The files requested are first for "network-confg" then for "cisconet.cfg". The default network configuration file should have IP address to hostname mappings using **ip host ip** *address hostname* entries. If a command line entry for the IP address of the new router is not included in the configuration file, AutoInstall will attempt to resolve its hostname using a DNS query. If the new router can determine its hostname, a TFTP request will then be sent for the "*hostname*-confg" or "*hostname*.cfg" file. The *hostname* variable is replaced by the first eight characters of the new router's hostname. If the new router is unable to map its IP address to a hostname, AutoInstall will send TFTP requests for the default configuration file "router-confg" or "router.cfg."

Note The default configuration file ("router-confg" or "router.cfg") typically sets the hostname of the new router to "router" and provides just enough configuration information to allow further remote configuration by a system administrator.

Table 9 shows the type of TFTP requests made by the new router using AutoInstall. The type of TFTP request depends on the availability of the TFTP server name or address and the host-specific configuration filename.

Table 9 TFTP Request Types

TFTP Server Address Available?	Host-Specific Router Configuration Filename Available?	TFTP Request Method
Yes	Yes	Unicast request for the host-specific router configuration file to the specified TFTP server.
Yes	No	Unicast request for a default router configuration file to the specified TFTP server.
No	Yes	Broadcast request for the host-specific router configuration file to any available TFTP server
No	No	Broadcast request for a default router configuration file to any available TFTP server.

Essentially, if the TFTP address is known, the router can send a unicast TFTP request for a configuration file, and if the host-specific configuration filename is known, the router can request the host-specific configuration file from the TFTP server. If the TFTP address is not known, the router can send a broadcast TFTP request, and if the configuration filename is not known, the router can request the default configuration file.

The TFTP server address can be deduced from the following sources:

- The sname field of a DHCP or BOOTP reply
- The TFTP server name (option 66) field of a DHCP reply
- The TFTP server address (option 150) field of a DHCP reply

- The siaddr field of a DHCP or BOOTP reply

This list above reflects the priority in which the information is inspected by the DHCP client. For the first two options, in which only the TFTP server name is given, a DNS server must be available to resolve the IP address.

The host-specific router configuration filename is deduced from the boot file (file) field of DHCP or BOOTP responses. The host-specific filename can also be specified in a "network-confg" or "network.cfg" file on a TFTP server for those cases in which a general network configuration file is downloaded before the host-specific configuration file. AutoInstall can also determine the host-specific filename from a DNS IP address to hostname mapping.

AutoInstall over Serial Interfaces

AutoInstall is supported over serial interfaces with HDLC encapsulation or Frame Relay encapsulation. HDLC is the default serial encapsulation. If the AutoInstall process fails over HDLC, it is attempted using Frame Relay encapsulation.

IP Address Acquisition for Serial Interfaces

For AutoInstall over a serial interface, a staging router must be directly connected to the new router using the serial 0 (S0) interface port.

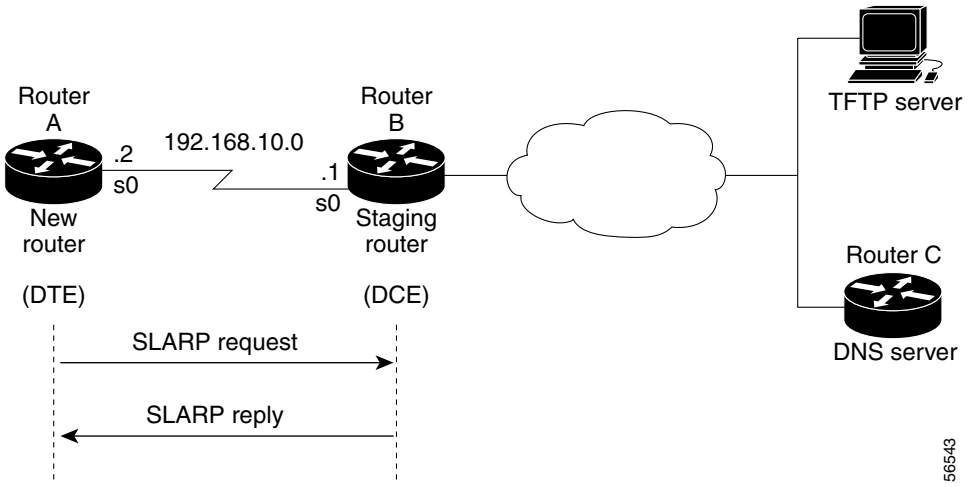
AutoInstall using HDLC

If the new router is connected by an HDLC-encapsulated serial line to the staging router, AutoInstall will send a SLARP request.

In response to a SLARP request, the staging router will send a SLARP reply packet to the new router.

The reply packet contains the IP address and netmask of the staging router's serial interface. If the host portion of the IP address in the SLARP response is 1, AutoInstall will configure the interface of the new router using the value 2 as the host portion of its IP address. If the host portion of the IP address in the SLARP response is 2, AutoInstall will configure the interface of the new router using the value 2 as the host portion of its IP address. Figure 4 shows an example of this address assignment.

Figure 4 Using SLARP to Assign an IP Address to a New Router



In Figure 4, the IP address of the Serial 0 interface on the staging router (Router B) is 192.168.10.1. AutoInstall therefore assigns the IP address 192.168.10.2 to the Serial 0 interface of the new router.

Note If you are using AutoInstall over HDLC, the last 8 bits of host portion of the IP address on the staging router must equal 1 or 2.

AutoInstall Using Frame Relay

If the new router is connected by a Frame Relay-encapsulated serial interface, AutoInstall will send a BOOTP request over the lowest numbered serial or HSSI interface. (The attempt to run AutoInstall over Frame Relay is performed only after attempts are made using SLARP over HDLC, DHCP, and RARP.)

The broadcast BOOTP request sent by the new router will contain the MAC address of the new router's interface. The staging router should be configured to forward the request using a helper address. A DHCP or BOOTP server will then return the IP address assigned to that MAC address. (Note that either a DHCP or BOOTP service can respond to the BOOTP request.)

AutoInstall using Frame Relay can be initiated over only the first serial interface on the new router. Specifically, Autoinstall over Frame Relay can be initiated over Serial 0 (S0), or Serial 1/0 (S1/0). For example, if the new router has serial interfaces S1/0 through S1/3 and S4/0 through S4/3, AutoInstall will be attempted over S1/0 only and cannot be forced to be initiated from S4/0. If AutoInstall over S1/0 fails, an Frame Relay attempt will not be made from any other serial port.

Only a helper address and a Frame Relay map need to be configured on the staging router. No MAC-to-IP address map is needed on the staging router. For configuration details, see the "Configuring a Frame Relay-Encapsulated Serial Interface Connection" section on page 52.

Configuration File Downloading for Serial Interfaces

After acquiring an IP address acquired from the RARP, DHCP, or BOOTP server, the new router will attempt to resolve its hostname from a network configuration file or from a DNS service.

The new router will first attempt to resolves its IP address-to-hostname mapping by sending a TFTP broadcast requesting the file "network-confg" or "cisconet.cfg".

The network configuration file is a configuration file generally shared by several routers. In this case, it is used to map the IP address of the new router to the name of the new router. The network configuration file must reside on a reachable TFTP server and must be globally readable. For example, to assign a hostname of "rtr1" to a new router with the address 192.168.10.2, the following line must appear in the network configuration file:

```
ip host rtr1 192.168.10.2
```

If the new router cannot locate and download a "network-confg" or a "cisconet.cfg" file, or if the IP address-to-hostname mapping does not match the newly acquired IP address, the new router sends a DNS broadcast request. If a DNS server is available and has an entry that maps the acquired IP address of the new router to its name, the new router successfully resolves its name.

If DNS does not have an entry that maps the new router's address to its name, the new router cannot resolve its hostname. The new router will then attempt to download a default configuration file ("router-confg", "router.cfg", or "ciscortr.cfg") from the TFTP server. If this attempt also fails, the router will enter Setup mode, or, if using Frame Relay-based AutoInstall, will enter user EXEC mode.

AutoInstall Configuration Task List

Because AutoInstall is an automated process, no configuration is required on the routing device that will perform AutoInstall. However, you must configure the other systems on the network to provide the necessary elements for the AutoInstall process to work. To configure the network for AutoInstall, perform the tasks described in the following sections. Tasks are identified as required or optional.

- Configuring a Staging Router (Required, unless the required servers are on the same network as the new router)
- Configuring a Cisco Router as a RARP Server (Optional)
- Configuring a Cisco Router as a DHCP Server (Optional)
- Monitoring and Completing the AutoInstall Process (Optional)

These sections are followed by an example of preparing the network for AutoInstall in the "AutoInstall Configuration Examples" section.

Configuring a Staging Router

In most cases you will need to configure a staging router through which the new router running AutoInstall should send TFTP, BOOTP, and DNS requests. This staging router is used to forward broadcast TFTP and DNS requests.

Assuming that the proper protocol is already configured on the interface that will be connected to the new router, the minimum configuration you will need to perform on the staging router is to use the **ip helper-address** *address* interface configuration mode command, where the *address* argument is the IP address of the TFTP server to be used during the AutoInstall process.

The following sections go into more detail on configuring the proper protocol for each interface type:

- Configuring a LAN Interface Connection on the Staging Router
- Configuring an HDLC-Encapsulated Serial Interface Connection
- Configuring a Frame Relay-Encapsulated Serial Interface Connection

Configuring a LAN Interface Connection on the Staging Router

To set up AutoInstall using an Ethernet, Token Ring, or FDDI interface, you must modify the configuration of the staging router. To configure the staging router to forward requests from the new router using a LAN interface connection, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface {ethernet | tokenring | fddi} [slot/]port`	Enters interface configuration mode for the specified LAN interface.
Step 2	`Router(config-if)# ip address address mask`	Specifies an IP address for the interface.
Step 3	`Router(config-if)# ip helper-address address`	Specifies the destination broadcast or host address for TFTP server, BOOTP server, and DNS server requests. A separate **ip helper-address** command is needed for each server if they are on different hosts. You can also configure multiple TFTP server targets by using multiple **ip helper-address** commands.

	Command	Purpose
Step 4	`Router(config-if)# ^Z`	Exits to EXEC mode.
Step 5	`Router# copy running-config startup-config`	Saves the running configuration to the startup configuration file.

Typically, the LAN interface and IP address are already configured on the existing router.

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

```
Router# show running-config
 .
 .
 .
interface Ethernet 0
 ip address 172.31.10.1 255.255.255.0
 ip helper-address 172.31.20.5
 .
 .
 .
```

Configuring an HDLC-Encapsulated Serial Interface Connection

To set up AutoInstall to use a serial line with HDLC encapsulation (the default for serial interfaces), you must configure the staging router. To configure the staging router to forward requests from the new router using a HDLC-encapsulated serial interface connection, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial` *interface-number*	Configures the serial interface that connects to the new router with HDLC encapsulation (the default), and enters interface configuration mode for the specified interface.
Step 2	`Router(config-if)# ip address` *address mask*	Enters an IP address for the interface. The host portion of the address must have a value of 1 or 2. (AutoInstall over Frame Relay does not have this address constraint.)
Step 3	`Router(config-if)# ip helper-address` *address*	Specifies the destination broadcast or host address for TFTP server, BOOTP server, and DNS server requests. A separate **ip helper-address** command is needed for each server if they are on different hosts. You can also configure multiple TFTP server targets by using multiple **ip helper-address** commands.
Step 4	`Router(config-if)# clock rate` *bps*	(Optional) Configures a DCE clock rate for the serial line. This step is needed only for DCE appliques.
Step 5	`Router(config-if)# ^Z`	Ends the current configuration session and returns you to privileged EXEC mode.
Step 6	`Router# copy running-config startup-config`	Saves the running configuration to the startup configuration file.

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

```
Router# show running-config
  .
  .
  .
interface serial 0
 ip address 172.31.10.1 255.255.255.0
 ip helper-address 172.31.20.5
  .
  .
  .
```

Configuring a Frame Relay-Encapsulated Serial Interface Connection

To set up AutoInstall to use a serial line with Frame Relay encapsulation, you must configure the staging router. To configure the staging router to forward requests from the new router using a Frame Relay-encapsulated serial interface connection, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial slot/port`	Configures the serial interface that connects to the new router, and enters interface configuration mode.
Step 2	`Router(config-if)# encapsulation frame-relay`	Configures Frame Relay encapsulation on the interface that connects to the new router.

Step 3	`Router(config-if)# frame-relay map ip ip-address dlci-number`
	or
	`Router(config-if)# frame-relay interface-dlci dlci-number [protocol ip ip-address]`
		Creates a Frame Relay map pointing back to the new router.
		or
		For point-to-point subinterfaces, assigns a data link connection identifier (DLCI) to the interface that connects to the new router, and provides the IP address of the serial port on the new router. This command should be used if the staging router is acting as the BOOTP server.

Step 4	`Router(config-if)# frame-relay intf-type dce`	Configures a Frame Relay switch type. This step is required only for DCE appliques.
Step 5	`Router(config-if)# ip address address mask`	Specifies an IP address for the interface. This step sets the IP address of the existing router.
Step 6	`Router(config-if)# ip helper-address address`	Configures a helper address for the TFTP server.
Step 7	`Router(config-if)# clock rate bps`	(Optional) Configures a DCE clock rate for the serial line. This step is needed only for DCE appliques.
Step 8	`Router(config-if) keepalive [seconds]`	(Optional) Configures the keepalive interval, which is the frequency at which the Cisco IOS software sends messages to itself (for Ethernet and Token Ring) or to the other end (for serial) to ensure a network interface is “up.”
Step 9	`Router(config-if)# ^Z`	Ends the current configuration session and returns you to privileged EXEC mode.
Step 10	`Router# copy running-config startup-config`	Saves the running configuration to the startup configuration file.

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

In the following example, the configuration file for the staging router contains the commands needed to configure the router for Frame Relay-based AutoInstall on a serial line:

```
Router# show running-config
 .
 .
 .
interface serial 0
 ip address 172.31.20.20 255.255.255.0
 encapsulation frame-relay
!In the following command, 172.31.10.1 is the IP address of the new router's interface
!and 48 is the PVC identifier
 frame-relay map ip 172.31.10.1 255.255.255.0 48 dlci
 ip helper-address 172.31.20.5
 .
 .
 .
```

Configuring a Cisco Router as a RARP Server

Use the **ip rarp-server** *ip-address* interface configuration command to enable the RARP service on a Cisco router, where *ip-address* is the IP address of the TFTP server. Use the **arp** *ip-address MAC-address* **arpa** global configuration command to map the MAC address of the new router to a specific IP address.

The following is an example of a static ARP entry in a configuration file for a typical Ethernet host:

```
arp 192.168.7.19 0800.0900.1834 arpa
```

Configuring a Cisco Router as a DHCP Server

The following sections describe how to configure a Cisco routing device to function as a DHCP server and provide the desired information to routers performing AutoInstall. The last section also describes the commands necessary to use another router to forward AutoInstall requests.

- Configuring a Configuration Filename on a Cisco DHCP Server
- Configuring the IP Address of the TFTP Server on a Cisco DHCP Server
- Configuring the TFTP Server Name on a Cisco DHCP Server
- Configuring the DNS Server Address on a Cisco DHCP Server
- Specifying the Default Router on a Cisco DHCP Server

You do not have to use a Cisco device as the DHCP server; any DHCP server can provide the required information to the new router.

Note The configuration tasks used in this section use manual bindings for DHCP address allocation, which requires you to enter the MAC address for the new router. However, you can also use the automatic binding available for DHCP. For complete DHCP configuration details, refer to the "Configuring DHCP" chapter of the Release 12.2 *Cisco IOS IP Configuration Guide*.

Configuring a Configuration Filename on a Cisco DHCP Server

You have the option of configuring the DHCP server to provide a specific configuration filename to be used by the DHCP client during the AutoInstall process. When you associate the MAC address of a new router with a specific configuration file, AutoInstall will be able to request this file from the TFTP server. To determine the name of the specific configuration file, the DHCP client will examine the "file" field of the DHCP/BOOTP reply, or, if there is an option overload of the "file" field, at the value of option 67. When a configuration file (bootfile) is specified, this configuration file should be in the /tftpboot directory of the TFTP server.

Tip We recommend that configuration files intended for specific hosts be saved with the name "*hostname*-confg" or, if using a DOS-based TFTP server, "*hostname*.cfg", where *hostname* is the name of the intended routing device. The hostname specified in the configuration file should match the filename.

To specify the name of the configuration file to be used by the DHCP client, use the following commands on the Cisco routing device running the DHCP server beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# service dhcp`	Enables the DHCP server on your routing device.
Step 2	`Router(config)# ip dhcp pool` *identifier*	Specifies a pool to be used for DHCP address allocation, and enters DHCP pool configuration mode. The *identifier* argument is the alphanumeric "name" of the DHCP pool to be configured.
Step 3	`Router(config-dhcp)# host` *address* [*mask* | **/***prefix-length*]	Specifies the IP address to be assigned to the DHCP client.
Step 4	`Router(config-dhcp)# hardware-address` *hardware-address* [*type*]	Specifies the MAC address of the DHCP client.
Step 5	`Router(config-dhcp)# bootfile` *filename*	Specifies the name of the configuration file that the DHCP client should download from the TFTP server.

Configuring the IP Address of the TFTP Server on a Cisco DHCP Server

There are two ways of configuring the DHCP server to return the IP address of the TFTP server to be used by the DHCP client during AutoInstall: by specifying the siaddr field or option 150. The recommended method is to configure the DHCP server to return a value for option 150.

To configure the value for option 150, use the following commands on the Cisco routing device running the DHCP server beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip dhcp pool` *identifier*	Specifies a pool to be used for DHCP address allocation and enters DHCP pool configuration mode. The *identifier* argument is the alphanumeric "name" of the DHCP pool to be configured.
Step 2	`Router(config-dhcp)# host` *address* [*mask* | **/***prefix-length*]	Specifies the IP address to be assigned to the DHCP client.

	Command	Purpose
Step 3	`Router(config-dhcp)# hardware-address MAC-address [type]`	Specifies the MAC address of the DHCP client.
Step 4	`Router(config-dhcp)# option 150 ip address`	Specifies the IP address of the TFTP server (option 150) to be used by the DHCP client.

To configure the value to be returned in the siaddr field, use the following commands on the Cisco routing device running the DHCP server beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip dhcp pool identifier`	Specifies a pool to be used for DHCP address allocation and enters DHCP pool configuration mode. The *identifier* is the alphanumeric "name" of the DHCP pool to be configured.
Step 2	`Router(config-dhcp)# host address [mask | /prefix-length]`	Specifies the IP address to be assigned to the DHCP client.
Step 3	`Router(config-dhcp)# hardware-address MAC-address [type]`	Specifies the MAC address of the DHCP client.
Step 4	`Router(config-dhcp)# next-server address [address2...address8]`	Specifies the IP address of the TFTP server (ipaddr) to be used by the DHCP client. The `[address2...address8]` syntax indicates that more than one TFTP server (up to eight) can be specified if you want to configure backup TFTP servers.

Configuring the TFTP Server Name on a Cisco DHCP Server

The alternative to specifying the IP address of the TFTP server to be used during AutoInstall is to specify the hostname of the TFTP server. This information can be returned to the DHCP client in the sname field of a BOOTP reply or in the option 66 field of a DHCP reply. If only the TFTP server name is to returned, a DNS server must be available to the new router to translate the TFTP server name to an IP address.

To configure the DHCP server to return the TFTP server name as option 66, use the following commands on the Cisco routing device running the DHCP server, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip dhcp pool identifier`	Specifies a pool to be used for DHCP address allocation and enters DHCP pool configuration mode. The *identifier* argument is the alphanumeric "name" of the DHCP pool to be configured.
Step 2	`Router(config-dhcp)# host ip-address [mask | /prefix-length]`	Specifies the IP address to be assigned to the DHCP client.
Step 3	`Router(config-dhcp)# hardware-address hardware-address [type]`	Specifies the MAC address of the DHCP client.
Step 4	`Router(config-dhcp)# option 66 ascii tftp-server-name`	Specifies the TFTP server name (option 66) to be returned to the DHCP client for use in the AutoInstall process. The TFTP server name should be a Fully Qualified Domain Name (FQDN).

Configuring the DNS Server Address on a Cisco DHCP Server

To specify the DNS sever that should be used during the AutoInstall process, use the following command on the DHCP server in DHCP pool configuration mode:

Command	Purpose
`Router(dhcp-config)# dns-server address`	Specifies the IP address of the DNS server to use to resolve hostnames during the AutoInstall process.

Specifying the Default Router on a Cisco DHCP Server

In cases where more than one router is connected to new router, you may need to specify a default router through which the new router running AutoInstall should send TFTP and DNS requests. To configure the DHCP server to return this information to the DHCP client, use the following command on the DHCP server in DHCP pool configuration mode:

Command	Purpose
`Router(dhcp-config)# default-router address`	Specifies the IP address of the router to use as the first hop for AutoInstall unicast requests.

If the TFTP server IP address can be deduced by AutoInstall and if the default router option is provided, Autoinstall will recognize that the default router is the first hop toward the TFTP server. If the TFTP server IP address cannot be deduced, the default router option (if present) will be ignored.

In the following example, the address 10.0.20.20 is specified as the next hop toward the TFTP server with the address 172.16.1.1:

```
ip dhcp pool 1
   host 10.0.20.54 255.255.255.240
   hardware-address 0000.0c59.fcb0
   bootfile R1-confg
!option 150 specifies the TFTP server address
   option 150 ip 172.16.1.1
   default-router 10.0.20.20
```

Monitoring and Completing the AutoInstall Process

To monitor the progress of the AutoInstall process, you have the option of attaching a terminal directly to the system console port of the new router before turning it on. Note that this monitoring is optional; AutoInstall will start without user input after a short period of time. When a router boots for the first time, the CLI begins in Setup mode. Setup mode prompts you for manual configuration of the Cisco IOS software at the console. (For more information on Setup mode, see the "Using Setup" section in this chapter.)

For most platforms, you will see prompts similar to the following. You should respond to these prompts as indicated by the bold text:

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

Note If you do not respond to the setup prompts, AutoInstall will begin automatically after a short period of time.

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

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

Typically, additional notifications will be displayed as the process goes through the steps outlined earlier in this chapter. If the AutoInstall succeeds, and the configuration file contains all of the required configuration information, the CLI will enter user EXEC mode. After the AutoInstall process is completed, you should verify that the entire configuration file was downloaded, and that the file is uncorrupted. This verification can be performed from the console terminal, or by using a Telnet connection from an external host on the network. You can view the running configuration file using the **show running-config** or **more system:running-config** EXEC mode commands.

After verifying the running configuration, you should save it as the start-up configuration using the **copy running-config startup-config** or **copy system:running-config nvram:startup-config** EXEC mode command.

If the AutoInstall process fails, or if the configuration file does not contain all of the required configuration information, the router will remain in Setup mode to allow you to complete the configuration process. If no input is entered, the new router will continue to issue AutoInstall broadcast requests to attempt to learn its hostname and acquire IP addresses. The broadcast frequency will dwindle to every 10 minutes after several attempts.

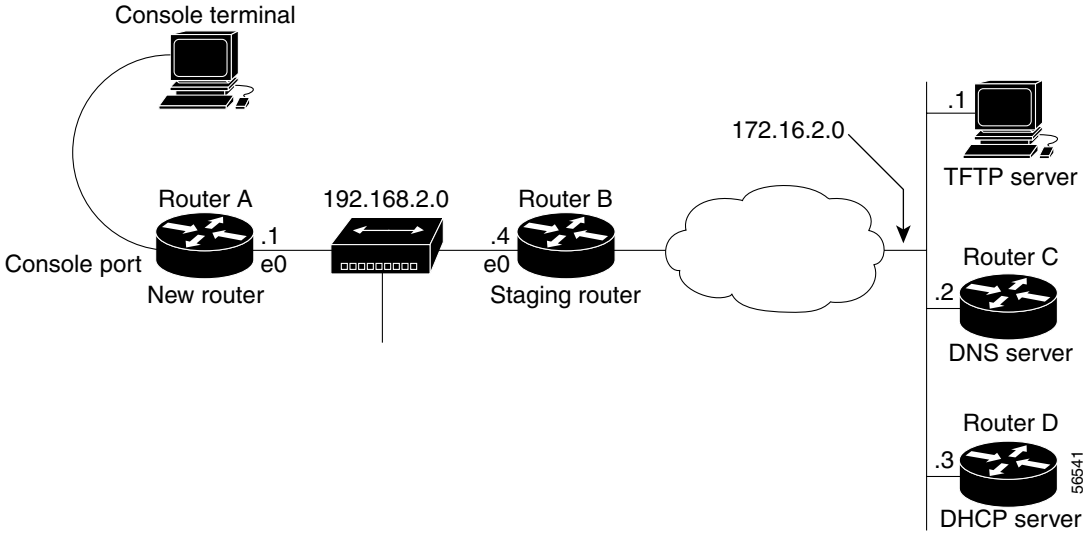
AutoInstall Configuration Examples

This section contains an example of configuring network elements for DHCP-based AutoInstall.

Network Configuration for DHCP-Based AutoInstall Example

For this example, the hypothetical network topology introduced in the "Understanding AutoInstall" section on page 40 is used to show the configuration options on the various hosts. Figure 5 shows the network topology with IP addresses. Note that the TFTP, DNS, and DHCP servers are shown to be on different devices for illustrational purposes only. In the sample output for the routers, only the commands relevant to the AutoInstall process are shown.

Figure 5 Example of AutoInstall Using an Ethernet Interface



Configuration for Router A

Router A is the new router. No manual configuration is needed. Prior to turning the router on, the user connects only the console port and the ethernet 0 (e0) port.

Configuration for Router B

Router B is the staging router. In this example, AutoInstall is performed over an Ethernet interface.

```
interface e0
!specify the address for the TFTP server
  ip helper-address 172.16.2.1
!specify the address for the DNS server
  ip helper-address 172.16.2.2
!specify the address for the DHCP server
  ip helper-address 172.16.2.3
```

Configuration for Router C

Router C is a Cisco router running a DNS service. DNS is enabled by default.

```
!assign the hostname "rtr1" to Router A
ip host rtr1 192.168.2.1
!specify the TFTP address to hostname mapping
ip host tftpserv 172.16.2.1
```

Configuration for Router D

Router D is a Cisco router running a DHCP service. This example uses a static DHCP pool with manual bindings. (To configure a dynamic pool, the **network** and **lease** DHCP pool configuration commands should be used.)

```
ip dhcp pool 1
!specify the IP address that will be assigned to the new router
  host 192.168.2.1
!specify the hardware (MAC) address of the new router
  hardware-address 0000.0c59.fcb0
!specify the address of the DNS server
  dns-server 172.16.2.2
```

```
!specify the TFTP server address (DHCP option 10)
  option 150 ip 172.16.1.1
!specify the configuration file that AutoInstall should get from the TFTP server
  bootfile rtr1-confg
```

Using Setup

Setup (also known as the System Configuration Dialog) is an interactive CLI mode that guides you through first-time configuration by prompting you for the details needed to start your router functioning in the network. While Setup mode is a quick and easy way to perform first-time configuration of a router, you can also use it after first-time startup to perform basic configuration changes, as described in the following sections:

- Using Setup After First-Time Startup
- Using Streamlined Setup

Before using Setup, you should have the following information so that you can configure the system properly:

- Which interfaces you want to configure
- Which routing protocols you wish to enable
- Whether the router is to perform bridging
- Network addresses for the protocols being configured
- Password strategy for your environment

Note Refer to the documentation for your particular hardware platform for information on how you should use Setup for first-time startup.

Using Setup After First-Time Startup

The Cisco IOS command-line interface (CLI) allows you to make very detailed changes to your system configuration. However, some major configuration changes do not require the granularity provided by the CLI. You can use Setup to configure general characteristics of the system. For example, you might want to use Setup to add a protocol suite, to make major addressing scheme changes, or to configure a newly installed interface. Although you can use the configuration modes available though the CLI to make these changes, the Setup mode provides you with a high-level view of the configuration and guides you through the configuration process.

If you are not familiar with Cisco products and the CLI, Setup is a particularly valuable tool because it prompts you for the specific information required to configure your system.

Note If you use Setup to modify a configuration because you have added to or modified the hardware, be sure to verify the physical connections using the **show version** EXEC command. Also, verify the logical port assignments using the **show running-config** EXEC command to ensure that you configure the proper port. Refer to the hardware documentation for your platform for details on physical and logical port assignments.

To enter Setup mode, use the following command in privileged EXEC mode:

Command	Purpose
`Router# setup`	Enters Setup mode.

When you enter the **setup** EXEC command after first-time startup, an interactive dialog called the *System Configuration Dialog* appears on the system console screen. The System Configuration Dialog guides you through the configuration process. It prompts you first for global parameters and then for interface parameters. The values shown in brackets next to each prompt reflect either the default settings or the last configured setting.

You must progress through the System Configuration Dialog until you come to the item that you intend to change. To accept default settings for items that you do not want to change, press the Return or Enter key. The default choice is indicated by square brackets (for example, [yes]) before the prompt colon (:).

To exit Setup and return to privileged EXEC mode without making changes and without progressing through the entire System Configuration Dialog, press **Ctrl**-**C**.

The facility also provides help text for each prompt. To access help text, press the question mark (**?**) key at a prompt.

When you complete your changes, the system will automatically display the configuration file that was created during the Setup session. It also asks you if you want to use this configuration. If you answer Yes, the configuration is saved to NVRAM as the startup configuration file. If you answer No, the configuration is not saved and the process begins again. There is no default for this prompt; you must answer either Yes or No.

Note If any problems exist with the configuration file pointed to in NVRAM, or if the ignore NVRAM bit (bit 6) is set in the configuration register, the router enters the streamlined Setup mode. For more information on the streamlined Setup mode, see the "Using Streamlined Setup" section.

In the following example Setup is used to configure interface serial 0 and to add ARAP and IP/IPX PPP support on the asynchronous interfaces. Note that prompts and the order in which they appear on the screen vary depending on the platform and the interfaces installed in the device.

```
Router# setup

         --- System Configuration Dialog ---

At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.

Continue with configuration dialog? [yes]:

First, would you like to see the current interface summary? [yes]:

Interface          IP-Address      OK?  Method    Status                  Protocol
Ethernet0          172.16.72.2     YES  manual    up                      up
Serial0            unassigned      YES  not set   administratively down  down
Serial1            172.16.72.2     YES  not set   up                      up

Configuring global parameters:

  Enter host name [Router]:

The enable secret is a one-way cryptographic secret used
instead of the enable password when it exists.


Enter enable secret [<Use current secret>]:

The enable password is used when there is no enable secret
and when using older software and some boot images.

  Enter enable password [ww]:
  Enter virtual terminal password [ww]:
  Configure SNMP Network Management? [yes]:
    Community string [public]:
  Configure DECnet? [no]:
  Configure AppleTalk? [yes]:
    Multizone networks? [no]: yes
  Configure IPX? [yes]:
  Configure IP? [yes]:
    Configure IGRP routing? [yes]:
      Your IGRP autonomous system number [15]:
  Configure Async lines? [yes]:
    Async line speed [9600]: 57600
    Configure for HW flow control? [yes]:
    Configure for modems? [yes/no]: yes
      Configure for default chat script? [yes]: no
    Configure for Dial-in IP SLIP/PPP access? [no]: yes
      Configure for Dynamic IP addresses? [yes]: no
      Configure Default IP addresses? [no]: yes
      Configure for TCP Header Compression? [yes]: no
      Configure for routing updates on async links? [no]:
    Configure for Async IPX? [yes]:
    Configure for Appletalk Remote Access? [yes]:
      AppleTalk Network for ARAP clients [1]: 20
      Zone name for ARAP clients [ARA Dialins]:


Configuring interface parameters:

Configuring interface Ethernet0:
```

```
  Is this interface in use? [yes]:
  Configure IP on this interface? [yes]:
    IP address for this interface [172.16.72.2]:
    Number of bits in subnet field [8]:
    Class B network is 172.16.0.0, 8 subnet bits; mask is /24
  Configure AppleTalk on this interface? [yes]:
    Extended AppleTalk network? [yes]:
    AppleTalk starting cable range [1]:
    AppleTalk ending cable range [1]:
    AppleTalk zone name [Sales]:
    AppleTalk additional zone name:
  Configure IPX on this interface? [yes]:
    IPX network number [1]:


Configuring interface Serial0:
  Is this interface in use? [no]: yes
  Configure IP on this interface? [no]: yes
  Configure IP unnumbered on this interface? [no]: yes
    Assign to which interface [Ethernet0]:
  Configure AppleTalk on this interface? [no]: yes
    Extended AppleTalk network? [yes]:
    AppleTalk starting cable range [2]: 3
    AppleTalk ending cable range [3]: 3
    AppleTalk zone name [myzone]: ZZ Serial
    AppleTalk additional zone name:
  Configure IPX on this interface? [no]: yes
    IPX network number [2]: 3


Configuring interface Serial1:
  Is this interface in use? [yes]:
  Configure IP on this interface? [yes]:
  Configure IP unnumbered on this interface? [yes]:
    Assign to which interface [Ethernet0]:
  Configure AppleTalk on this interface? [yes]:
    Extended AppleTalk network? [yes]:
    AppleTalk starting cable range [2]:
    AppleTalk ending cable range [2]:
    AppleTalk zone name [ZZ Serial]:
    AppleTalk additional zone name:
  Configure IPX on this interface? [yes]:
    IPX network number [2]:
Configuring interface Async1:
    IPX network number [4]:
    Default client IP address for this interface [none]: 172.16.72.4
Configuring interface Async2:
    IPX network number [5]:
    Default client IP address for this interface [172.16.72.5]:
Configuring interface Async3:
    IPX network number [6]:
    Default client IP address for this interface [172.16.72.6]:
Configuring interface Async4:
    IPX network number [7]:
    Default client IP address for this interface [172.16.72.7]:
Configuring interface Async5:
    IPX network number [8]:
    Default client IP address for this interface [172.16.72.8]:


Configuring interface Async6:
    IPX network number [9]:
    Default client IP address for this interface [172.16.72.9]:
Configuring interface Async7:
```

```
    IPX network number [A]:
    Default client IP address for this interface [172.16.72.10]:
Configuring interface Async8:
    IPX network number [B]:
    Default client IP address for this interface [172.16.72.11]:
Configuring interface Async9:
    IPX network number [C]:
    Default client IP address for this interface [172.16.72.12]:
Configuring interface Async10:
    IPX network number [D]:
    Default client IP address for this interface [172.16.72.13]:
Configuring interface Async11:
    IPX network number [E]:
    Default client IP address for this interface [172.16.72.14]:
Configuring interface Async12:
    IPX network number [F]:
    Default client IP address for this interface [172.16.72.15]:
Configuring interface Async13:
    IPX network number [10]:
    Default client IP address for this interface [172.16.72.16]:
Configuring interface Async14:
    IPX network number [11]:
    Default client IP address for this interface [172.16.72.17]:
Configuring interface Async15:
    IPX network number [12]:
    Default client IP address for this interface [172.16.72.18]:
Configuring interface Async16:
    IPX network number [13]:
    Default client IP address for this interface [172.16.72.19]:

The following configuration command script was created:
!
!This is the running configuration file that will be used
!if you answer yes to the prompt which follows
!
hostname Router
enable secret 5 $1$krIg$emfYm/1OwHVspDuS8Gy0K1
enable password ww
line vty 0 4
password ww
snmp-server community public
!
no decnet routing
appletalk routing
ipx routing
ip routing
!
line 1 16
speed 57600
flowcontrol hardware
modem inout
!
arap network 20 ARA Dialins
line 1 16
arap enable
autoselect
!
! Turn off IPX to prevent network conflicts.
interface Ethernet0
no ipx network
interface Serial0
no ipx network
interface Serial1
no ipx network
```

```
!
interface Ethernet0
ip address 172.16.72.2 255.255.255.0
appletalk cable-range 1-1 1.204
appletalk zone Sales
ipx network 1
no mop enabled
!
interface Serial0
no shutdown
no ip address
ip unnumbered Ethernet0
appletalk cable-range 3-3
appletalk zone ZZ Serial
ipx network 3
no mop enabled
!
interface Serial1
no ip address
ip unnumbered Ethernet0
appletalk cable-range 2-2 2.2
appletalk zone ZZ Serial
ipx network 2
no mop enabled
!
Interface Async1
ipx network 4
ip unnumbered Ethernet0
peer default ip address 172.16.72.4
async mode interactive
!
Interface Async2
ipx network 5
ip unnumbered Ethernet0
peer default ip address 172.16.72.5
async mode interactive
!
Interface Async3
ipx network 6
ip unnumbered Ethernet0
peer default ip address 172.16.72.6
async mode interactive
!
Interface Async4
ipx network 7
ip unnumbered Ethernet0
peer default ip address 172.16.72.7
async mode interactive
async dynamic address
!
Interface Async5
ipx network 8
ip unnumbered Ethernet0
peer default ip address 172.16.72.8
async mode interactive
!
Interface Async6
ipx network 9
ip unnumbered Ethernet0
peer default ip address 172.16.72.9
async mode interactive
!
Interface Async7
ipx network A
```

```
ip unnumbered Ethernet0
peer default ip address 172.16.72.10
async mode interactive
!
Interface Async8
ipx network B
ip unnumbered Ethernet0
peer default ip address 172.16.72.11
async mode interactive
!
Interface Async9
ipx network C
ip unnumbered Ethernet0
peer default ip address 172.16.72.12
async mode interactive
!
Interface Async10
ipx network D
ip unnumbered Ethernet0
peer default ip address 172.16.72.13
async mode interactive
!
Interface Async11
ipx network E
ip unnumbered Ethernet0
peer default ip address 172.16.72.14
async mode interactive
!
Interface Async12
ipx network F
ip unnumbered Ethernet0
peer default ip address 172.16.72.15
async mode interactive
!
Interface Async13
ipx network 10
ip unnumbered Ethernet0
peer default ip address 172.16.72.16
async mode interactive
!
Interface Async14
ipx network 11
ip unnumbered Ethernet0
peer default ip address 172.16.72.17
async mode interactive
!
Interface Async15
ipx network 12
ip unnumbered Ethernet0
peer default ip address 172.16.72.18
async mode interactive
!
Interface Async16
ipx network 13
ip unnumbered Ethernet0
peer default ip address 172.16.72.19
async mode interactive
!
router igrp 15
network 172.16.0.0
!
end

Use this configuration? [yes/no]: yes
```

```
Building configuration...

Use the enabled mode 'configure' command to modify this configuration.

Router#
```

Using Streamlined Setup

The streamlined **setup** mode permits your router to load a system image from a network server when there are problems with the startup configuration. If RXBOOT is on your system, the system automatically enters the streamlined Setup mode when your router is rebooted (or you are attempting to load a system image from a network server) under any of the following circumstances:

- You issued an **erase startup-config** EXEC command, thereby deleting the startup configuration file.
- You have bit 6 (ignore NVRAM configuration) set in the configuration register.
- Your startup configuration has been corrupted.
- You configured the router to boot from a network server (the last four bits of the configuration register are not equal to 0 or 1) and there is no Flash or no valid image in Flash.
- You configured the router to boot the RXBOOT image.

Note that these conditions are the same as for entering standard ROM monitor mode.

Note The streamlined Setup mode is available only if your router has an RXBOOT ROM image installed. If a RXBOOT image is not available, the system will enter ROM monitor mode instead.

The streamlined Setup mode differs from the standard Setup mode because streamlined Setup does not ask you to configure global router parameters. You are prompted only to configure interface parameters, which permit your router to boot.

As with ROM monitor mode, the configuration information you provide in RXBOOT setup mode is *temporary* and exists only so that you can proceed with booting your system. When you reload the system, your original configuration is left intact. If your startup configuration is corrupted, enter the **setup** EXEC command, and configure the basic parameters. Then issue the **copy running-config startup-config** EXEC command to write this configuration to NVRAM.

The following example shows a router entering streamlined Setup mode:

```
        --- System Configuration Dialog ---

Default settings are in square brackets '[]'.

Configuring interface IP parameters for netbooting:
```

Note The message “Configuring interface IP parameters for netbooting” only appears if you are booting over a network server and your configuration has insufficient IP information.

The streamlined Setup mode continues by prompting you for interface parameters for each installed interface. The facility asks if an interface is in use. If so, the facility then prompts you to provide an IP address and subnet mask bits for the interface. Enter the subnet mask bits as a decimal value, such as 5.

The following example shows the portion of the streamlined Setup mode that prompts for interface parameters. In the example, the streamlined Setup mode is prompting for Ethernet 0 interface parameters and Serial 0 interface parameters:

```
Configuring interface Ethernet0:
  Is this interface in use? [yes]:
  Configure IP on this interface? [yes]:
    IP address for this interface: 192.168.78.50
    Number of bits in subnet field [0]: 5
    Class C network is 192.168.78.0, 5 subnet bits; mask is 255.255.255.248

Configuring interface Serial0:
  Is this interface in use? [yes]:
  Configure IP on this interface? [yes]:
    IP address for this interface: 192.169.78.34
    Number of bits in subnet field [5]:
    Class C network is 192.168.78.0, 5 subnet bits; mask is 255.255.255.248
```

Using Configuration Applications

You can also configure Cisco IOS software-based devices using the Cisco ConfigMaker application.

For more information on configuration tools, go to Cisco.com.

Cisco ConfigMaker

Cisco ConfigMaker is an easy-to-use Microsoft Windows (95/98/NT) application used to configure a small network of Cisco routers (800, 1000, 1600, 1700, 2500, 2600, 3600, and 4000 series), switches, hubs, and other network devices from a single PC. Using Cisco ConfigMaker does not require knowledge of the Cisco IOS software command-line interface. Cisco ConfigMaker is designed for resellers and network administrators of small to medium-sized businesses that are familiar with LAN and WAN fundamentals and basic network design.

ConfigMaker makes configuring an HDLC, Frame Relay, or ISDN wide-area network connection between routers or the Internet as easy as drawing a network diagram. The tool guides users step-by-step through network design and addressing tasks and automatically delivers configuration files to individual routers on the network. ConfigMaker provides a graphical view of the entire network and lets the user build network diagrams using standard copy/paste, drag/drop, and online editing functions. ConfigMaker enables the user to monitor router and network configuration status at a glance with simple color codes.

The Cisco ConfigMaker software download is made available to customers free of charge. For details about the Cisco ConfigMaker application, and to download a copy of the software, go to http://www.cisco.com/go/configmaker.

This document first published April 2001. Last updated September 2003.

Configuring Operating Characteristics for Terminals

This chapter describes how to configure operating characteristics for terminals. For a complete description of the terminal operation commands in this chapter, refer to the "Terminal Operating Characteristics Commands" chapter in the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Terminal Operating Characteristics Configuration Task List

To configure operating characteristics for terminals, perform any of the tasks described in the following sections. All tasks in this chapter are optional.

- Displaying Information About the Current Terminal Session
- Setting Local Terminal Parameters
- Saving Local Settings Between Sessions
- Ending a Session
- Changing Terminal Session Parameters
- Displaying Debug Messages on the Console and Terminals
- Recording the Serial Device Location
- Changing the Retry Interval for a Terminal Port Queue
- Configuring LPD Protocol Support on a Printer

Note For additional information about configuring terminal services, see the Release 12.2 *Cisco IOS Terminal Services Configuration Guide* and the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

Displaying Information About the Current Terminal Session

To display terminal line information, use the following commands in user or privileged EXEC mode, as needed:

Command	Purpose
`Router> show whoami` *text*	Displays information about the terminal line being used for the current session, including host name, line number, line speed, and location. If text is included as an argument in the command, that text is displayed as part of the additional data about the line.
`Router> where`	Lists all open sessions associated with the current terminal line. An asterisk (*) in the output indicates the current terminal session.

The following example shows sample output of the **show whoami** command:

```
Router> show whoami

Comm Server "Router", Line 0 at 0bps.  Location "Second floor, West"

--More--
Router>
```

To prevent the information from disappearing from the screen, the **show whoami** command always displays a --More-- prompt before returning to the CLI prompt. Press the Spacebar to return to the prompt.

Setting Local Terminal Parameters

The **terminal** EXEC mode commands enable or disable features for the current session only. You can use these commands to temporarily change terminal line settings without changing the stored configuration file.

To display a list of the commands for setting terminal parameters for the current session, use the following command in EXEC mode:

Command	Purpose
`Router# terminal ?`	Lists the commands for setting terminal parameters.

The following example shows sample output for the **terminal ?** command. Commands available on your routing device will vary depending on the software image and hardware you are using.

```
Router> terminal ?
  autohangup              Automatically hangup when last connection closes
  data-character-bits     Size of characters being handled
  databits                Set number of data bits per character
  dispatch-character      Define the dispatch character
  dispatch-timeout        Set the dispatch timer
  download                Put line into 'download' mode
  editing                 Enable command line editing
  escape-character        Change the current line's escape character
  exec-character-bits     Size of characters to the command exec
  flowcontrol             Set the flow control
```

```
full-help               Provide help to unprivileged user
help                    Description of the interactive help system
history                 Enable and control the command history function
hold-character          Define the hold character
ip                      IP options
keymap-type             Specify a keymap entry to use
lat                     DEC Local Area Transport (LAT) protocol-specific
                        configuration
length                  Set number of lines on a screen
no                      Negate a command or set its defaults
notify                  Inform users of output from concurrent sessions
padding                 Set padding for a specified output character
parity                  Set terminal parity
rxspeed                 Set the receive speed
special-character-bits  Size of the escape (and other special) characters
speed                   Set the transmit and receive speeds
start-character         Define the start character
stop-character          Define the stop character
stopbits                Set async line stop bits
telnet                  Telnet protocol-specific configuration
telnet-transparent      Send a CR as a CR followed by a NULL instead of a CR
                        followed by a LF
terminal-type           Set the terminal type
transport               Define transport protocols for line
txspeed                 Set the transmit speeds
width                   Set width of the display terminal
```

Throughout this chapter, many terminal settings can be configured for all terminal sessions or for just the current terminal session. Settings for all terminal sessions are configured in line configuration mode and can be saved. Settings for the current session are specified using EXEC mode commands that generally begin with the word **terminal**.

Saving Local Settings Between Sessions

You can configure the Cisco IOS software to save local parameters (set with **terminal** EXEC mode commands) between sessions. Saving these local settings ensures that the parameters the user sets will remain in effect between terminal sessions. This function is useful for servers in private offices. To save local settings between sessions, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# private`	Saves local settings between sessions.

If the **private** line configuration command is not used, user-set terminal parameters are cleared when the session ends with either the **exit** EXEC mode command or when the interval set with the **exec-timeout** line configuration command has passed.

Ending a Session

To end a session, use the following command in EXEC mode:

Command	Purpose
`Router> quit`	Ends the current session.

Refer to the "Managing Connections, Menus, and System Banners" chapter for more information on ending sessions and closing connections.

Changing Terminal Session Parameters

This section explains how to change terminal and line settings both for a particular line and locally. The local settings are set with the **terminal** EXEC mode commands. They temporarily override the settings made by the system administrator and remain in effect only until you exit the system. In line configuration mode, you can set terminal operation characteristics that will be in operation for that line until the next time you change the line parameters.

The following sections describe the tasks used to make the more common changes to the terminal and line settings:

- Defining the Escape Character and Other Key Sequences
- Specifying Telnet Operation Characteristics
- Configuring Data Transparency for File Transfers
- Specifying an International Character Display

The following sections describe the tasks used to make the less common changes to the terminal and line settings:

- Setting Character Padding
- Specifying the Terminal and Keyboard Type
- Changing the Terminal Screen Length and Width
- Enabling Pending Output Notifications
- Creating Character and Packet Dispatch Sequences
- Changing Flow Control for the Current Session
- Enabling Session Locking
- Configuring Automatic Baud Rate Detection
- Setting a Line as Insecure
- Configuring Communication Parameters for Terminal Ports

Defining the Escape Character and Other Key Sequences

You can define or modify the default keys used to execute functions for system escape, terminal activation, disconnect, and terminal pause. Generally, the keys used are actually combinations of keys, such as pressing the Control (Ctrl) key and another key (or keys) at the same time (such as Ctrl-^).

Sequences of keys, such as pressing the Control key and another key, then pressing yet another key, are also sometimes used (for example Ctrl-^, x). However, in each case these keys are referred to as characters, because each key or combination of keys is represented by a single ASCII character. For a complete list of available ASCII characters and their decimal and keyboard equivalents, see the "ASCII Character Set" appendix of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*.

Globally Defining Escape Character and Other Key Sequences

To define or change the default key sequences involved with terminal session activation, disconnection, escape, or pausing, use the following commands in line configuration mode, as needed:

Command	Purpose
`Router(config-line)# escape-character {ascii-number | ascii-character | break | default | none}`	Changes the system escape character. We recommend the use of the ASCII characters represented by the decimal numbers 1 through 30. The escape character can be a single character (such as ‘), a key combination (such as Ctrl-X), or a sequence of keys (such as Ctrl-^, X). The default escape character (key combination) is Ctrl-Shift-6 (Ctrl-^), or Ctrl-Shift-6, X (Ctrl-^, X).
`Router(config-line)# activation-character ascii-number`	Defines a session activation character. Entering this character at a vacant terminal begins a terminal session. The default activation character is the Return key.
`Router(config-line)# disconnect-character ascii-number`	Defines the session disconnect character. Entering this character at a terminal ends the session with the router. There is no default disconnect character.
`Router(config-line)# hold-character ascii-number`	Defines the hold character that causes output to the screen to pause. After this character has been set, a user can enter the character at any time to pause output to the terminal screen. To resume output, the user can press any key. To use the hold character in normal communications, precede it with the escape character. There is no default hold character.

For most of the commands described, you can reinstate the default value by using the **no** form. However, to return the escape character to its default, you should use the **escape-character default** line-configuration command.

Note If you are using the autoselect function (enabled using the **autoselect** line configuration command), the activation character should not be changed from the default value of Return. If you change this default, the autoselect feature may not function.

Defining Escape and Pause Characters for the Current Session

For the current terminal session, you can modify key sequences to execute functions for system escape and terminal pause. To modify these sequences, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router>` **`terminal escape-character`** *`ascii-number`*	Changes the system escape sequence for the current session. The escape sequence indicates that the codes that follow have special meaning. The default key combination is Ctrl-Shift-6 (Ctrl-^).
`Router>` **`terminal hold-character`** *`ascii-number`*	Defines the hold sequence or character that causes output to the terminal screen to pause for this session. There is no default sequence. To continue the output, type any character after the hold character. To use the hold character in normal communications, precede it with the escape character. You cannot suspend output on the console terminal.

The **terminal escape-character** EXEC command is useful, for example, if you have the default escape character defined for a different purpose in your keyboard file. Entering the escape character followed by the X key returns the router to EXEC mode when the router is connected to another device.

Specifying Telnet Operation Characteristics

To set Telnet operation characteristics for access servers, perform the tasks described in the following sections:

- Generating a Hardware Break Signal for a Reverse Telnet Connection
- Setting the Line to Refuse Full-Duplex, Remote Echo Connections
- Allowing Transmission Speed Negotiation
- Synchronizing the Break Signal
- Changing the End-of-Line Character

Note The commands in this section apply only to access servers.

Generating a Hardware Break Signal for a Reverse Telnet Connection

To cause the access server to generate a hardware Break signal on the EIA/TIA-232 line that is associated with a reverse Telnet connection for the current line and session, use the following command in EXEC mode:

Command	Purpose
`Router>` **`terminal telnet break-on-ip`**	Generates a hardware Break signal on the EIA/TIA-232 line that is associated with a reverse Telnet connection for the current line and session.

The hardware Break signal occurs when a Telnet Interrupt-Process command is received on that connection. This command can be used to control the translation of Telnet IP commands into X.25 Break indications.

This command is also a useful workaround in the following situations:

- Several user Telnet programs send an Interrupt-Process command, but cannot send a Telnet Break signal.
- Some Telnet programs implement a Break signal that sends an Interrupt-Process command.

Some EIA/TIA-232 hardware devices use a hardware Break signal for various purposes. A hardware Break signal is generated when a Telnet Break command is received.

Setting the Line to Refuse Full-Duplex, Remote Echo Connections

You can set the line to allow the Cisco IOS software to refuse full-duplex, remote echo connection requests from the other end. This refusal suppresses negotiation of the Telnet Remote Echo and Suppress Go Ahead options. To set the current line to refuse to negotiate full-duplex for the current session or remote echo options on incoming connections, use the following command in EXEC mode:

Command	Purpose
`Router>` **`terminal telnet refuse-negotiations`**	Sets the current line to refuse to negotiate full-duplex for the current session.

Allowing Transmission Speed Negotiation

To allow the Cisco IOS software to negotiate transmission speed for the current line and session, use the following command in EXEC mode:

Command	Purpose
`Router>` **`terminal telnet speed`** *`default-speed maximum-speed`*	Allows the Cisco IOS software to negotiate transmission speed for the current line and session.

You can match line speeds on remote systems in reverse Telnet, on host machines that connect to the network through an access server, or on a group of console lines hooked up to an access server when disparate line speeds are in use at the local and remote ends of the connection. Line speed negotiation adheres to the Remote Flow Control option, defined in RFC 1080.

Synchronizing the Break Signal

You can set lines on the access server to cause a reverse Telnet line to send a Telnet Synchronize signal when it receives a Telnet Break signal. The TCP Synchronize signal clears the data path, but interprets incoming commands. To cause the Cisco IOS software to send a Telnet Synchronize signal when it receives a Telnet Break signal on the current line and session, use the following command in EXEC mode:

Command	Purpose
`Router> terminal telnet sync-on-break`	Causes the Cisco IOS software to send a Telnet Synchronize signal when it receives a Telnet Break signal on the current line and session.

Changing the End-of-Line Character

The end of each line typed at the terminal is ended with a CR+LF (Carriage Return plus Line Feed) signal. The CR+LF signal is sent when a user presses Enter or Return. To cause the current terminal line to send a CR signal as a CR followed by a NULL instead of a CR followed by a line feed (LF), use the following command in EXEC mode:

Command	Purpose
`Router> terminal telnet transparent`	Causes the current terminal line to send a CR signal as a CR followed by a NULL instead of a CR followed by an LF.

This command ensures interoperability with different interpretations of end-of-line handling in the Telnet protocol specification.

Configuring Data Transparency for File Transfers

Data transparency enables the Cisco IOS software to pass data on a terminal connection without the data being interpreted as a control character.

During terminal operations, some characters are reserved for special functions. For example, the key combination Ctrl-Shift-6, X (^^x) suspends a session. When transferring files over a terminal connection (using the Xmodem or Kermit protocols, for example), you must suspend the recognition of these special characters to allow a file transfer. This process is called *data transparency.*

You can set a line to act as a transparent pipe so that programs such as Kermit, Xmodem, and CrossTalk can download a file across a terminal line. To temporarily configure a line to act as a transparent pipe for file transfers, use the following command in EXEC mode:

Command	Purpose
`Router> terminal download`	Configures the terminal line to act as a transparent pipe for file transfers.

The terminal download command is equivalent to using all the following commands:

- **terminal telnet transparent**
- **terminal no escape-character**

- **terminal no hold-character**
- **terminal no padding 0**
- **terminal no padding 128**
- **terminal parity none**
- **terminal databits 8**

Specifying an International Character Display

The classic U.S. ASCII character set is limited to 7 bits (128 characters), which adequately represents most displays in the U.S. Most defaults on the modem router work best on a 7-bit path. However, international character sets and special symbol display can require an 8-bit wide path and other handling.

You can use a 7-bit character set (such as ASCII), or you can enable a full 8-bit international character set (such as ISO 8859). This allows special graphical and international characters for use in banners and prompts, and adds special characters such as software flow control. Character settings can be configured globally, per line, or locally at the user level. Use the following criteria for determining which configuration mode to use when you set this international character display:

- If a large number of connected terminals support nondefault ASCII bit settings, use the global configuration commands.
- If only a few of the connected terminals support nondefault ASCII bit settings, use line configuration commands or the EXEC local terminal setting commands.

Note Setting the EXEC character width to an 8-bit character set can cause failures. If a user on a terminal that is sending parity enters the **help** command, an "unrecognized command" message appears because the system is reading all eight bits, although the eighth bit is not needed for **help**.

If you are using the **autoselect** function, the activation character should be set to the default Return, and the EXEC character bit should be set to 7. If you change these defaults, the application does not recognize the activation request.

Specifying the Character Display for All Lines

To specify a character set for all lines (globally), use one or both of the following commands in global configuration mode:

Command	Purpose
`Router(config)# default-value exec-character-bits {7 | 8}`	Specifies the character set used in command characters.
`Router(config)# default-value special-character-bits {7 | 8}`	Specifies the character set used in special characters such as software flow control, hold, escape, and disconnect characters.

Specifying the Character Display for a Line

To specify a character set based on hardware, software, or on a per-line basis, use any of the following commands in line configuration mode:

Command	Purpose
`Router(config-line)# databits {5 | 6 | 7 | 8}`	Sets the number of data bits per character that are generated and interpreted by hardware.
`Router(config-line)# data-character-bits {7 | 8}`	Sets the number of data bits per character that are generated and interpreted by software.
`Router(config-line)# exec-character-bits {7 | 8}`	Specifies the character set used in EXEC and configuration command characters on a per-line basis.
`Router(config-line)# special-character-bits {7 | 8}`	Specifies the character set used in special characters (such as software flow control, hold, escape, and disconnect characters) on a per-line basis.

Specifying the Character Display for the Current Session

To specify a character set based on hardware, software, or on a per-line basis for the current terminal session, use the following commands in EXEC mode:

Command	Purpose
`Router> terminal databits {5 | 6 | 7 | 8}`	Sets the number of data bits per character that are generated and interpreted by hardware for the current session.
`Router> terminal data-character-bits {7 | 8}`	Sets the number of data bits per character that are generated and interpreted by software for the current session.
`Router> terminal exec-character-bits {7 | 8}`	Specifies the character set used in EXEC and configuration command characters on a per-line basis for the current session.
`Router> terminal special-character-bits {7 | 8}`	Specifies the character set used in special characters (such as software flow control, hold, escape, and disconnect characters) on a per-line basis for the current session.

Setting Character Padding

Character padding adds a number of null bytes to the end of a line and can be used to make that line an expected length for conformity. You can change the character padding on a specific output character.

Setting Character Padding for a Line

To set character padding for a line, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# padding` *ascii-number count*	Sets padding on a specific output character for the specified line.

Changing Character Padding for the Current Session

To change character padding on a specific output character for the current session, use the following command in EXEC mode:

Command	Purpose
`Router> terminal padding ascii-number count`	Sets padding on a specific output character for the specified line for the current session.

Specifying the Terminal and Keyboard Type

You can specify the type of terminal connected to a line. This feature has two benefits: It provides a record of the type of terminal attached to a line, and it can be used in Telnet terminal negotiations to inform the remote host of the terminal type for display management.

Specifying the Terminal Type for a Line

To specify the terminal type for a line, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# terminal-type {terminal-type}`	Specifies the terminal type. Any string is accepted for the *terminal-type* argument.

This feature is used by TN3270 terminals to identify the keymap and ttycap passed by the Telnet protocol to the end host.

Specifying the Terminal and Keyboard Type for the Current Session

To specify the type of terminal connected to the current line for the current session, use the following command in EXEC mode:

Command	Purpose
`Router> terminal terminal-type terminal-type`	Specifies the terminal type for the current session.

Indicate the terminal type if it is different from the default of VT100. This default is used by TN3270 terminals for display management and by Telnet and rlogin to inform the remote host of the terminal type.

To specify the current keyboard type for a session, use the following command in EXEC mode:

Command	Purpose
`Router> terminal keymap-type keymap-name`	Specifies the keyboard type for the current session.

You must specify the keyboard type when you use a keyboard other than the default of VT100. The system administrator can define other keyboard types (using the **terminal-type** line configuration command) and provide these names to terminal users.

Changing the Terminal Screen Length and Width

By default, the Cisco IOS software provides a screen display of 24 lines by 80 characters. You can change these values if they do not meet the requirements of your terminal. The screen values you set are passed during rsh and rlogin sessions.

The screen values set can be learned by some host systems that use this type of information in terminal negotiation. To disable pausing between screens of output, set the screen length to 0.

The screen length specified can be learned by remote hosts. For example, the rlogin protocol uses the screen length to set terminal parameters on a remote UNIX host. The width specified also can be learned by remote hosts.

Setting the Terminal Screen Length and Width for a Line

To set the terminal screen length and width for all sessions on a line, use either of the following commands in line configuration mode, as needed:

Command	Purpose
`Router(config-line)# length screen-length`	Sets the screen length.
`Router(config-line)# width characters`	Sets the screen width.

Setting the Terminal Screen Length and Width for the Current Session

To set the number of lines or character columns on the current terminal screen for the current session, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router> terminal length screen-length`	Sets the screen length for the current session.
`Router> terminal width characters`	Sets the screen width for the current session.

Enabling Pending Output Notifications

You can enable the system to inform users when output is pending on a connection other than the active connection. This feature is for situations in which users are likely to have multiple, concurrent telnet connections through the system. For example, the user might want to know when another connection receives mail or a message.

Enabling Pending Output Notifications for a Line

To enable pending output notifications for a line, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# notify`	Enables a line to notify users of pending output on another connection.

Setting Pending Output Notification for the Current Session

To set pending output notification for the current session, use the following command in EXEC mode:

Command	Purpose
`Router> terminal notify`	Sets up a line to notify a user of pending output for the current session.

Creating Character and Packet Dispatch Sequences

The Cisco IOS software supports dispatch sequences and TCP state machines that send data packets only when they receive a defined character or sequence of characters. You can configure dispatch characters that allow packets to be buffered, then sent upon receipt of a character. You can configure a state machine that allows packets to be buffered, then sent upon receipt of a sequence of characters. This feature enables packet transmission when the user presses a function key, which is typically defined as a sequence of characters, such as Esc I C.

TCP state machines can control TCP processes with a set of predefined character sequences. The current state of the device determines what happens next, given an expected character sequence. The state-machine commands configure the server to search for and recognize a particular sequence of characters, then cycle through a set of states. The user defines these states—up to eight states can be defined. (Think of each state as a task that the server performs based on the assigned configuration commands and the type of character sequences received.)

The Cisco IOS software supports user-specified state machines for determining whether data from an asynchronous port should be sent to the network. This functionality extends the concept of the dispatch character and allows the equivalent of multicharacter dispatch strings.

Up to eight states can be configured for the state machine. Data packets are buffered until the appropriate character or sequence triggers the transmission. Delay and timer metrics allow for more efficient use of system resources. Characters defined in the TCP state machine take precedence over those defined for a dispatch character.

Setting Character and Packet Dispatch Sequences for a Line

To configure your system, use the following commands in line configuration mode:

Command	Purpose
`Router(config-line)# state-machine name state firstchar lastchar [nextstate | transmit]`	Specifies the transition criteria for the states in a TCP state machine.
`Router(config-line)# dispatch-machine name`	Specifies the state machine for TCP packet dispatch.
`Router(config-line)# dispatch-character ascii-number [ascii-number2 . . . ascii-number]`	Defines a character that triggers packet transmission.
`Router(config-line)# dispatch-timeout milliseconds`	Sets the dispatch timer.
`Router(config-line)# buffer-length length`	Specifies the maximum length of the data stream to be forwarded.

Changing the Packet Dispatch Character for the Current Session

To change the packet dispatch character for the current session, use the following command in EXEC mode:

Command	Purpose
`Router> terminal dispatch-character ascii-number1 [ascii-number2 . . . ascii-number]`	Defines a character that triggers packet transmission for the current session.

Changing Flow Control for the Current Session

To change flow control between the router and attached device for this session, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router> terminal flowcontrol {none | software [in | out] | hardware}`	Sets the terminal flow control for this session.
`Router> terminal start-character ascii-number`[1]	Sets the flow control start character in the current session.
`Router> terminal stop-character ascii-number`[1]	Sets the flow control stop character in the current session.

1. This command is seldom used. Typically, you only need to use the **terminal flowcontrol** command.

Enabling Session Locking

The **lock** EXEC command temporarily locks access to a session, denying access to other users. Session locking must be enabled on the line for the **lock** command to work. To allow session locking by users on a specific line or group of lines, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# lockable`	Enables a temporary terminal-locking mechanism.

Configuring Automatic Baud Rate Detection

You can configure a line to automatically detect the baud rate being used. To set up automatic baud rate detection, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# autobaud`	Configures a line to automatically detect the baud rate.

Note Do not use the **autobaud** command with the **autoselect** command.

To start communications using automatic baud detection, use multiple Returns at the terminal. A 600-, 1800-, or 19200-baud line requires three Returns to detect the baud rate. A line at any other baud rate requires only two Returns. If you use extra Returns after the baud rate is detected, the EXEC facility simply displays another system prompt.

Setting a Line as Insecure

You can set up a terminal line to appear as an insecure dialup line. The information is used by the local-area transport (LAT) software, which reports such dialup connections to remote systems.

To set a line as insecure, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# insecure`	Sets the line as a dialup line.

In early releases of Cisco IOS software, any line that used modem control was reported as dialup connection through the LAT protocol; this command allows more direct control of your line.

Configuring Communication Parameters for Terminal Ports

You can change the following parameters as necessary to meet the requirements of the terminal or host to which you are attached. To do so, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router> terminal {speed | txspeed | rxspeed} bps`	Sets the line speed for the current session. Choose from line speed, transmit speed, or receive speed.
`Router> terminal databits {5 | 6 | 7 | 8}`	Sets the data bits for the current session.
`Router> terminal stopbits {1 | 1.5 | 2}`	Sets the stop bits for the current session.
`Router> terminal parity {none | even | odd | space | mark}`	Sets the parity bit for the current session.

Displaying Debug Messages on the Console and Terminals

To display **debug** command output and system error messages in EXEC mode on the current terminal, use the following command in privileged EXEC mode:

Command	Purpose
`Router# terminal monitor`	Displays **debug** command output and system error messages in EXEC mode on the current terminal.

Remember that all terminal parameter-setting commands are set locally and do not remain in effect after a session is ended. You must use this command at the privileged-level EXEC prompt at each session to display the debugging messages.

Recording the Serial Device Location

You can record the location of a serial device. The text provided for the location appears in the output of the EXEC monitoring commands. To record the device location, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# location` *text*	Records the location of a serial device.

Changing the Retry Interval for a Terminal Port Queue

If you attempt to connect to a remote device such as a printer that is busy, the connection attempt is placed in a terminal port queue. If the retry interval is set too high, and several routers or other devices are connected to the remote device, your connection attempt can have long delays. To change the retry interval for a terminal port queue, use the following command in global configuration mode:

Command	Purpose
`Router(config)# terminal-queue entry-retry-interval` *interval*	Changes the retry interval for a terminal port queue.

Configuring LPD Protocol Support on a Printer

The Cisco IOS software supports a subset of the Berkeley UNIX Line Printer Daemon (LPD) protocol used to send print jobs between UNIX systems. This subset of the LPD protocol permits the following:

- Improved status information
- Cancellation of print jobs
- Confirmation of printing and automatic retry for common print failures
- Use of standard UNIX software

The Cisco implementation of LPD permits you to configure a printer to allow several types of data to be sent as print jobs (for example, PostScript or raw text).

To configure a printer for the LPD protocol, use the following command in global configuration mode:

Command	Purpose
`Router(config)# printer` *printername* `{line` *number* `| rotary` *number*`} [newline-convert]`	Configures a printer and specifies a tty line (or lines) for the device.

If you use the **printer** command, you also must modify the /etc/printcap file on the UNIX system to include the definition of the remote printer on the router. Use the optional **newline-convert** keyword on UNIX systems that do not handle single character line terminators to convert a new line to a character Return, line-feed sequence.

The following example includes the configuration of the printer named saturn on the host memphis:

```
comm1pt|Printer on cisco AccessServer:\
    :rm=memphis:rp+saturn:\
    :sd+/usr/spool/lpd/comm1pt:\
    :lf=?var/log/lpd/commlpt:
```

The content of the actual file may differ, depending on the configuration of your UNIX system.

To print, users use the standard UNIX lpr command.

Support for the LPD protocol allows you to display a list of currently defined printers and current usage statistics for each printer. To do so, use the following command in EXEC mode:

Command	Purpose
`Router> show printer`	Lists currently defined printers and their current usage statistics.

To provide access to LPD features, your system administrator must configure a printer and assign a TTY line (or lines) to the printer. The administrator must also modify the /etc/printcap file on your UNIX system to include the definition of the remote printer in the Cisco IOS software.

Managing Connections, Menus, and System Banners

This chapter describes how to manage connections to other hosts, set banner messages for router users, and create menus of specific user tasks.

For a complete description of the connections, menu, and system banner commands in this chapter, refer to the "Connection, Menu, and System Banner Commands" chapter in the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Managing Connections, Menus, and System Banners Task List

To manage connections, configure messages and banners, and create user menus, perform any of the tasks described in the following sections, as needed. All tasks in this chapter are optional.

- Managing Connections
- Configuring Terminal Messages
- Configuring Terminal Banners
- Creating Menus

Examples for these sections can be found at the end of the chapter in the "Connection Management, System Banner, and User Menu Configuration Examples" section.

Managing Connections

To configure connection-management activities that apply to all supported connection protocols, perform the tasks described in the following sections. All tasks are optional.

- Displaying Current Terminal Characteristics
- Escaping Terminal Sessions and Switching to Other Connections
- Assigning a Logical Name to a Connection
- Changing a Login Name
- Locking Access to a Terminal
- Sending Messages to Other Terminals
- Clearing TCP Connections
- Exiting a Session Started from a Router
- Logging Out of a Router
- Disconnecting a Line

Displaying Current Terminal Characteristics

To display the current settings for the terminal line connection, use the following command in EXEC mode:

Command	Purpose
`Router# show terminal`	Displays current settings for the terminal.

The following example shows sample output:

```
AccessServer1> show terminal

Line 2, Location: "", Type: "VT220"
Length: 24 lines, Width: 80 columns
Baud rate (TX/RX) is 9600/9600
Status: PSI Enabled, Ready, Active, No Exit Banner
Capabilities: none
Modem state: Ready
Group codes:    0
Special Chars: Escape  Hold  Stop  Start  Disconnect  Activation
                ^^x    none   -     -       none
Timeouts:      Idle EXEC    Idle Session   Modem Answer  Session   Dispatch
               00:10:00        never                        none     not set
                            Idle Session Disconnect Warning
                              never
                            Login-sequence User Response
                             00:00:30
                            Autoselect Initial Wait
                              not set
Modem type is unknown.
Session limit is not set.
Time since activation: 00:01:07
Editing is enabled.
History is enabled, history size is 10.
DNS resolution in show commands is enabled
```

```
Full user help is disabled
Allowed transports are lat pad v120 mop telnet rlogin nasi.  Preferred is lat.
No output characters are padded
No special data dispatching characters
```

Escaping Terminal Sessions and Switching to Other Connections

After you have started a connection, you can escape out of the current terminal session by using the escape key sequence (Ctrl-Shift-6 then X by default). You can type the command character as you hold down the Ctrl key or with the Ctrl key released; you can type either uppercase or lowercase letters.

Note In screen output examples that show two caret (^^) symbols together, the first caret represents the Control key (Ctrl) and the second caret represents the key sequence Shift-6. The double-caret combination (^^) means hold down the Ctrl key while you press the Shift and the 6 key.

By default, the escape key sequence is Ctrl-Shift-6, X. However, the escape key sequence can be changed using the **escape-character** line configuration command. To determine the current setting for the escape character, use the **show terminal** EXEC command.

You can have several concurrent sessions open and switch back and forth between them.

The number of sessions that can be open at one time is defined by the **session-limit** command.

To switch between sessions by escaping one session and resuming a previously opened session, perform the following steps:

Step 1 Escape out of the current session by pressing the escape key sequence (Ctrl-Shift-6 then X [Ctrl^, X] by default) and return to the EXEC prompt.

Step 2 Enter the **where** command to list the open sessions. All open sessions associated with the current terminal line are displayed.

Step 3 Enter the **resume** command and the session number to make the connection.

You also can resume the previous session by pressing the Return key.

The Ctrl^, X key combination and the **where** and **resume** EXEC commands are available with all supported connection protocols (for example, Telnet).

Assigning a Logical Name to a Connection

To assign a logical name to a connection, use the following command in EXEC mode:

Command	Purpose
`Router# name-connection`	Assigns a logical name to a connection.

The logical name can be useful for keeping track of multiple connections.

You are prompted for the connection number and name to assign. The **where** EXEC command displays a list of the assigned logical connection names.

Changing a Login Name

You can change your login username if you must match outgoing access list requirements or other login prompt requirements. A login server must be running and available to use this command. To change a login username, use the following command in user EXEC mode:

Command	Purpose
`Router> login`	Allows you to log in to the system a second time for the purposes of changing your login name.

When you enter this command, the system prompts you for a username and password. Enter the new username and the original password. If the username does not match, but the password does, the Cisco IOS software updates the session with the new username used by the **login** command attempt. For example, assume that a user logged in as user1 needs to change the login name to user2:

```
Router> login
Username: user2
Password: <letmein>
Router>
```

In this example, the password letmein is the same password used at the initial login. (The angle brackets in the example indicate that the password is not displayed on the screen when entered.) At the second `Router>` prompt, the user is now logged in as user2.

If no username and password prompts appear, the network administrator did not specify that a username and password be required at login time. If both the username and password are entered correctly, the session becomes associated with the specified username.

To access a system with Terminal Access Controller Access Control System (TACACS) security, enter your login name or specify a TACACS server by using the *user@tacacs-server* syntax when the "Username:" prompt appears, as shown in the following steps:

	Command	Purpose
Step 1	`Router> login`	Allows you to log in to the system a second time for the purposes of changing your login name.
Step 2	`Username: user@tacacs-server`	Specifies the new username and authenticates the name with the server specified with the *tacacs-server* argument.
Step 3	`Password: <password>`	Specifies the TACACS password for the username specified in Step 2.

Only the specified host (tacacs-server) is accessed for user authentication information.

In the following example, user2 specifies the TACACS host host1 to authenticate the password:

```
Router> login
Username: user2@host1
Translating "HOST1"...domain server (131.108.1.111) [OK]
Password: <letmein2>
```

If you do not specify a host, the router tries each of the TACACS servers in the list until it receives a response. If you specify a host that does not respond, no other TACACS server will be queried. The router either will deny access or function, according to the action specified by the **tacacs-server**

last-resort global configuration command, if it is configured. If you specified a TACACS server host with the *user@tacacs-server* argument, the TACACS server specified is used for all subsequent authentication or notification queries, with the possible exception of Serial Line Internet Protocol (SLIP) address queries.

For more information on configuring TACACS, refer to the **tacacs-server host** global configuration command in the "TACACS, Extended TACACS, and TACACS+ Commands" chapter of the *Cisco IOS Security Command Reference*.

For an example of changing a login name, see the "Changing a Login Name Example" section at the end of this chapter.

Locking Access to a Terminal

You can prevent access to your terminal session while keeping your connection open by setting a temporary password. For this temporary locking feature to work, the line must first be configured to allow locking (using the **lockable** line-configuration mode command). To lock access to the terminal, perform the following steps:

Step 1 Issue the **lock** command in user or privileged EXEC mode.

When you issue this command, the system will prompt you for a password.

Step 2 Enter a password, which can be any arbitrary string. The system will prompt you to confirm the password. The screen then is cleared, and the message "Locked" is displayed.

Step 3 To regain access to your session, reenter the password.

The Cisco IOS software honors session timeouts on locked lines. You must clear the line to remove this feature.

The following is an example of the prompts displayed after the **lock** command is entered. Note that the entered password does not appear on screen.

```
Router# lock
Password:
Again:
                        Locked
Password:
Router#
```

Sending Messages to Other Terminals

You can send messages to one or all terminal lines. A common reason for doing this is to inform users of an impending shutdown. To send a message to other terminals, use the following command in EXEC mode:

Command	Purpose
`Router# send {line-number | *}`	Sends a message to other terminals.

The system prompts for the message, which can be up to 500 characters long. Press **Ctrl-Z** to end the message. Press **Ctrl-C** to abort the command.

Clearing TCP Connections

To clear a Transmission Control Protocol (TCP) connection, use the following command in privileged EXEC mode:

Command	Purpose
Router# **clear tcp** {**line** *line-number* | **local** *host-name port* **remote** *host-name port* | **tcb** *tcb-address*}	Clears a TCP connection.

The **clear tcp** command is particularly useful for clearing hung TCP connections.

The **clear tcp line** *line-number* command terminates the TCP connection on the specified tty line. All TCP sessions initiated from that tty line are also terminated.

The **clear tcp local** *host-name port* **remote** *host-name port* command terminates the specific TCP connection identified by the host name/port pair of the local and remote router.

Exiting a Session Started from a Router

The protocol used to initiate a session determines how you exit that session.

To exit from SLIP and PPP connections, you must hang up the dial-in connection, usually with a command that your dial-in software supports.

To exit a local area transport (LAT), Telnet, rlogin, TN3270, or X.3 packet assembler/disassembler (PAD) session begun from the router to a remote device, press the escape key sequence (Ctrl-Shift-6 then X [Ctrl^X] by default for some systems, Ctrl-Z by default for other systems) and enter the **disconnect** command at the EXEC prompt. You can also log out of the remote system.

You can use either the **exit** or **logout** command in EXEC mode to terminate an active terminal session.

To exit a Telnet session *to* a router, see the "Logging Out of a Router" section, which follows.

Logging Out of a Router

The method you use to disconnect from a router depends on where you are located in relation to the router, and the port on the router to which you log in.

If your terminal or computer running a terminal-emulation application is remotely connected to the console port of the router, you disconnect by issuing the command or key sequence used by your terminal-emulation package. For example, if you are on a Macintosh computer running the application TCP/Connect from InterCon Corporation, you would press Ctrl-] at the user or privileged EXEC prompt to disconnect.

If you are on a remote terminal and connect to a vty through a synchronous interface on the router, you can issue one of the following commands in EXEC mode to log out:

- **exit**
- **logout**

Disconnecting a Line

Note Avoid disconnecting a line to end a session. Instead, log out of the host to allow the router to clear the connection. You should disconnect a line only if you cannot log out of an active session (for example, if the line is stuck or frozen).

To disconnect a line, use the following command in EXEC mode:

Command	Purpose
`Router# disconnect [connection]`	Disconnects a line.

If your terminal or computer running a terminal-emulation application is connected physically to the console port of the router, you can also disconnect from the router by physically disconnecting the cable from the console port of the router.

Configuring Terminal Messages

To configure messages that can be displayed to terminal users that connect to the system, perform any of the tasks found in the following sections. All tasks are optional.

- Configuring an Idle Terminal Message
- Configuring a “Line in Use” Message
- Configuring a “Host Failed” Message

Configuring an Idle Terminal Message

You can configure the system to display a message when a console or terminal not in use. Also called a *vacant message*, this message is different from the banner message displayed when a user logs in to the system. To enable the idle terminal message, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# vacant-message [d message d]`	Configures the system to display an idle terminal message. The argument *d* indicates any delimiting character.

Tips Commands requiring a delimiting character (the *d* argument) are used throughout this chapter. Any character can be used as the delimiting character, but we recommend the use of the quote sign ("), because this character is unlikely to be needed within the message itself. Other commonly used delimiting characters include the percent sign (%) or the forward slash (/), but because these characters have meanings within certain Cisco IOS commands, they are not recommended. For example, to set the vacant message to `This terminal is idle` you would enter the command **vacant-message " This terminal is idle "**.

Configuring a "Line in Use" Message

To configure the system to display a "line in use" message when an incoming connection is attempted and all rotary group or other lines are in use, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# refuse-message d message d`	Configures the system to display a "line in use" message. The argument *d* indicates any delimiting character.

If you do not define such a message, the user receives a system-generated error message when all lines are in use. You also can use this message to provide the user with further instructions.

Configuring a "Host Failed" Message

To configure the system to display a "host failed" message when a Telnet connection with a specific host fails, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# busy-message hostname d message d`	Configures the system to display a "host failed" message. The argument *d* indicates any delimiting character.

Configuring Terminal Banners

Banners are informational messages that can be displayed to users. To enable terminal banners, perform any of the tasks in the following sections. All tasks are optional.

- Configuring a Message-of-the-Day Banner
- Configuring a Login Banner
- Configuring an EXEC Banner
- Configuring an Incoming Banner
- Configuring a SLIP-PPP Banner Message
- Enabling or Disabling the Display of Banners

For an example of displaying terminal banner messages, see the "Configuring Banners Example" section at the end of this chapter.

Using Banner Tokens

Banners can be customized with the use of banner tokens. Tokens are keywords in the form $(*token*) that, when used in a banner message, display the currently configured value of the token argument (for example, the router host name, domain name, or IP address). Using these tokens, you can design your own banners that will display current Cisco IOS configuration variables. Only Cisco IOS supported tokens may be used. There is no facility for you to define your own tokens.

Table 10 lists the tokens supported by the different **banner** commands.

Table 10 Tokens Allowed by Banner Type

Token	Description	motd banner	login banner	exec banner	incoming banner	slip-ppp banner
$(hostname)	Router Host Name	Yes	Yes	Yes	Yes	Yes
$(domain)	Router Domain Name	Yes	Yes	Yes	Yes	Yes
$(peer-ip)	IP Address of the Peer Machine	No	No	No	No	Yes
$(gate-ip)	IP Address of the Gateway Machine	No	No	No	No	Yes
$(encap)	Encapsulation Type (SLIP or PPP)	No	No	No	No	Yes
$(encap-alt)	Encapsulation Type Displayed as SL/IP instead of SLIP	No	No	No	No	Yes
$(mtu)	Maximum Transmission Unit Size	No	No	No	No	Yes
$(line)	vty or tty (async) Line Number	Yes	Yes	Yes	Yes	No
$(line-desc)	User-specified description of the Line	Yes	Yes	Yes	Yes	No

Configuring a Message-of-the-Day Banner

You can configure a message-of-the-day (MOTD) banner to be displayed on all connected terminals. This banner is displayed at login and is useful for sending messages that affect all network users (such as impending system shutdowns). To do so, use the following command in global configuration mode:

Command	Purpose
`Router(config)# banner motd` *`d message d`*	Configures the system to display a message-of-the-day banner. The argument *d* indicates any delimiting character.

Configuring a Login Banner

You can configure a login banner to be displayed on all connected terminals. This banner is displayed after the MOTD banner appears and before the login prompts.

To configure a login banner, use the following command in global configuration mode:

Command	Purpose
`Router(config)# banner login d message d`	Configures the system to display a banner before the username and password login prompts. The argument *d* indicates any delimiting character.

The login banner cannot be disabled on a per-line basis. To globally disable the login banner, you must delete the login banner with the **no banner login** command.

Configuring an EXEC Banner

You can configure a banner to be displayed whenever a EXEC process is initiated. For example, this banner will be displayed to a user Telneting to the system after entering their username and password, but before the user EXEC mode prompt is displayed. To configure an EXEC banner, use the following command in global configuration mode:

Command	Purpose
`Router(config)# banner exec d message d`	Configures the system to display a banner whenever an EXEC process is initiated. The argument *d* indicates any delimiting character.

Configuring an Incoming Banner

You can configure a banner to be displayed on terminals connected to reverse Telnet lines. This banner is useful for providing instructions to users of these types of connections. Reverse Telnet connections are described in more detail in the "Configuring and Managing External Modems" chapter of the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

To configure a banner that is sent on incoming connections, use the following command in global configuration mode:

Command	Purpose
`Router(config)# banner incoming d message d`	Configures the system to display a banner when there is an incoming connection to a terminal line from a host on the network. The argument *d* indicates any delimiting character.

Configuring a SLIP-PPP Banner Message

Default banner messages have been known to cause connectivity problems in some non-Cisco SLIP and PPP dialup software. You can now customize the SLIP-PPP banner message to make Cisco SLIP and PPP compatible with non-Cisco dialup software. To configure a SLIP-PPP banner message, use the following command in global configuration mode:

Command	Purpose
`Router(config)# banner slip-ppp d message d`	Configures a SLIP-PPP banner to display a customized message. The argument *d* indicates any delimiting character.

Enabling or Disabling the Display of Banners

You can control display of the MOTD and line-activation (EXEC) banners. By default, these banners are displayed on all lines. To suppress or reinstate the display of such banners, use the following commands in line configuration mode, as needed:

Command	Purpose
`Router(config-line)# no exec-banner`	Suppresses the display of MOTD and EXEC banners.
`Router(config-line)# exec-banner`	Reinstates the display of the EXEC or MOTD banners.
`Router(config-line)# no motd-banner`	Suppresses the display of MOTD banners.
`Router(config-line)# motd-banner`	Reinstates the display of the MOTD banners.

These commands determine whether the router will display the EXEC banner and the MOTD banner when an EXEC session is created. These banners are defined with the **banner motd** and **banner exec** global configuration commands. By default, the MOTD banner and the EXEC banner are enabled on all lines.

Disable the EXEC and MOTD banners using the **no exec-banner** command.

The MOTD banners can also be disabled by the **no motd-banner** line configuration command, which disables MOTD banners on a line. If the **no exec-banner** command is configured on a line, the MOTD banner will be disabled regardless of whether the **motd-banner** command is enabled or disabled. Table 11 summarizes the effects of the combination of the **exec-banner** command and the **motd-banner** command.

Table 11 Banners Displayed by exec-banner and motd-banner Command Combinations

	exec-banner (default)	**no exec-banner**
motd-banner (default)	MOTD banner EXEC banner	None
no motd-banner	EXEC banner	None

For reverse Telnet connections, the EXEC banner is never displayed. Instead, the incoming banner is displayed. The MOTD banner is displayed by default, but it is disabled if either the **no exec-banner** command or **no motd-banner** command is configured. Table 12 summarizes the effects of the combination of the **exec-banner** command and the **motd-banner** command for reverse Telnet connections.

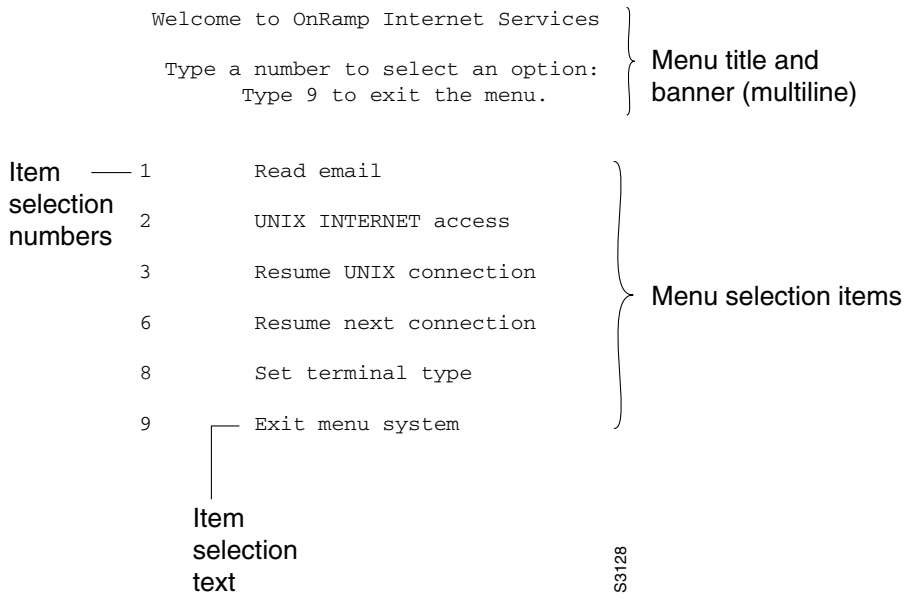
Table 12 Banners Displayed Based On exec-banner and motd-banner Command Combinations for Reverse Telnet Sessions to Async Lines

	exec-banner (default)	**no exec-banner**
motd-banner (default)	MOTD banner Incoming banner	Incoming banner
no motd-banner	Incoming banner	Incoming banner

Creating Menus

A menu is a displayed list of actions from which a user can select without needing to know anything about the underlying command-level details. A menu system (also known as a user menu) effectively controls the functions a user can access. Figure 6 illustrates the parts that make up a typical menu.

Figure 6 Typical Menu Example



Any user that can enter configuration mode can create menus. Remember the following guidelines when you create menus:

- Each menu item represents a single user command.
- The menu system default is a standard “dumb” terminal that only displays text in a 24-line-by-80-column format.
- A menu can have no more than 18 menu items. Menus containing more than 9 menu items are automatically configured as single-spaced menus; menus containing 9 or fewer menu items are automatically configured as double-spaced menus, but can be configured as single-spaced menus using the **menu single-space** global configuration command. (For more information about menu display configuration options, see the section “Specifying Menu Display Configuration Options” later in this chapter.)
- Item keys can be numbers, letters, or strings. If you use strings, you must configure the **menu line-mode** global configuration command.
- When you construct a menu, always specify how a user exits a menu and where the user goes. If you do not provide an exit from a menu—such as with the **menu-exit** command (described in the section “Specifying the Underlying Command for the Menu Item” later in this chapter), the user will be trapped.

The **exec-timeout** line configuration command can be used to close and clean up an idle menu; the **session-timeout** command can be used to clean up a menu with an open connection.

Creating a Menu Task List

To create menus, perform the tasks described in the following sections:

- Specifying the Menu Title (Required)
- Specifying the Menu Prompt (Optional)
- Specifying the Menu Item Text (Required)
- Specifying the Underlying Command for the Menu Item (Required)
- Specifying the Default Command for the Menu (Required)
- Creating a Submenu (Optional)
- Creating Hidden Menu Entries (Optional)
- Specifying Menu Display Configuration Options (Optional)
- Specifying per-Item Menu Options (Optional)
- Invoking the Menu (Required)
- Deleting the Menu from the Configuration (Optional)

Specifying the Menu Title

You can specify an identifying title for the menu. To specify the menu title, use the following command in global configuration mode:

Command	Purpose
`Router(config)# menu` *menu-name* `title` *d title d*	Specifies the title for the menu. The argument *d* indicates any delimiting character.

The following example specifies the title that is displayed when the OnRamp menu is selected. The following four main elements create the title:

- The **menu title** command
- Delimiter characters that open and close the title text
- Escape characters to clear the screen (optional)
- Title text

The following example shows the command used to create the title for the menu shown in Figure 6:

```
Router(config)# menu OnRamp title %^[[H^[[J
Enter TEXT message.  End with the character '%'.
        Welcome to OnRamp Internet Services

        Type a number to select an option;
              Type 9 to exit the menu.
%
Router(config)#
```

You can position the title of the menu horizontally by preceding the title text with blank characters. You can also add lines of space above and below the title by pressing Enter.

In this example, the title text consists of the following elements:

- One-line title
- Space
- Two-line menu instruction banner

Title text must be enclosed within text delimiter characters—the percent sign character (%) in this example. Title text delimiters are characters that do not ordinarily appear within the text of a title, such as slash (/), double quote ("), or tilde (~). You can use any character that is not likely to be used within the text of the title as delimiter characters. Ctrl-C is reserved for special use and should not be used in the text of the title.

This title text example also includes an escape character sequence to clear the screen before displaying the menu. In this case the string `^[[H^[[J` is an escape string used by many VT100-compatible terminals to clear the screen. To enter it, you must enter Ctrl-V before each escape character (^[).

You can also use the **menu clear-screen** global configuration command to clear the screen before displaying menus and submenus, instead of embedding a terminal-specific string in the menu title. This option uses a terminal-independent mechanism based on termcap entries defined in the router and the terminal type configured for the user terminal. The **menu clear-screen** command allows the same menu to be used on multiple types of terminals instead of terminal-specific strings being embedded within menu titles. If the termcap entry does not contain a clear string, the menu system inserts 24 new lines, causing all existing text to scroll off the top of the terminal screen.

To clear the screen before displaying the menu, use the following command in global configuration mode:

Command	Purpose
`Router(config)# menu` *menu-name* `clear-screen`	Specifies screen clearing before displaying menus and submenus.

The following example clears the screen before displacing the OnRamp menu or a submenu:

```
Router(config)# menu OnRamp clear-screen
```

Specifying the Menu Prompt

To specify a menu prompt, use the following command in global configuration mode:

Command	Purpose
`Router(config)# menu` *menu-name* `prompt` *d prompt d*	Specifies the prompt for the menu. The argument *d* indicates any delimiting character.

Specifying the Menu Item Text

Each displayed menu entry consists of the selection key (number, letter, or string) and the text describing the action to be performed. You can specify descriptive text for a maximum number of 18 menu items. Because each menu entry represents a single user interface command, you must specify the menu item text one entry at a time. To specify the menu item text, use the following command in global configuration mode:

Command	Purpose
`Router(config)# menu` *menu-name* `text` *menu-item menu-text*	Specifies the text for the menu item.

The following example specifies the text that is displayed for the three entries in the OnRamp menu:

```
Router(config)# menu OnRamp text 1 Read email
Router(config)# menu OnRamp text 2 UNIX Internet Access
Router(config)# menu OnRamp text 9 Exit menu system
```

You can provide access to context-sensitive help by creating a "help server" host and using a menu entry to make a connection to that host.

Menu selection keys need not be contiguous. You can provide consistency across menus by assigning a particular number, letter, or string to a special function—such as Help or Exit—regardless of the number of menu entries in a given menu. For example, menu entry H could be reserved for help across all menus.

When more than nine menu items are defined in a menu, the **menu line-mode** and **menu single-space** global configuration commands are activated automatically. The commands can be configured explicitly for menus of nine items or fewer. For more information on these commands, see the section "Specifying Menu Display Configuration Options" later in this chapter.

Specifying the Underlying Command for the Menu Item

Each displayed menu entry issues a user interface command when the user enters its key. Each menu entry can have only a single command associated with it. To specify the underlying menu item command, use the following command in global configuration mode:

Command	Purpose
`Router(config)# menu` *menu-name* `command` *menu-item command*	Specifies the command to be performed when the menu item is selected.

The following example specifies the commands that are associated with the three entries in the OnRamp menu:

```
Router(config)# menu OnRamp command 1 rlogin mailsys
Router(config)# menu OnRamp command 2 rlogin unix.cisco.com
Router(config)# menu OnRamp command 9 menu-exit
```

The **menu-exit** command is available only from within menus. This command provides a way to return to a higher-level menu or to exit the menu system.

When a menu item allows you to make a connection, the menu item should also contain entries that can be used to resume connections; otherwise, when you try to escape from a connection and return to the menu, there is no way to resume the session. It will sit idle until you log out.

You can build the **resume connection** EXEC command into a menu entry so that the user can resume a connection, or you can configure the line using the **escape-char none** command to prevent users from escaping their sessions.

To specify connection resumption as part of the menu item command, use the following command in global configuration mode:

Command	Purpose
`Router(config)# menu` *menu-name* `command` *menu-item* `resume` [*connection*] `/connect` [*connect string*]	Specifies that the **resume** command will be performed when the menu item is selected.

Embedding the **resume** command within the **menu** command permits a user to resume the named connection or make another connection using the specified name, if there is no active connection by that name. As an option, you can also supply the connect string needed to connect initially. When you do not supply this connect string, the command uses the specified connection name.

You can use the **resume** command in the following menu entries:

- Embedded in a menu entry
- As a separate, specific menu entry
- As a "rotary" menu entry that steps through several connections

In the following example, the **resume** command is embedded in the **menu** command so that selecting the menu item either starts the specified connection session (if one is not already open) or resumes the session (if one is already open):

```
Router(config)# menu Duluth text 1 Read email
Router(config)# menu Duluth command 1 resume mailsys /connect rlogin mailsys
```

In the following example, the **resume** command is used in a separate menu entry (entry 3) to resume a specific connection:

```
Router(config)# menu Duluth text 3 Resume UNIX Internet Access
Router(config)# menu Duluth command 3 resume unix.cisco.com
```

You use the **resume/next** command to resume the next open connection in the user list of connections. This command allows you to create a single menu entry that advances through all of the user connections. To specify **resume/next** connection resumption as part of the menu item command, use the following command in global configuration mode:

Command	Purpose
`Router(config)# menu` *menu-name* `command` *menu-item* `resume/next`	Specifies **resume/next** connection resumption.

The following example shows a menu entry (entry 6) created to advance through all of the user connections:

```
Router(config)# menu Duluth text 6 Resume next connection
Router(config)# menu Duluth command 6 resume/next
```

Specifying the Default Command for the Menu

When a user presses Enter without specifying an item, the router performs the command for the default item. To specify the default item, use the following command in global configuration mode:

Command	Purpose
`Router(config)# menu` menu-*name* `default` *menu-item*	Specifies the command to be performed when the menu users does not select a menu item.

Creating a Submenu

To create submenus that are opened by selecting a higher-level menu entry, use the **menu** command to invoke a menu in a line menu entry. To specify a submenu item command, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# menu` *menu-name* `text` *menu-item menu-text*	Specifies the menu item that invokes the submenu.
Step 2	`Router(config)# menu` *menu-name* `command` *menu-item* `menu` *menu-name2*	Specifies the command to be used when the menu item is selected.
Step 3	`Router(config)# menu` *menu-name* `title` *delimiter menu-title delimiter*	Specifies the title for the submenu.
Step 4	`Router(config)# menu` *menu-name* `text` *menu-item menu-text*	Specifies the submenu item.
Step 5	`Router(config)# menu` *menu-name* `command` *menu-item command*	Specifies the command to be used when the submenu item is selected. Repeat this command as needed.

The following example specifies that the menu item (entry 8) activates the submenu in the OnRamp menu:

```
Router(config)# menu OnRamp text 8 Set terminal type
```

The following example specifies the command that is performed when the menu item (entry 8) is selected in the OnRamp menu:

```
Router(config)# menu OnRamp command 8 menu Terminals
```

The following example specifies the title for the Terminals submenu:

```
Router(config)# menu Terminals title /
               Supported Terminal Types

         Type a number to select an option;
        Type 9 to return to the previous menu.
```

The following example specifies the submenu items for the Terminals submenu:

```
Router(config)# menu Terminals text 1 DEC VT420 or similar
Router(config)# menu Terminals text 2 Heath H-19
Router(config)# menu Terminals text 3 IBM 3051 or equivalent
Router(config)# menu Terminals text 4 Macintosh with gterm emulator
Router(config)# menu Terminals text 9 Return to previous menu
```

The following example specifies the commands associated with the items in the Terminals submenu:

```
Router(config)# menu Terminals command 1 term terminal-type vt420
Router(config)# menu Terminals command 2 term terminal-type h19
Router(config)# menu Terminals command 3 term terminal-type ibm3051
Router(config)# menu Terminals command 4 term terminal-type gterm
Router(config)# menu Terminals command 9 menu-exit
```

When you select entry 8 on the main menu, the following Terminals submenu appears:

```
      Supported Terminal Types

  Type a number to select an option;
Type 9 to return to the previous menu.

1     DEC VT420 or similar

2     Heath H-19

3     IBM 3051 or equivalent

4     Macintosh with gterm emulator

9     Return to previous menu
```

Note If you nest too many levels of menus, the system displays an error message on the terminal and returns to the previous menu level.

Creating Hidden Menu Entries

A hidden menu entry is a menu item that contains a selection key but no associated text describing the action to be performed. Include this type of menu entry to aid system administrators that provide help to users. The normal procedure is to specify a menu command but omit specifying any text for the item. To specify a hidden menu item, use the following command in global configuration mode:

Command	Purpose
Router(config)# **menu** *menu-name* **command** *menu-item command*	Specifies the command to be used when the hidden menu entry is selected.

The following example shows the command associated with the submenu entry in the OnRamp menu:

```
Router(config)# menu OnRamp command 7 show whoami
```

If additional text is appended to the **show whoami** command, that text is displayed as part of the data about the line. For example, the hidden menu entry created by the command

```
Router(config)# menu OnRamp command 7 show whoami Terminals submenu of OnRamp Internet
Access menu
```

will display information similar to the following:

```
Comm Server "cs101", Line 0 at 0 bps. Location "Second floor, West"
Additional data: Terminals submenu of OnRamp Internet Access menu
```

To prevent the information from being lost if the menu display clears the screen, this command always displays a --More-- prompt before returning.

Specifying Menu Display Configuration Options

In addition to the **menu clear-screen** global configuration command (described in the "Specifying the Menu Title" section) the following are the three other **menu** commands that define menu functions:

- **menu line-mode**
- **menu single-space**
- **menu status-line**

Using Line Mode in Menus

In a menu of nine or fewer items, you ordinarily select a menu item by entering the item number or a letter. In line mode, you select a menu entry by entering the item key and pressing Enter. The line mode allows you to backspace over the selection and enter another before pressing Enter to issue the command. This function allows you to change the selection before you invoke the command.

To configure the menu to operate in line mode, use the following command in global configuration mode:

Command	Purpose
`Router(config)# menu` *menu-name* `line-mode`	Configures the menu to use line mode for entering menu items.

The line-mode option is invoked automatically when more than nine menu items are defined, but it can also be configured explicitly for menus of nine items or fewer.

In order to use strings as selection keys, you must enable the **menu line-mode** command.

Displaying Single-Spaced Menus

If there are nine or fewer menu items, the Cisco IOS software ordinarily displays the menu items double-spaced. In a menu of more than nine items, the **single-space** option is activated automatically to fit the menu into a normal 24-line terminal screen. However, the single-space option also can be configured explicitly for menus of nine or fewer items.

To invoke the **single-space** option, use the following command in global configuration mode:

Command	Purpose
`Router(config)# menu` *menu-name* `single-space`	Configures the specified menu to display single-spaced.

Displaying an Informational Status Line

The **status-line** option displays a line of status information about the current user at the top of the terminal screen before the menu title is displayed. This status line includes the router host name, the user line number, and the current terminal type and keymap type (if any).

To display the **status-line** option, use the following command in global configuration mode:

Command	Purpose
`Router(config)# menu` *menu-name* `status-line`	Configures the specified menu to display a status line.

Specifying per-Item Menu Options

To configure per-item options, use the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)# menu` *menu-name* `options` *menu-item* `pause`	Configures the system to pause after the specified menu item is selected by the user. Enter this command once for each menu item that pauses.
`Router(config)# menu` *menu-name* `options` *menu-item* `login`	Configures the specified menu item to require a login before executing the command. Enter this command once for each menu item that requires a login.

Invoking the Menu

To invoke (access) a configured menu, use the following command in EXEC mode:

Command	Purpose
`Router# menu` *menu-name*	Invokes a preconfigured user menu.

You can define menus containing privileged EXEC commands, but users must have privileged access when they start up the menu.

To ensure that a menu is automatically invoked on a line, make sure the menu does not have any exit paths that leave users in an interface they cannot operate, then configure that line with the **autocommand menu** *menu-name* line configuration command. (The **autocommand menu** *menu-name* command configures the line to automatically execute the **menu** *menu-name* command when a user initiates a connection over that line.)

Menus also can be invoked on a per-user basis by defining an **autocommand** command for that local username.

In the following example, the OnRamp menu is invoked:

```
Router# menu OnRamp

      Welcome to OnRamp Internet Services

       Type a number to select an option;
            Type 9 to exit the menu.

1     Read email

2     UNIX Internet access

3     Resume UNIX connection


6     Resume next connection

9     Exit menu system
```

Deleting the Menu from the Configuration

To delete the menu from the configuration, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no menu` *menu-name*	Deletes the menu by specifying the menu name.

In order to use the menu again, you must reconfigure the entire menu.

The following example deletes the OnRamp menu from the configuration:

```
Router(config)# no menu OnRamp
```

Connection Management, System Banner, and User Menu Configuration Examples

This section provides the following examples:

- Changing a Login Name Example
- Sending Messages to Other Terminals Example
- Clearing a TCP/IP Connection Example
- Configuring Banners Example
- Setting a SLIP-PPP Banner with Banner Tokens Example
- Configuring a Menu Example

Changing a Login Name Example

The following example shows how login usernames and passwords can be changed. In this example, a user currently logged in under the username user1 attempts to change that login name to user2. After entering the **login** command, the user enters the new username, but enters an incorrect password. Because the password does not match the original password, the system rejects the attempt to change the username.

```
Router> login
Username: user2
Password:
% Access denied
Still logged in as "user1"
```

Next, the user attempts the login change again, with the username user2, but enters the correct (original) password. This time the password matches the current login information, the login username is changed to user2, and the user is allowed access to the EXEC at the user-level.

```
Router> login
Username: user2
Password:
Router>
```

Sending Messages to Other Terminals Example

The following example shows the process of sending a message to all terminal connections on the router:

```
Router# send *
Enter message, end with CTRL/Z; abort with CTRL/C:
this is a message^Z
Send message? [confirm]
Router#


***
***
*** Message from tty50 to all terminals:
***
this is a message


Router#
```

Clearing a TCP/IP Connection Example

The following example clears a TCP connection using its tty line number. The **show tcp** EXEC command displays the line number (tty2) that is used in the **clear tcp** EXEC command.

```
Router# show tcp

    tty2, virtual tty from host router20.cisco.com
    Connection state is ESTAB, I/O status: 1, unread input bytes: 0
    Local host: 171.69.233.7, Local port: 23
    Foreign host: 171.69.61.75, Foreign port: 1058

    Enqueued packets for retransmit: 0, input: 0, saved: 0

    Event Timers (current time is 0x36144):
    Timer          Starts    Wakeups            Next
    Retrans             4          0             0x0
    TimeWait            0          0             0x0
    AckHold             7          4             0x0
    SendWnd             0          0             0x0
    KeepAlive           0          0             0x0
    GiveUp              0          0             0x0
    PmtuAger            0          0             0x0

    iss: 4151109680  snduna: 4151109752  sndnxt: 4151109752     sndwnd:  24576
    irs: 1249472001  rcvnxt: 1249472032  rcvwnd:       4258  delrcvwnd:     30

    SRTT: 710 ms, RTTO: 4442 ms, RTV: 1511 ms, KRTT: 0 ms
    minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 300 ms

Router# clear tcp line 2
    [confirm]
     [OK]
```

The following example clears a TCP connection by specifying its local router host name and port and its remote router host name and port. The **show tcp brief** EXEC command displays the local (Local Address) and remote (Foreign Address) host names and ports to use in the **clear tcp** EXEC command.

```
Router# show tcp brief
    TCB       Local Address           Foreign Address        (state)
    60A34E9C  router1.cisco.com.23      router20.cisco.1055  ESTAB

Router# clear tcp local router1 23 remote router20 1055
    [confirm]
     [OK]
```

The following example clears a TCP connection using its TCB address. The **show tcp brief** EXEC command displays the TCB address to use in the **clear tcp** EXEC command.

```
Router# show tcp brief
    TCB       Local Address           Foreign Address        (state)
    60B75E48  router1.cisco.com.23      router20.cisco.1054  ESTAB

Router# clear tcp tcb 60B75E48
    [confirm]
     [OK]
```

Configuring Banners Example

The following example shows how to use the **banner** global configuration commands to notify your users that the server will be reloaded with new software. The **no exec-banner** line configuration command is used to disable EXEC banners and message-of-the-day banners on the vty lines.

```
!
line vty 0 4
 no exec-banner
!
banner exec /
 This is Cisco Systems training group router.

 Unauthorized access prohibited.
 /
!
banner incoming /
 You are connected to a Hayes-compatible modem.

 Enter the appropriate AT commands.
 Remember to reset anything you have changed before disconnecting.
 /
!
banner motd /
 The router will go down at 6pm today for a software upgrade
 /
```

When someone connects to the router, the MOTD banner appears before the login prompt. After the user logs in to the router, the router will display the EXEC banner or incoming banner, depending on the type of connection. For a reverse Telnet login, the router will display the incoming banner. For all other connections, the router will display the EXEC banner.

Setting a SLIP-PPP Banner with Banner Tokens Example

The following example sets the SLIP-PPP banner using several tokens and the percent sign (%) as the delimiting character:

```
Router(config)# banner slip-ppp %
Enter TEXT message.  End with the character '%'.
Starting $(encap) connection from $(gate-ip) to $(peer-ip) using a maximum packet size of
$(mtu) bytes... %
```

When a user enters the **slip** command, that user will see the following banner. Notice that the **$(***token***)** syntax is replaced by the corresponding configuration variable.

```
Starting SLIP connection from 192.168.69.96 to 172.16.80.8 using a maximum packet size of
1500 bytes...
```

Configuring a Menu Example

The following example allows menu users to use Telnet to access one of three different machines. The user also can view the output of the **show user** EXEC command and exit the menu. One hidden menu item (configured as `menu new command here show version`) allows system administrators to display the current software version.

```
menu new title ^C


              Telnet Menu


^C
menu new prompt ^C

Please enter your selection: ^C
menu new text 1 telnet system1
menu new command 1 telnet system1
menu new options 1 pause
menu new text 2 telnet system2
menu new command 2 telnet system2
menu new options 2 pause
menu new text b telnet systemblue
menu new command b telnet systemblue
menu new options b pause
menu new text me show user
menu new command me show user
menu new options me pause
menu new command here show version
menu new text Exit Exit
menu new command Exit menu-exit
menu new clear-screen
menu new status-line
menu new default me
menu new line-mode
!
```

Using the Cisco Web Browser User Interface

The Cisco IOS software includes a Web browser user interface (UI) from which you can issue Cisco IOS commands. The Cisco IOS Web browser UI is accessed from the router home page, and can be customized for your business environment. For example, you can view pages in different languages and save them in Flash memory for easy retrieval.This chapter discusses the tasks associated with using and customizing the Cisco Web browser UI.

For a complete description of the Cisco Web browser UI configuration commands in this chapter, refer to the "Cisco IOS Web Browser User Interface Commands" chapter of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

Cisco Web Browser UI Task List

You can issue most Cisco IOS commands using a Web browser by connecting to the home page generated by the Cisco IOS software for your system. Most Cisco routers and access servers automatically generate a password protected home page when the HTTP server is enabled on the device. To access the home page, your computer must be on the same network as the router.

To use the Cisco Web browser UI, your computer must have a World Wide Web browser application. The Cisco Web browser UI works with most web browsers, including Internet Explorer and Netscape Navigator. Your Web browser must be able to read and submit forms.

To use the Cisco Web browser UI, perform the tasks in the following sections:

- Enabling the Cisco Web Browser UI (Required)
- Configuring Access to the Cisco Web Browser UI (Required)
- Accessing and Using the Cisco Web Browser UI (Required)
- Customizing the Cisco Web Browser UI (Optional)

Enabling the Cisco Web Browser UI

The Web browser UI is automatically enabled on the Cisco 1003, Cisco 1004, or Cisco 1005 router to allow you to use ClickStart to configure your router. For all other Cisco devices, you must enable the Cisco Web browser UI as described here.

To enable the Cisco Web browser UI, you must enable the HTTP server on your router. To enable the HTTP server, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip http server`	Enables the HTTP server (web server) on the system.

Configuring Access to the Cisco Web Browser UI

To control access to the Cisco Web browser UI, you can specify the authentication method for the HTTP server, apply an access list to the HTTP server, and assign a port number for the HTTP server, as described in the following sections.

Specifying the Method for User Authentication

To specify how HTTP server users are authenticated, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip http authentication {aaa | enable | local | tacacs}`	Specifies how the HTTP server users are authenticated.

The **ip http authentication** command specifies the authentication method to be used for login when a client connects to the HTTP server. Use of the **ip http authentication aaa** command option is recommended. The **enable**, **local**, and **tacacs** methods should be specified using the **aaa authentication login** command.

If you do not use this command, the default authentication method is used. The default method of authentication for the HTTP server is to use the configured "enable" password. The "enable" password is configured with the **enable password** global configuration command. If the enable password is used as the HTTP server login authentication method, the client connects to the HTTP server with a default privilege level of 15.

Note When the "enable" password is used as the HTTP server login authentication method, any username entered will be ignored; the server will only verify the "enable" password. This may make it easier for an attacker to access the router. Because a username and password pair is more secure than using only a password for authentication, using only "enable" password for authentication is strongly discouraged. Instead, use of the **local** or **tacacs** authentication options, configured as part of a global Authentication, Authorization, and Accounting (AAA) framework, is recommended.
To configure HTTP access as part of a AAA policy, use the **ip http authentication aaa** command option. The "local", "tacacs", or "enable" authentication methods should then be configured using the **aaa authentication login** command.

For information about adding users into the local username database, refer to the *Cisco IOS Security Configuration Guide*.

Example: Configuring the HTTP Server Authentication Method

The following example specifies that the method configured for AAA should be used for authentication for HTTP server users. The AAA login method is configured as the "local" username/password authentication method.

```
Router(config)# ip http authentication aaa
Router(config)# aaa authentication login default local
```

Applying an Access List to the HTTP Server

To control which hosts can access the HTTP server used by the Cisco Web browser UI, you can apply an access list to the HTTP server. To apply an access list to the HTTP server, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip http access-class {access-list-number | access-list-name}`	Applies an access list to the HTTP server used by the Cisco IOS ClickStart software or the Cisco Web browser user interface.

Example: Configuring an Access List for HTTP Server Access

In the following example the access list identified as "20" is defined and assigned to the HTTP server:

```
Router(config)# ip access-list standard 20
Router(config-std-nacl)# permit 209.165.202.0 0.0.0.255
Router(config-std-nacl)# permit 209.165.0.0 0.0.255.255
Router(config-std-nacl)# permit 209.0.0.0 0.255.255.255
! (Note: all other access implicitly denied)
Router(config-std-nacl)# exit
Router(config)# ip http access-class 20
```

Changing the HTTP Server Port Number

By default, the HTTP server uses port 80 on the router. To assign the Cisco Web browser UI to a different port, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip http port number`	Assigns a port number to be used by the Cisco Web browser interface.

Accessing and Using the Cisco Web Browser UI

This section describes the tasks used to access the Cisco Web browser UI and issue commands.

Accessing the Router Home Page

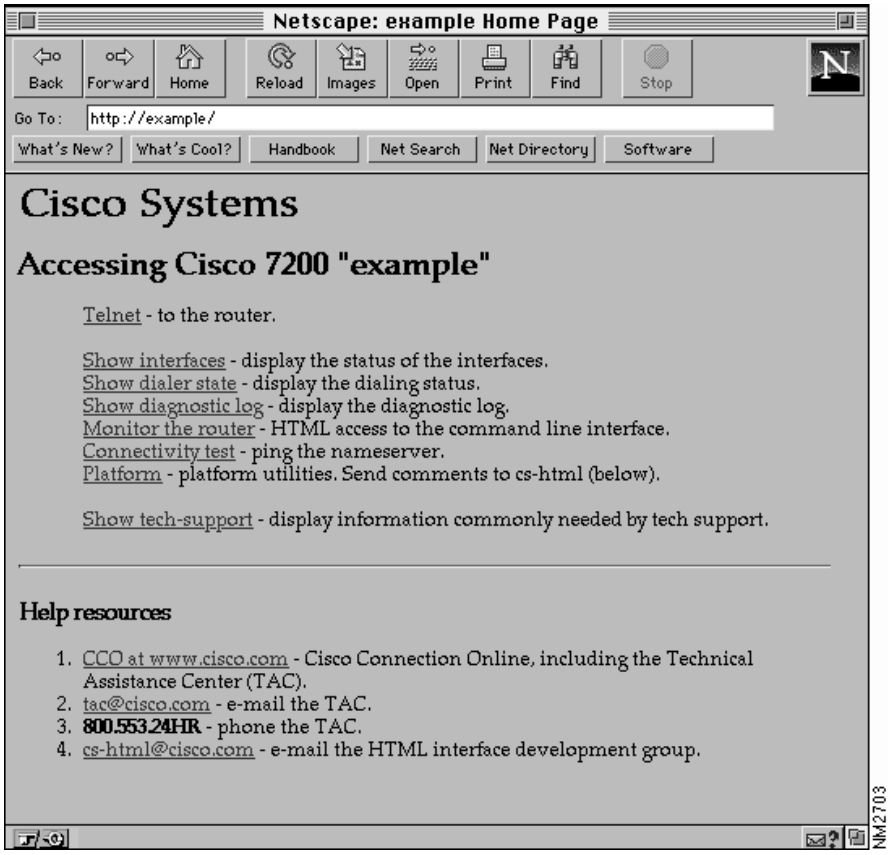
To access a router home page, perform the following steps:

Step 1 Enter **http://***router-name***/** in the URL field of your Web browser and press **Return** . (For example, to access a Cisco router named cacophony, type **http://cacophony/**.) The browser then prompts you for the password.

Step 2 Enter the password. The required password is dependent on the user authentication method configured for the HTTP server (using the **ip http authentication** global configuration command).

After entering the password, the browser will display the router home page. An example of a router home page is shown in shown in Figure 7.

Figure 7 Example of a Home Page for a Cisco 7200 Series Router



The default privilege level when accessing a router home page is privilege level 15 (global access). If privilege levels have been configured on the router and you have been assigned a privilege level other than 15, you must specify the privilege level to access the router home page.

When you specify a privilege level, the Cisco Web Browser UI will display and accept only those commands that have been defined for your user level. (For more information about privilege levels, see the "Configuring Passwords and Privileges" chapter in the Release 12.2 *Cisco IOS Security Configuration Guide*.)

To access a router Web page for a preassigned privilege level other than the default of 15, perform the following steps:

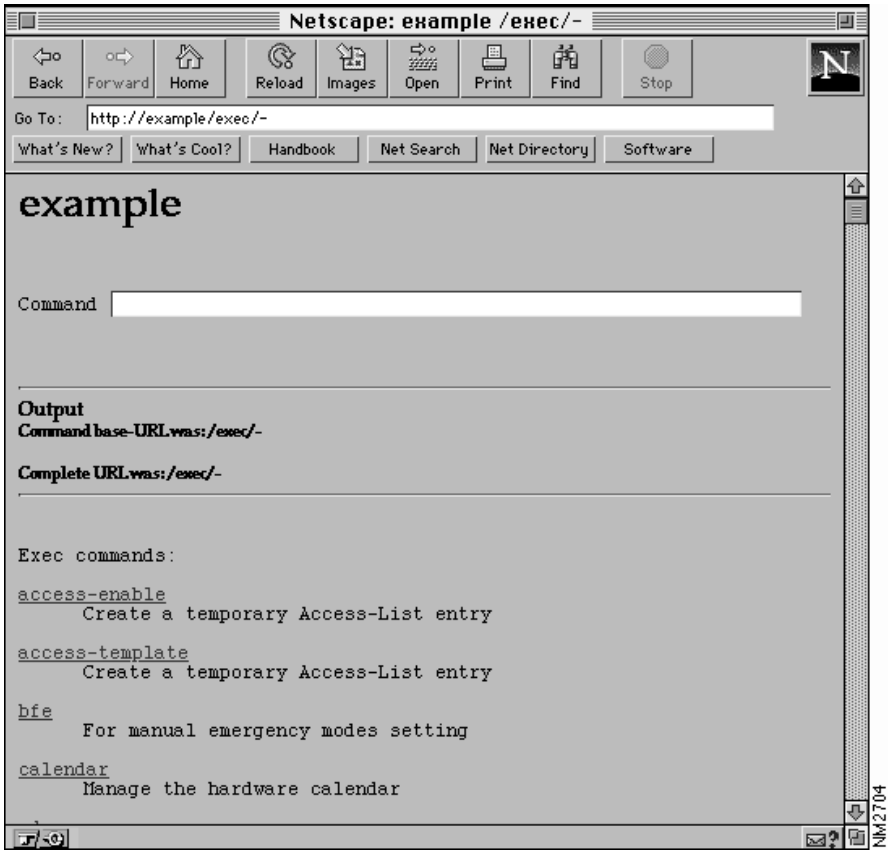
Step 1 Enter **http://***router-name***/level/***level***/exec** in the URL field of your Web browser and press **Return**. For example, to request access to EXEC mode at user privilege level of 12 on a Cisco router named cacophony, type **http://cacophony/level/12/exec**. The browser will then prompt you for your username and password.

Step 2 Enter your username and password and press **Return**. The required password is dependent on the user authentication method configured for the HTTP server. The Web browser will display a Web page specific to your user privilege level.

Issuing Commands Using the Cisco Web Browser UI

From the router home page, click the hypertext link titled **Monitor the Router**. This link takes you to a Web page that has a Command field. An example is shown in Figure 8. You can enter commands in the command field in the same way as you would enter commands using the Cisco IOS command-line interface. The page also displays a list of commands. You can execute these commands by clicking them, as if you were clicking hypertext links.

Figure 8 The Command Field Web Page for a Router Named example



Entering Commands Using Hypertext Links

To enter a command using hypertext links, scroll through the commands listed at the bottom of the screen and click the one you want to execute. If the link is a complete command, it is executed. If the command has more parameters, another list of command hypertext links is displayed. Scroll through this second list and click the one you want to execute.

If the command is a request for information, like a **show** EXEC command, the information is displayed in the Web browser window.

If the command requires a variable, a form in which you can enter the variable is displayed.

Entering Commands Using the Command Field

Entering the command in the command field is just like entering it at a terminal console. Enter the command using the syntax documented in the Cisco IOS command reference. If you are uncertain of the options available for a particular command, type a question mark (**?**).

For example, entering **show ?** in the command field displays the parameters for the **show** EXEC command. The Cisco Web browser UI displays the parameters as hypertext links. To select a parameter, you can either click on one of the links or you can enter the parameter in the command field.

Entering Commands Using the URL Window

You can issue a command using the URL window for the Web browser. To issue a command using the URL window, use the following syntax:

http://*router-name***/**[**level/***level***/**]*command-mode***/***command*

Table 13 lists the URL arguments you must use when requesting a web page.

Table 13 Web Browser URL Argument Descriptions

Argument	Description
router-name	Name of the router being configured.
level/*level*	(Optional) The privilege level you are requesting at which you are requesting access.
mode	The mode the command will be executed in, such as EXEC, configuration, or interface.
command	The command you want to execute. Replace spaces in the command syntax with forward slashes. If you do not specify a command in the URL, your browser will display a web page listing all of the commands available for the specified command mode.

For example, to execute a **show running-configuration** EXEC command on a router named example, you would enter the following in the URL window:

```
http://example/exec/show/running-configuration
```

After issuing this command, the Cisco Web browser UI will display the running configuration for the router.

The difference between entering a command in the Command field and entering a command in the URL window is that in the URL window, forward slashes should be used instead of spaces in the command syntax.

Customizing the Cisco Web Browser UI

You can customize the HTML pages used by the Cisco Web browser UI to display Cisco IOS command output and Cisco IOS platform-specific variables (for example, a router host name or router address). You can display this information using HTML formatted Server Side Includes (SSIs) that you insert into your custom HTML pages.

Understanding SSIs

SSIs are HTML formatted commands or variables that you insert into HTML pages when you customize Cisco IOS platform configuration pages for a Web browser. These SSI commands and SSI variables display Cisco IOS command output and Cisco IOS platform-specific variables.

Note The majority of the customization features in this section are for the ClickStart EZsetup feature for the Cisco 1000 series, Cisco 1003/1004 series, and Cisco 1005 series routers only.

The Cisco IOS software supports two HTML SSI commands defined for customizing HTML pages: the SSI EXEC command and the SSI ECHO command. The HTML format of the SSI EXEC command is **<!--#exec cmd="***xxx***"-->**, and the HTML format of the SSI ECHO command is **<!--#echo var="***yyy***"-->**. (See the section "Customizing HTML Pages Using SSIs" later in this chapter for a description of how to use these commands).

In addition to the two SSI commands, the Cisco IOS software supports several SSI variables defined for customizing HTML pages. SSI variables are used with the SSI ECHO command. One SSI variable is defined for all Cisco IOS platforms (SERVER_NAME), and other SSI variables are specifically defined for ISDN, Frame Relay, and asynchronous serial platforms. The format and a description of all the available SSI variables are provided in Table 14. (See the section "Customizing HTML Pages Using SSIs" later in this chapter for a description of how to use these SSI variables with the SSI ECHO command).

The SSI EXEC command is supported on all platforms. The SSI ECHO command, used with SSI variables, is supported on all platforms listed in Table 14.

Table 14 Description of SSI Variables

HTML Format of SSI Variable	Description of Variable Displayed on Browser Page	Cisco IOS Platforms This SSI Is Supported On
SERVER_NAME	Host name of the HTTP server.	All Cisco IOS platforms
EZSETUP_PASSWORD	Enable password (currently left blank).	Cisco 1000 series
EZSETUP_PASSWORD_VERIFY	Repeat of the enable password to verify accuracy (currently left blank).	Cisco 1000 series
EZSETUP_ETHERNET0_ADDRESS	IP address of the Ethernet interface 0.	Cisco 1000 series
EZSETUP_ETHERNET0_MASK	IP mask of the Ethernet interface 0.	Cisco 1000 series
EZSETUP_DNS_ADDRESS	Domain Name System (DNS) address used by the router.	Cisco 1000 series
EZSETUP_STANDARD_DEBUG_Y	Standard debug variable. Returns CHECKED if set to TRUE; otherwise, it is blank.	Cisco 1000 series
EZSETUP_STANDARD_DEBUG_N	Standard debug variable. Returns CHECKED if set to FALSE; otherwise, it is blank.	Cisco 1000 series
EZSETUP_ISDN_SWITCHTYPE	ISDN switch type.	Cisco 1003 and Cisco 1004
EZSETUP_ISDN_REMOTE_NAME	Name of remote ISDN system.	Cisco 1003 and Cisco 1004
EZSETUP_ISDN_REMOTE_NUMBER	Phone number of remote ISDN system.	Cisco 1003 and Cisco 1004
EZSETUP_ISDN_CHAP_PASSWORD	CHAP password of remote ISDN system.	Cisco 1003 and Cisco 1004
EZSETUP_ISDN_SPID1	ISDN SPID 1.	Cisco 1003 and Cisco 1004
EZSETUP_ISDN_SPID2	ISDN SPID 2.	Cisco 1003 and Cisco 1004
EZSETUP_ISDN_SPEED_56	Speed of ISDN interface. Returns CHECKED if set to 56K; otherwise, it is blank.	Cisco 1003 and Cisco 1004

Table 14 Description of SSI Variables (continued)

HTML Format of SSI Variable	Description of Variable Displayed on Browser Page	Cisco IOS Platforms This SSI Is Supported On
EZSETUP_ISDN_SPEED_64	Speed of ISDN interface. Returns CHECKED if set to 64K; otherwise, it is blank.	Cisco 1003 and Cisco 1004
EZSETUP_FR_ADDRESS	Frame Relay IP address.	Cisco 1005
EZSETUP_FR_MASK	Frame Relay IP mask.	Cisco 1005
EZSETUP_FR_DLCI	Frame Relay DLCI.	Cisco 1005
EZSETUP_ASYNC_REMOTE_NAME	Name of remote system.	Cisco 1005
EZSETUP_ASYNC_REMOTE_NUMBER	Phone number of remote system.	Cisco 1005
EZSETUP_ASYNC_CHAP_PASSWORD	CHAP password for remote system.	Cisco 1005
EZSETUP_ASYNC_LINE_PASSWORD	Async line password.	Cisco 1005
EZSETUP_ASYNC_MODEM_SPEED	Speed of async modem (either 14.4K or 28.8K).	Cisco 1005
EZSETUP_ASYNC_MODEM_SPEED_144K	Returns CHECKED if async modem speed is 14.4K; otherwise it is blank.	Cisco 1005
EZSETUP_ASYNC_MODEM_SPEED_288K	Returns CHECKED if async modem speed is 28.8K; otherwise it is blank.	Cisco 1005

Once you have designed a set of HTML pages that include SSIs, you can copy these pages to a Cisco IOS platform's Flash memory. When you retrieve these pages from Flash memory and display them using a Web browser, any SSI command that was designed into these pages will display either Cisco IOS command output or a current variable or identifier defined in Table 14. For example, the SSI ECHO command with the variable SERVER_NAME will display the current host name of the HTTP server you are using, and the SSI ECHO command with the variable EZSETUP_ISDN_SWITCHTYPE will display the current ISDN switch type you are using.

Using SSIs, you can customize set of HTML pages to appear in languages other than English and copy these pages to Flash memory on multiple Cisco IOS platforms. When you retrieve these pages from the Flash memory of a Cisco IOS platform, current variables and identifiers associated with the platform you are currently using are displayed. SSIs save you from needing to duplicate these international pages (considered relatively large images that contain 8-bit or multibyte characters) and store them in the source code for each platform you are using.

Customizing HTML Pages Using SSIs

When you are customizing an HTML page for a Web browser, type **<!--#exec cmd="***xxx***"-->** in your HTML file where you want Cisco IOS command output to appear on the browser page. Replace the *xxx variable* with any Cisco IOS EXEC mode command.

When you are customizing an HTML page for a Web browser, type **<!--#echo var="***yyy***"-->** in your HTML file where you want a value or identifier associated with a particular Cisco IOS platform (for example, an ISDN or Frame Relay platform) to appear on the browser page. Replace the *yyy* variable with an SSI variable described in Table 14.

Copying HTML Pages to Flash Memory

Once you have customized HTML pages using SSIs, copy your HTML pages to a Cisco IOS platform's Flash memory. To do this, save your pages using a filename appended with ".shtml" (for example, *filename*.shtml) and copy your file to Flash memory using a **copy** EXEC command (for example, the **copy tftp flash** command). (Refer to the Cisco IOS command references for a **copy** command compatible with your platform.)

Displaying HTML Files Containing SSIs

Once the Cisco Web browser UI is enabled, you can retrieve your HTML page from Flash memory and display it on the Cisco Web browser by typing **http://***router***/flash/***filename* in the URL window. Replace *router* with the host name or IP address of the current Cisco IOS platform you are using, and replace *filename* with the name of the file you created with ".shtml" appended, for example, http://myrouter/flash/ssi_file.shtml.

Cisco Web Browser UI Customization Examples

This section provides the following examples:

- Using the SSI EXEC Command Example
- Using the SSI ECHO Command Example

Using the SSI EXEC Command Example

The following example shows how the HTML SSI EXEC command can be used to execute a command. In this example, the Cisco IOS **show users** EXEC command is executed.

The contents of the HTML file in Flash memory are as follows:

```
<HTML>
<HEAD>
<TITLE> SSI EXEC Command Example</TITLE>
</HEAD>
<BODY>
This is an example of the SSI EXEC command
<HR>
<PRE>
<!--#exec cmd="show users"-->
</PRE>
<BR>
</BODY>
</HTML>
```

The contents that the Web browser receives when the HTML file is retrieved from Flash memory are as follows:

```
<HTML>
<HEAD>
<TITLE> SSI EXEC Command Example</TITLE>
</HEAD>
<BODY>
This is an example of the SSI EXEC command
<HR>
USERS:<BR>
<PRE>

Line    User  Host(s) Idle  Location
0 con 0       idle      12
2 vty 0       idle       0  router.cisco.com

</PRE>
<BR>
</BODY>
</HTML>
```

The Web browser shows the following text:

```
This is an example of the SSI EXEC command
----------------------------------------
USERS:
Line    User  Host(s) Idle  Location
0 con 0       idle      12
2 vty 0       idle       0  router.cisco.com
```

Using the SSI ECHO Command Example

The following is an example of the HTML SSI ECHO command used with the SSI variable *SERVER_NAME* (see Table 5) to display the Cisco IOS platform host name "rain."

The contents of the HTML file in Flash memory is as follows:

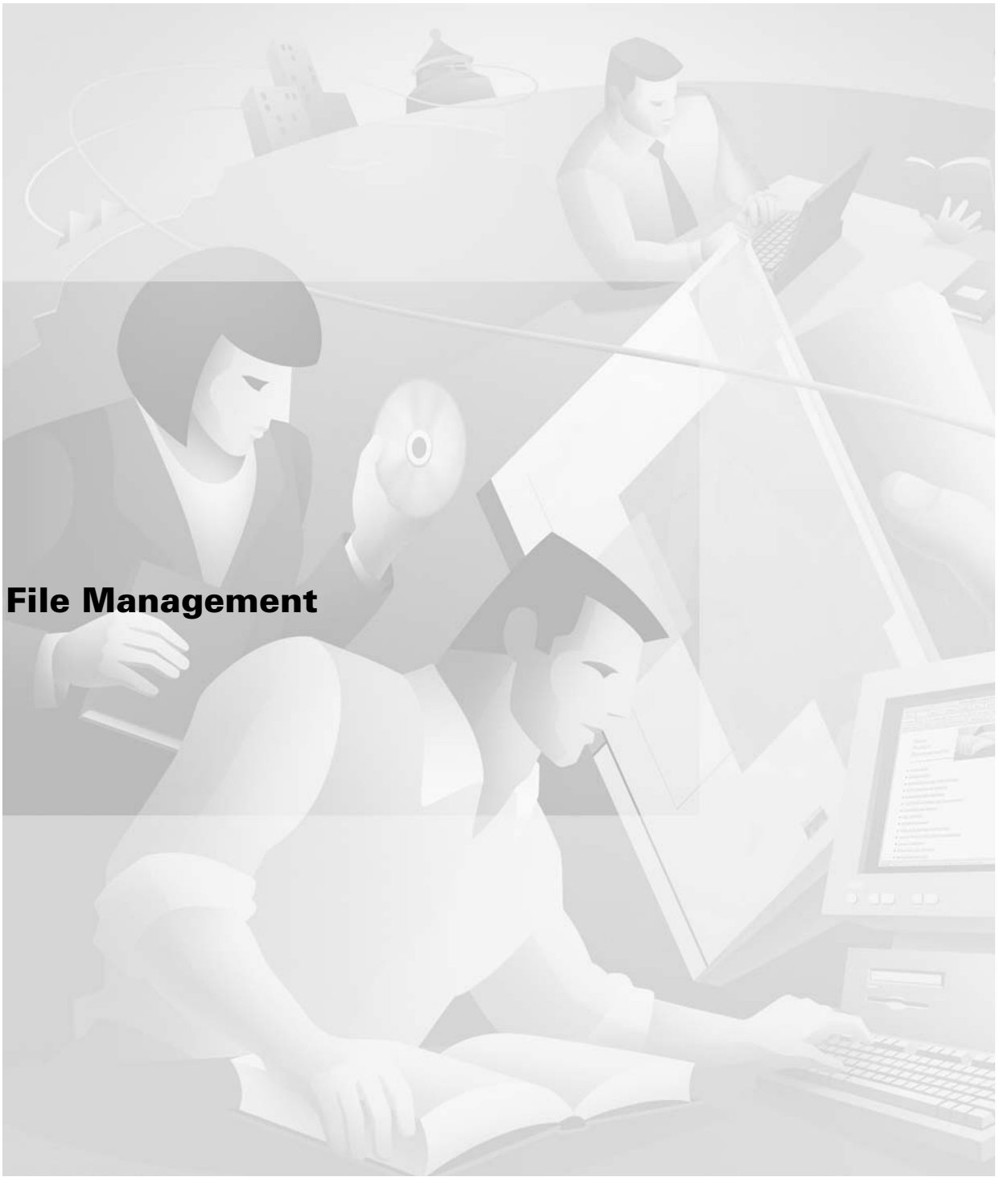
```
<HTML>
<HEAD>
<TITLE>SSI Echo Command Example</TITLE>
</HEAD>
<BODY>
This is an example of the SSI echo command
<HR>
The name of this server is:<BR>
<!--#echo var="SERVER_NAME"-->
<BR>
</BODY>
</HTML>
```

The contents that the Web browser receives when the HTML file is retrieved from Flash memory are as follows:

```
<HTML>
<HEAD>
<TITLE>SSI Echo Command Example</TITLE>
</HEAD>
<BODY>
This is an example of the SSI echo command
<HR>
The name of this server is:<BR>
rain
<BR>
</BODY>
</HTML>
```

The Web Browser shows the following text:

```
This is an example of the SSI echo command
----------------------------------------
The name of this server is:
rain
```



File Management

Using the Cisco IOS File System

This chapter describes the Cisco IOS File System (IFS) feature, which provides a single interface to all the file systems available on your routing device, including the following:

- Flash memory file systems
- Network file systems (TFTP, rcp, and FTP)
- Any other endpoint for reading or writing data (such as NVRAM, the running configuration, ROM, raw system memory, system bundled microcode, Xmodem, Flash load helper log, modems, and BRI multiplexing device [mux] interfaces)

For a complete description of the IFS commands in this chapter, refer to the "Cisco IOS File System Commands" chapter in the "File Management Commands" part of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

IFS Use and Management Task List

This chapter describes the tasks you can perform to manage files using the Cisco IFS. Information about the IFS and its optional file management tasks are described in the following sections:

- Understanding IFS
- Copying Files Using URLs
- Using URLs in Commands
- Managing File Systems
- Flash Memory File System Types
- Remote File System Management
- NVRAM File System Management
- System File System Management

Understanding IFS

IFS capabilities and benefits are described in the following sections:

- Displaying and Classifying Files
- Platform-Independent Commands
- Minimal Prompting for Commands
- Creating and Navigating Directories

Displaying and Classifying Files

With IFS, all files can be viewed and classified (image, text file, and so on), including files on remote servers. For example, you may want to determine the size and type of an image on a remote server before you copy it to ensure that it is a valid image. You can also display a configuration file on a remote server to verify that it is the correct configuration file before you load the file on the router.

Platform-Independent Commands

With IFS, the file system user interface is no longer platform-specific. Commands have the same syntax, regardless of which platform is used. Thus, you can use the same commands for all of your routers.

However, not all commands are supported on all platforms and file systems. Because different types of file systems support different operations, certain commands are not available for all file systems. Platforms will support commands for the file systems they use.

Minimal Prompting for Commands

IFS minimizes the required prompting for many commands, such as the **copy** EXEC command. You can enter all of the required information in the command line, rather than needing to provide information when the system prompts you for it. For example, if you want to copy a file to an FTP server, on a single line you can specify the specific location on the router of the source file, the specific location of the destination file on the FTP server, and the username and password to use when connecting to the FTP server. However, to have the router prompt you for the needed information, you can still enter the minimal form of the command.

Depending on the current configuration of the **file prompt** global configuration command and the type of command you entered, the router may prompt you for confirmation, even if you have provided all the information in the command. In these cases, the default value will be the value entered in the command. Press Return to confirm the values.

Creating and Navigating Directories

With IFS, you can navigate to different directories and list the files in a directory. On newer platforms, you can create subdirectories in Flash memory or on a disk.

Copying Files Using URLs

The new file system interface uses Uniform Resource Locators (URLs) to specify the location of a file. URLs are commonly used to specify files or locations on the World Wide Web. However, on Cisco routers, they can now be used to specify the location of files on the router or remote file servers.

On Cisco routers, use URLs in commands to specify the location of the file or directory. For example, if you want to copy a file from one location to another, use the **copy** *source-url destination-url* EXEC command.

The format of URLs used by the routers can vary from the format you may be used to using. There are also a variety of formats that can be used, based on the location of the file.

Information for copying files using URLs is included in the following sections:

- Specifying Files on a Network Server
- Specifying Local Files
- Using URL Prefixes

Specifying Files on a Network Server

To specify a file on a network server, use one of the following forms:

- **ftp:**[[//[*username*[:*password*]@]*location*]/*directory*]/*filename*
- **rcp:**[[//[*username*@]*location*]/*directory*]/*filename*
- **tftp:**[[//*location*]/*directory*]/*filename*

The *location* can be an IP address or a host name. The *username* variable, if specified, overrides the username specified by the **ip rcmd remote-username** or **ip ftp username** global configuration command. The *password* overrides the password specified by the **ip ftp password** global configuration command.

The file path (directory and filename) is specified relative to the directory used for file transfers. For example, on UNIX file servers, TFTP pathnames start in the /tftpboot directory, and rcp and FTP paths start in the home directory associated with the username.

The following example specifies the file named c7200-j-mz.112-current on the TFTP server named myserver.cisco.com. The file is located in the directory named /tftpboot/master.

```
tftp://myserver.cisco.com/master/c7200-j-mz.112-current
```

The following example specifies the file named mill-config on the server named enterprise.cisco.com. The router uses the username liberty and the password secret to access this server via FTP.

```
ftp://liberty:secret@enterprise.cisco.com/mill-config
```

Specifying Local Files

Use the *prefix***:**[*directory*/]*filename* syntax to specify a file located on the router. You can use this form to specify a file in Flash memory or NVRAM.

For example, `nvram:startup-config` specifies the startup configuration in NVRAM, and `flash:configs/backup-config` specifies the file named backup-config in the configs directory of Flash memory.

When referring to a file system instead of a file, use the *prefix***:** form. This form specifies the file system itself, rather than a file in the file system. Use this form to issue commands on file systems themselves, such as commands to list the files in a file system or to format the file system.

For example, slot0: can indicate the first Personal Computer Memory Card Industry Association (PCMCIA) Flash memory card in slot 0.

Using URL Prefixes

The URL prefix specifies the file system. The list of available file systems differs by platform and operation. Refer to your product documentation or use the **show file systems** EXEC command to determine which prefixes are available on your platform. File system prefixes are listed in Table 15.

Table 15 File System Prefixes

Prefix	File System
bootflash:	Boot Flash memory.
disk0:	Rotating media.
flash:	Flash memory. This prefix is available on all platforms. For platforms that do not have a device named flash:, the prefix flash: is aliased to slot0:. Therefore, you can use the prefix flash: to refer to the main Flash memory storage area on all platforms.
flh:	Flash load helper log files.
ftp:	FTP network server.
null:	Null destination for copies. You can copy a remote file to null to determine its size.
nvram:	NVRAM.
rcp:	Remote copy protocol network server.
slavebootflash:	Internal Flash memory on a slave RSP card of a router configured for high system availability (HSA).
slavenvram:	NVRAM on a slave Route/Switch Processor (RSP) card of a router configured for HSA.
slaveslot0:	First PCMCIA card on a slave RSP card of a router configured for HSA.
slaveslot1:	Second PCMCIA card on a slave RSP card of a router configured for HSA.
slot0:	First PCMCIA Flash memory card.
slot1:	Second PCMCIA Flash memory card.
system:	Contains the system memory, including the running configuration.
tftp:	TFTP network server.

Table 15 File System Prefixes (continued)

Prefix	File System
xmodem:	Obtain the file from a network machine using the Xmodem protocol.
ymodem:	Obtain the file from a network machine using the Ymodem protocol.

Note Maintenance Operation Protocol (MOP) servers are no longer supported as file systems.

In all commands, the colon is required after the file system name. However, commands that did not require the colon previously will continue to be supported, although they will not be available in the context-sensitive help.

URL Prefix for Partitioned Devices

For partitioned devices, the URL prefix includes the partition number. The syntax is *device*:*partition-number*: for the prefix on a partitioned device.

For example, flash:2: refers to the second partition in Flash memory.

URL Component Lengths

Table 16 lists the maximum lengths in characters of the different URL components.

Table 16 URL Component Lengths

Component	Length (Number of Characters)
Prefix	31
Username	15
Password	15
Hostname	31
Directory	63
Filename	63

Using URLs in Commands

Depending on which command you are using, different file systems are available. Some file systems can only serve as a source for files, not a destination. For example, you cannot copy to another machine using Xmodem. Other operations, such as **format** and **erase**, are only supported by certain file systems on certain platforms.

The following sections describe the use of for using URLs in commands:

- Determining File Systems Supporting a Command
- Using the Default File System
- Using Tab Completion
- Listing Files in a File System

Determining File Systems Supporting a Command

Use the context-sensitive help to determine which file systems can be used for a particular command. In the following example, the context-sensitive help displays which file systems can be used as sources for the **copy** EXEC command. The output will vary based on the platform.

```
Router# copy ?
  /erase      Erase destination file system.
  bootflash:  Copy from bootflash: file system
  flash:      Copy from flash: file system
  ftp:        Copy from ftp: file system
  null:       Copy from null: file system
  nvram:      Copy from nvram: file system
  rcp:        Copy from rcp: file system
  system:     Copy from system: file system
  tftp:       Copy from tftp: file system
```

Using the Default File System

For most commands, if no file system is specified, the file is assumed to be in the default directory, as specified by the **cd** command.

```
Router# pwd
slot0:
Router# dir
Directory of slot0:/

  1  -rw-     4720148   Aug 29 1997 17:49:36  hampton/nitro/c7200-j-mz
  2  -rw-     4767328   Oct 01 1997 18:42:53  c7200-js-mz
  5  -rw-         639   Oct 02 1997 12:09:32  foo
  7  -rw-         639   Oct 02 1997 12:37:13  the_time

20578304 bytes total (3104544 bytes free)
Router# cd nvram:
Router# dir
Directory of nvram:/

  1  -rw-        2725              <no date>  startup-config
  2  ----           0              <no date>  private-config
  3  -rw-        2725              <no date>  underlying-config

129016 bytes total (126291 bytes free)
```

Using Tab Completion

You can use tab completion to reduce the number of characters you need to type for a command. Type the first few characters of the filename, and press the Tab key. If the characters are unique to a filename, the router will complete the filename for you. Continue entering the command as normal and press Return to execute the command.

In the following example, the router completes the filename startup-config because it is the only file in the nvram: file system that starts with "s":

```
Router# show file info nvram:s<tab>
Router# show file info nvram:startup-config<Enter>
```

If you use tab completion without specifying any characters, the router uses the first file in the file system.

```
Router# show file info nvram:<tab>
Router# show file info nvram:private-config<Enter>
```

Listing Files in a File System

For many commands, you can get a listing of the files in a file system on the router by using the context-sensitive help. In the following example, the router lists the files in NVRAM:

```
Router# show file info nvram:?
nvram:private-config  nvram:startup-config  nvram:underlying-config
```

Managing File Systems

To manage file systems, perform the tasks described in the following sections.

- Listing Available File Systems
- Setting the Default File System
- Displaying the Current Default File System
- Displaying Information About Files on a File System
- Displaying a File

Listing Available File Systems

Not all file systems are supported on every platform. To list the file systems available on your platform, use the following EXEC mode command:

Command	Purpose
`Router> show file systems`	Lists the file systems available on your platform. This command also displays information about each file system.

Setting the Default File System

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

To set a default file system, use the following command in EXEC mode:

Command	Purpose
Router> **cd** *filesystem*:	Sets a default Flash memory device.

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

```
cd slot0:
```

Displaying the Current Default File System

To display the current default file system, as specified by the **cd** EXEC command, use the following command in EXEC mode:

Command	Purpose
Router> **pwd**	Displays the current file system.

The following example shows that the default file system is slot 0:

```
Router> pwd
slot0:
```

The following example uses the **cd** command to change the default file system to system and then uses the **pwd** command to verify that the default file system was changed:

```
Router> cd system:
Router> pwd
system:
```

Displaying Information About Files on a File System

You can display a list of the contents of a file system before manipulating its contents. For example, before copying a new configuration file to Flash memory, you may want to verify that the file system does not already contain a configuration file with the same name. Similarly, before copying a Flash configuration file to another location, you may want to verify its filename for use in another command.

To display information about files on a file system, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# **dir** [**/all**] [*filesystem***:**][*filename*]	Displays a list of files on a file system.
Router# **show file systems**	Displays detailed information about each of the files on a file system.
Router# **show file information** *file-url*	Displays information about a specific file.
Router# **show file descriptors**	Displays a list of open file descriptors.

The following example compares the different commands used to display information about files for the PCMCIA card in the first slot. Notice that deleted files appear in the **dir /all** command output but not in the **dir** command output.

```
Router# dir slot0:
Directory of slot0:/

  1  -rw-     4720148   Aug 29 1997 17:49:36  hampton/nitro/c7200-j-mz
  2  -rw-     4767328   Oct 01 1997 18:42:53  c7200-js-mz
  5  -rw-         639   Oct 02 1997 12:09:32  foo
  7  -rw-         639   Oct 02 1997 12:37:13  the_time

20578304 bytes total (3104544 bytes free)

Router# dir /all slot0:
Directory of slot0:/

  1  -rw-     4720148   Aug 29 1997 17:49:36  hampton/nitro/c7200-j-mz
  2  -rw-     4767328   Oct 01 1997 18:42:53  c7200-js-mz
  3  -rw-     7982828   Oct 01 1997 18:48:14  [rsp-jsv-mz]
  4  -rw-         639   Oct 02 1997 12:09:17  [the_time]
  5  -rw-         639   Oct 02 1997 12:09:32  foo
  6  -rw-         639   Oct 02 1997 12:37:01  [the_time]
  7  -rw-         639   Oct 02 1997 12:37:13  the_time

20578304 bytes total (3104544 bytes free)

Router# show slot0:
-#- ED --type-- --crc--- -seek-- nlen -length- -----date/time------ name
1   .. unknown  317FBA1B  4A0694   24  4720148 Aug 29 1997 17:49:36 hampton/nitz
2   .. unknown  9237F3FF  92C574   11  4767328 Oct 01 1997 18:42:53 c7200-js-mz
3   .D unknown  71AB01F1 10C94E0   10  7982828 Oct 01 1997 18:48:14 rsp-jsv-mz
4   .D unknown  96DACD45 10C97E0    8      639 Oct 02 1997 12:09:17 the_time
5   .. unknown  96DACD45 10C9AE0    3      639 Oct 02 1997 12:09:32 foo
6   .D unknown  96DACD45 10C9DE0    8      639 Oct 02 1997 12:37:01 the_time
7   .. unknown  96DACD45 10CA0E0    8      639 Oct 02 1997 12:37:13 the_time

3104544 bytes available (17473760 bytes used)
```

Displaying a File

To display the contents of any readable file, including a file on a remote file system, use the following command in EXEC mode:

Command	Purpose
`Router# more [/ascii | /binary | /ebcdic] file-url`	Displays the specified file.

The following example displays the contents of a configuration file on a TFTP server:

```
Router# more tftp://serverA/hampton/savedconfig

!
! Saved configuration on server
!
version 11.3
service timestamps log datetime localtime
service linenumber
service udp-small-servers
service pt-vty-logging
!
end
```

Flash Memory File System Types

Cisco platforms use one of the following three different Flash memory file system types:

- Class A Flash File Systems
- Class B Flash File Systems
- Class C Flash File Systems

The methods used for erasing, deleting, and recovering files depend on the class of the Flash file system. Some commands are supported on only one or two file system types. The command reference documentation notes commands that are not supported on all file system types.

See Table 17 to determine which Flash memory file system type your platform uses.

Table 17 Flash Memory File System Types

Type	Platforms
Class A	Cisco 7000 series (including the Cisco 7500 series), Cisco 12000 Gigabit Switch Router (GSR), LS1010
Class B	Cisco 1003, Cisco 1004, Cisco 1005, Cisco 2500 series, Cisco 3600 series, Cisco 4000 series, Cisco AS5200
Class C	Cisco MC3810, disk0 of SC3640

Class A Flash File Systems

On Class A Flash file systems, you can delete individual files using the **delete** EXEC command and later recover these files with the **undelete** EXEC command. The **delete** command marks the files as "deleted," but the files still take up space in Flash memory. To permanently delete the files, use the **squeeze** EXEC command. The **squeeze** command removes all of the files marked "deleted" from the specified Flash memory device. These files can no longer be recovered. To erase all of the files on a Flash device, use the **format** EXEC command.

Deleting Files on a Flash Memory Device

When you no longer need a file on a Flash memory device, you can delete it. When you delete a file, the router simply marks the file as deleted, but it does not erase the file. This feature allows you to recover a deleted file, as discussed in the following section. You may want to recover a "deleted" image or configuration file if the new image or configuration file becomes corrupted.

To delete a file from a specified Flash memory device, use the following EXEC mode command:

Command	Purpose
Router# **delete** [*device*:]*filename*	Deletes a file from a Flash memory device.

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

If you attempt to delete the file specified by the CONFIG_FILE or BOOTLDR environment variable, the system prompts you to confirm the deletion. Also, if you attempt to delete the last valid system image specified in the BOOT environment variable, the system prompts you to confirm the deletion.

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

```
delete slot0:myconfig
```

Recovering Deleted Files on a Flash Memory Device

You can undelete a deleted file. For example, you may want to revert to a previous configuration file because the current one is corrupt.

To undelete a deleted file on a Flash memory device, use the following commands in EXEC mode:

	Command	Purpose
Step 1	Router# **dir /all** [*filesystem*:]	Determines the index of the deleted file.
Step 2	Router# **undelete** *index* [*filesystem*:]	Restores a deleted file on a Flash memory device.

You must undelete a file by its index because you can have multiple deleted files with the same name. For example, the "deleted" list could contain multiple configuration files with the name router-config. You undelete by index to indicate which of the many router-config files from the list to undelete. Use the **dir** command with the **/all** option to learn the index number of the file you want to undelete.

You cannot undelete a file if a valid file with the same name exists. Instead, first delete the existing file and then undelete the file you want. For example, if you had a file with the name router-config and you wanted to use a file with the same name that you had previously deleted, you cannot simply undelete the

previous version by index. You must first delete the existing router-config file and then undelete the previous router-config file by index. You can undelete a file as long as the file has not been permanently erased with the **squeeze** EXEC command. You can delete and undelete a file up to 15 times.

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

```
undelete 1 slot0:
```

Permanently Deleting Files on a Flash Memory Device

When a Flash memory device is full, you may need to rearrange the files so that the space used by the deleted files can be reclaimed. To determine whether a Flash memory device is full, use the **dir** EXEC command.

To permanently delete files on a Flash memory device, use the following command in privileged EXEC mode:

Command	Purpose
Router# **squeeze** *filesystem*:	Permanently deletes all files marked "deleted" on a Flash memory device.

On Cisco 2600 and 3600 series routers,the entire flash file system needs to be erased once before the **squeeze** command can be used. After being erased once, the squeeze command should operate properly on the flash file system for the rest of the flash file system's history.

To erase an entire flash file system on a Cisco 2600 or 3600 series router, perform the following steps:

Command	Purpose
Router# **no partition** *flash-filesystem:*	Removes all partitions on the specified flash file system. **Note** The reason for removing partitions is to ensure that the entire flash file system is erased. The **squeeze** command can be used in a flash file system with partitions after the flash file system is erased once.
Router# **erase** *filesystem*:	Erases all of the file on the specified flash file system.

When you issue the **squeeze** command, the router copies all valid files to the beginning of Flash memory and erases all files marked "deleted." At this point, you cannot recover deleted files, and you can now write to the reclaimed Flash memory space.

Note The squeeze operation can take as long as several minutes because it can involve erasing and rewriting almost an entire Flash memory space.

Verifying Flash

To recompute and verify the checksum of a file in Flash memory on a Class A Flash file system, use the **verify** EXEC command.

Deleting and Recovering a Class A Flash File System Example

In the following example, the image named c7200-js-mz is deleted and undeleted. Note that the deleted file does not appear in the output for the first **dir** EXEC command, but it appears in the output for the **dir /all** EXEC command.

```
Router# delete slot1:
Delete filename []? c7200-js-mz
Delete slot1:c7200-js-mz? [confirm]
Router# dir slot1:
Directory of slot1:/

No such file

20578304 bytes total (15754684 bytes free)
Router# dir /all slot1:
Directory of slot1:/

  1  -rw-     4823492   Dec 17 1997 13:21:53  [c7200-js-mz]

20578304 bytes total (15754684 bytes free)
Router# undelete 1 slot1:
Router# dir slot1:
Directory of slot1:/

  1  -rw-     4823492   Dec 17 1997 13:21:53 c7200-js-mz

20578304 bytes total (15754684 bytes free)
```

In the following example, the image is deleted. In order to reclaim the space taken up by the deleted file, the **squeeze** EXEC command is issued.

```
Router# delete slot1:c7200-js-mz
Delete filename [c7200-js-mz]?
Delete slot1:c7200-js-mz? [confirm]
Router# squeeze slot1:
All deleted files will be removed. Continue? [confirm]
Squeeze operation may take a while. Continue? [confirm]
Erasing squeeze log
Squeeze of slot1: complete
Router# dir /all slot1:
Directory of slot1:/

No such file

20578304 bytes total (20578304 bytes free)
```

Class B Flash File Systems

On Class B Flash file systems, you can delete individual files with the **delete** EXEC command. The **delete** command marks the file as "deleted." The file is still present in Flash memory and takes up space. To recover the file, use the **undelete** EXEC command. To reclaim any space in Flash memory, you must erase the entire Flash file system with the **erase** EXEC command.

Deleting Files on a Flash Memory Device

When you no longer need a file on a Flash memory device, you can delete it. When you delete a file, the router simply marks the file as deleted, but it does not erase the file. This feature allows you to recover a deleted file, as discussed in the following section. You may want to recover a "deleted" image or configuration file if the new image or configuration file becomes corrupted.

To delete a file from a specified Flash memory device, use the following EXEC mode command:

Command	Purpose
Router# **delete** [*device*:]*filename*	Deletes a file from a Flash memory device.

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

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

```
delete slot0:myconfig
```

Recovering Deleted Files on a Flash Memory Device

You can undelete a deleted file. For example, you may want to revert to a previous configuration file because the current one is corrupt.

To undelete a deleted file on a Flash memory device, use the following EXEC mode commands:

	Command	Purpose
Step 1	Router# **dir /all** [*filesystem*:]	Determines the index of the deleted file.
Step 2	Router# **undelete** *index* [*filesystem*:]	Undeletes a deleted file on a Flash memory device.

You must undelete a file by its index because you can have multiple deleted files with the same name. For example, the "deleted" list could contain multiple configuration files with the name router-config. You undelete by index to indicate which of the many router-config files from the list to undelete. Use the **dir** command with the **/all** option to learn the index number of the file you want to undelete.

You cannot undelete a file if a valid (undeleted) one with the same name exists. Instead, first delete the existing file and then undelete the file you want. For example, if you had an undeleted version of the router-config file and you wanted to use a previous, deleted version instead, you cannot simply undelete the previous version by index. You must first delete the existing router-config file and then undelete the previous router-config file by index. You can undelete a file as long as the file system has not been permanently erased with the **erase** EXEC command. You can delete and undelete a file up to 15 times.

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

```
undelete 1 slot0:
```

Erasing Flash Memory

In order to reclaim any space taken up by files in Flash memory, you must erase the entire file system using the **erase flash:** or **erase bootflash:** EXEC command. These commands reclaim all of the space in Flash memory, erasing all files, deleted or not, in the process. Once erased, these files cannot be recovered. Before erasing Flash memory, save any files you want to keep in another location (an FTP server, for example). Copy the files back to Flash memory after you have erased the device.

To erase a Flash memory device, use the following command in EXEC mode:

Command	Purpose
Router# **erase** *filesystem*:	Erases the Flash file system.

Erasing a File System Example

The following example erases all files in the second partition in Flash memory:

```
Router# erase flash:2

System flash directory, partition 2:
File  Length   Name/status
  1   1711088  dirt/gate/c1600-i-mz
[1711152 bytes used, 15066064 available, 16777216 total]

Erase flash device, partition 2? [confirm]
Are you sure? [yes/no]: yes
Erasing device... eeee
```

Formatting Flash

To format a Class C Flash file system, use the following command in EXEC mode:

Command	Purpose
`Router# format filesystem`	Formats a Flash file system.

If you format a Flash device, all of the files are erased and cannot be recovered.

Creating and Removing Directories

On Class C Flash file systems, you can create a new directory with the **mkdir** EXEC command. To remove a directory from a Flash file system, use the **rmdir** EXEC command.

On Class C Flash file systems, you can rename a file using the **rename** EXEC command.

Checking Flash File Systems

On Class C Flash file systems, you can check a file system for damage and repair any problems using the **fsck** EXEC command.

Remote File System Management

On remote file systems (file systems on FTP, rcp, or TFTP servers) you can perform the following tasks:

- View the contents of a file with the **more** EXEC command.
- Copy files to or from the router using the **copy** EXEC command.
- Display information about a file using the **show file information** EXEC command.

Note You cannot delete files on remote systems.

NVRAM File System Management

On most platforms, NVRAM contains the startup configuration. On Class A Flash file system platforms, the CONFIG_FILE environment variable specifies the location of the startup configuration. However, the file URL nvram:startup-config always specifies the startup configuration, regardless of the CONFIG_FILE environment variable.

You can display the startup-config (with the **more nvram:startup-config** EXEC command), replace the startup config with a new configuration file (with the **copy** *source-url* **nvram:startup-config** EXEC command), save the startup configuration to another location (with the **copy nvram:startup-config** *destination-url* EXEC command), and erase the contents of NVRAM (with the **erase nvram:** EXEC command). The **erase nvram:** command also deletes the startup configuration if another location is specified by the CONFIG_FILE variable.

The following example displays the startup configuration:

```
nnm3640-2# more nvram:startup-config
```

```
Using 2279 out of 129016 bytes
!
! Last configuration change at 10:57:25 PST Wed Apr 22 1998
! NVRAM config last updated at 10:57:27 PST Wed Apr 22 1998
!
version 11.3
service timestamps log datetime localtime
service linenumber
service udp-small-servers
service pt-vty-logging
...
end
```

The following example displays the contents of the NVRAM file system on a Class A Flash file system platform. The file named startup-config is the current startup configuration file, in physical NVRAM or in Flash memory. If the file is located in a Flash memory file system, this entry is a symbolic link to the actual file. The file named underlying-config is always the NVRAM version of the configuration.

```
Router# dir nvram:
  Directory of nvram:/

    1  -rw-        2703              <no date>  startup-config
    2  ----           5              <no date>  private-config
    3  -rw-        2703              <no date>  underlying-config

  129016 bytes total (126313 bytes free)
```

System File System Management

The "system" file system contains the system memory and the current running configuration. You can display the current configuration (with the **show running-config** or **more system:running-config** EXEC command), save the current configuration to another location (with the **copy system:running-config** *destination-url* EXEC command), and add configuration commands to the current configuration (with the **copy** *source-url* **system:running-config** EXEC command).

The following example changes to the "system" file system, displays the contents of the file system, and displays the running configuration:

```
Router# cd ?
  bootflash:  Directory name
  flash:      Directory name
  lex:        Directory name
  modem:      Directory name
  null:       Directory name
  nvram:      Directory name
  system:     Directory name
  vfc:        Directory name
  <cr>

Router# cd system:?
system:memory  system:running-config  system:ucode  system:vfiles

Router# cd system:
Router# dir
Directory of system:/

    6  dr-x           0              <no date>  memory
    1  -rw-        7786   Apr 22 2001 03:41:39  running-config

No space information available
```

```
nnm3640-2# more system:running-config
!
! No configuration change since last restart
!
version 12.2
service timestamps log datetime localtime
service linenumber
service udp-small-servers
service pt-vty-logging
!
.
.
.
end
```

On some platforms, the system file system contains microcode in its ucode directory, as follows:

```
Router# dir system:/ucode
Directory of system:/ucode/

 21  -r--       22900             <no date>  aip20-13
 18  -r--       32724             <no date>  eip20-3
 25  -r--      123130             <no date>  feip20-6
 19  -r--       25610             <no date>  fip20-1
 22  -r--        7742             <no date>  fsip20-7
 23  -r--       17130             <no date>  hip20-1
 24  -r--       36450             <no date>  mip22-2
 29  -r--      154752             <no date>  posip20-0
 28  -r--      704688             <no date>  rsp220-0
 20  -r--       33529             <no date>  trip20-1
 26  -r--      939130             <no date>  vip22-20
 27  -r--     1107862             <no date>  vip222-20

No space information available
```

Managing Configuration Files

This chapter describes how to create, load, and maintain configuration files. Configuration files contain a set of user-configured commands that customize the functionality of your Cisco routing device using Cisco IOS Release 12.2.

The tasks in this chapter assume that you have at least a minimal configuration running on your system. You can create a basic configuration file using the **setup** command (see the "Using AutoInstall and Setup" chapter in this document for details).

For a complete description of the configuration file management commands in this chapter, refer to the "Configuration File Management Commands" chapter in the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Understanding Configuration Files

Configuration files contain the Cisco IOS software commands used to customize the functionality of your Cisco routing device (router, access server, switch, and so on). Commands are parsed (translated and executed) by the Cisco IOS software when the system is booted (from the startup-config file) or when you enter commands at the CLI in a configuration mode.

Types of Configuration Files

Startup configuration files (startup-config) are used during system startup to configure the software. Running configuration files (running-config) contain the current configuration of the software. The two configuration files can be different. For example, you may want to change the configuration for a short time period rather than permanently. In this case, you would change the running configuration using the **configure terminal** EXEC command but not save the configuration using the **copy running-config startup-config** EXEC command.

To change the running configuration, use the **configure terminal** command, as described in the "Modifying the Configuration File at the CLI" section later in this chapter. As you use the Cisco IOS configuration modes, commands generally are executed immediately and are saved to the running configuration file either immediately after you enter them or when you exit a configuration mode.

To change the startup configuration file, you can either save the running configuration file to the startup configuration using the **copy running-config startup-config** EXEC command or copy a configuration file from a file server to the startup configuration (see the "Copying Configuration Files from a Network Server to the Router" section for more information).

Location of Configuration Files

Configuration files are stored in the following locations:

- The running configuration is stored in RAM.
- On all platforms except the Class A Flash file system platforms, the startup configuration is stored in nonvolatile random-access memory (NVRAM).
- On Class A Flash file system platforms, the startup configuration is stored in the location specified by the CONFIG_FILE environment variable (see the "Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems" section for more information). The CONFIG_FILE variable defaults to NVRAM and can be a file in the following file systems:
  - **nvram:** (NVRAM)
  - **bootflash:** (internal Flash memory)
  - **slot0:** (first PCMCIA slot)
  - **slot1:** (second PCMCIA slot)

Configuration File Management Task List

To understand the management of Cisco IOS software configuration files, perform the tasks described in the following sections:

- Displaying Configuration File Information
- Entering Configuration Mode and Selecting a Configuration Source
- Modifying the Configuration File at the CLI
- Copying Configuration Files from the Router to a Network Server
- Copying Configuration Files from a Network Server to the Router
- Maintaining Configuration Files Larger than NVRAM
- Controlling the Parser Cache
- Copying Configuration Files Between Different Locations
- Reexecuting the Configuration Commands in the Startup Configuration File
- Clearing Configuration Information
- Specifying the Startup Configuration File

Displaying Configuration File Information

To display information about configuration files, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show bootvar`	Lists the contents of the BOOT environment variable, the name of the configuration file pointed to by the CONFIG_FILE environment variable, and the contents of the BOOTLDR environment variable.
`Router# more file-url`	Displays the contents of a specified file.
`Router# show running-config`	Displays the contents of the running configuration file. (Command alias for the **more system:running-config** command.)
`Router# show startup-config`	Displays the contents of the startup configuration file. (Command alias for the **more nvram:startup-config** command.) On all platforms except the Class A Flash file system platforms, the default startup-config file usually is stored in NVRAM. On the Class A Flash file system platforms, the CONFIG_FILE environment variable points to the default startup-config file. The CONFIG_FILE variable defaults to NVRAM.

Entering Configuration Mode and Selecting a Configuration Source

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

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

Configuring from the terminal allows you to enter configuration commands at the command line, as described in the following section. Configuring from memory loads the startup configuration file. See the "Reexecuting the Configuration Commands in the Startup Configuration File" section for more information. Configuring from the network allows you to load and execute configuration commands over the network. See the "Copying Configuration Files from a Network Server to the Router" section for more information.

Modifying the Configuration File at the CLI

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

You can add comments to a configuration file describing the commands you have entered. Precede a comment with an exclamation point (!). Because comments are *not* stored in NVRAM or in the active copy of the configuration file, comments do not appear when you list the active configuration with the **show running-config** or **more system:running-config** EXEC command. Comments do not display when you list the startup configuration with the **show startup-config** or **more nvram:startup-config**

EXEC mode command. Comments are stripped out of the configuration file when it is loaded onto the router. However, you can list the comments in configuration files stored on a File Transfer Protocol (FTP), remote copy protocol (rcp), or Trivial File Transfer Protocol (TFTP) server.

When you configure the software using the CLI, the software executes the commands as you enter them. To configure the software using the CLI, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2		Enter the necessary configuration commands. The Cisco IOS documentation set describes configuration commands organized by technology.

Step 3	`Router(config)# end`
	or
	`Router(config)# ^Z`
		Ends the configuration session and exits to EXEC mode.
		Note When you press the Ctrl and Z keys simultaneously, ^Z is displayed to the screen.

Step 4	`Router# copy system:running-config nvram:startup-config`	Saves the running configuration file as the startup configuration file. You may also use the **copy running-config startup-config** command alias, but you should be aware that this command is less precise. On most platforms, this command saves the configuration to NVRAM. On the Class A Flash file system platforms, this step saves the configuration to the location specified by the CONFIG_FILE environment variable (the default CONFIG_FILE variable specifies that the file should be saved to NVRAM).

In the following example, the router prompt name of the router is configured. The comment line, indicated by the exclamation mark (!), does not execute any command.

In this example, the **hostname** command is used to change the router name from Router to new_name. By pressing Ctrl-Z (^Z) or entering the **end** command, the user quits configuration mode. The **copy system:running-config nvram:startup-config** command saves the current configuration to the startup configuration.

```
Router# configure terminal
Router(config)# !The following command provides the router host name.
Router(config)# hostname new_name
new_name(config)# end
new_name# copy system:running-config nvram:startup-config
```

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

Note Some specific commands might not get saved to NVRAM. You will need to enter these commands again if you reboot the machine. These commands are noted in the documentation. We recommend that you keep a list of these settings so that you can quickly reconfigure your router after rebooting.

Copying Configuration Files from the Router to a Network Server

You can copy configuration files from the router to a file server using FTP, rcp, or TFTP. For example, you might perform this task to back up a current configuration file to a server before changing its contents, thereby allowing you to later restore the original configuration file from the server.

To copy configuration files from a router to a server, perform the tasks described in the following sections:

- Copying a Configuration File from the Router to a TFTP Server
- Copying a Configuration File from the Router to an rcp Server
- Copying a Configuration File from the Router to an FTP Server

The protocol you use depends on which type of server you are using. The FTP and rcp transport mechanisms provide faster performance and more reliable delivery of data than TFTP because FTP and rcp use the TCP/IP stack, which is connection-oriented.

Copying a Configuration File from the Router to a TFTP Server

In some implementations of TFTP, you must create a dummy file on the TFTP server and give it read, write, and execute permissions before copying a file over it. Refer to your TFTP documentation for more information.

To copy configuration information on a TFTP network server, use the following commands in the EXEC mode, as needed:

Command	Purpose
Router# **copy system:running-config tftp:**[[[**//***location*]**/***directory*]**/***filename*]	Copies the running configuration file to a TFTP server.
Router# **copy nvram:startup-config tftp:**[[[**//***location*]**/***directory*]**/***filename*]	Copies the startup configuration file to a TFTP server.

After you have issued the **copy** command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

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

```
Tokyo# copy system:running-config tftp://172.16.2.155/tokyo-confg
Write file tokyo-confg on host 172.16.2.155? [confirm] y

Writing tokyo-confg!!! [OK]
```

Copying a Configuration File from the Router to an rcp Server

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

One of the first attempts to use the network as a resource in the UNIX community resulted in the design and implementation of the remote shell protocol, which included the remote shell (rsh) and remote copy (rcp) functions. Rsh and rcp give users the ability to execute commands remotely and copy files to and from a file system residing on a remote host or server on the network. The Cisco implementation of rsh and rcp interoperates with standard implementations.

The rcp **copy** commands rely on the rsh server (or daemon) on the remote system. To copy files using rcp, you need not create a server for file distribution, as you do with TFTP. You need only to have access to a server that supports the remote shell (rsh). (Most UNIX systems support rsh.) Because you are copying a file from one place to another, you must have read permission on the source file and write permission on the destination file. If the destination file does not exist, rcp creates it for you.

Although the Cisco rcp implementation emulates the functions of the UNIX rcp implementation—copying files among systems on the network—the Cisco command syntax differs from the UNIX rcp command syntax. The Cisco rcp support offers a set of **copy** commands that use rcp as the transport mechanism. These rcp **copy** commands are similar in style to the Cisco TFTP **copy** commands, but they offer an alternative that provides faster performance and reliable delivery of data. These improvements are possible because the rcp transport mechanism is built on and uses the TCP/IP stack, which is connection-oriented. You can use rcp commands to copy system images and configuration files from the router to a network server and vice versa.

You also can enable rcp support to allow users on remote systems to copy files to and from the router.

To configure the Cisco IOS software to allow remote users to copy files to and from the router, use the **ip rcmd rcp-enable** global configuration command.

About the rcp Username

The rcp protocol requires a client to send a remote username on each rcp request to a server. When you copy a configuration file from the router to a server using rcp, the Cisco IOS software sends the first valid username it encounters in the following sequence:

1. The username specified in the **copy** EXEC command, if a username is specified.
2. The username set by the **ip rcmd remote-username** global configuration command, if the command is configured.
3. The remote username associated with the current tty (terminal) process. For example, if the user is connected to the router through Telnet and was authenticated through the **username** command, the router software sends the Telnet username as the remote username.
4. The router host name.

For the rcp copy request to execute successfully, an account must be defined on the network server for the remote username. If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the remote username on the server. For example, if the system image resides in the home directory of a user on the server, you can specify that user name as the remote username.

Use the **ip rcmd remote-username** command to specify a username for all copies. (Rcmd is a UNIX routine used at the super-user level to execute commands on a remote machine using an authentication scheme based on reserved port numbers. Rcmd stands for “remote command”). Include the username in the **copy** command if you want to specify a username for that copy operation only.

If you are writing to the server, the rcp server must be properly configured to accept the rcp write request from the user on the router. For UNIX systems, you must add an entry to the .rhosts file for the remote user on the rcp server. For example, suppose the router contains the following configuration lines:

```
hostname Rtr1
ip rcmd remote-username User0
```

If the router IP address translates to Router1.company.com, then the .rhosts file for User0 on the rcp server should contain the following line:

```
Router1.company.com Rtr1
```

Refer to the documentation for your rcp server for more information.

Copying a Configuration File from the Router to an rcp Server

To copy a startup configuration file or a running configuration file from the router to an rcp server, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	(Optional) Enters global configuration mode.
Step 2	`Router(config)# ip rcmd remote-username username`	(Optional) Changes the default remote username.
Step 3	`Router(config)# end`	(Optional) Exits global configuration mode.
Step 4	`Router# copy system:running-config rcp:[[[//[username@]location]/directory]/filename]`	Specifies that the router running configuration file be stored on an rcp server.
	or	or
	`Router# copy nvram:startup-config rcp:[[[//[username@]location]/directory]/filename]`	Specifies that the router startup configuration file be stored on an rcp server.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Storing a Running Configuration File on an rcp Server Example

The following example copies the running configuration file named rtr2-confg to the netadmin1 directory on the remote host with an IP address of 172.16.101.101:

```
Router# copy system:running-config rcp://netadmin1@172.16.101.101/Rtr2-confg
Write file rtr2-confg on host 172.16.101.101?[confirm]
Building configuration...[OK]
Connected to 172.16.101.101
Router#
```

Storing a Startup Configuration File on an rcp Server Example

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

```
Rtr2# configure terminal
Rtr2(config)# ip rcmd remote-username netadmin2
Rtr2(config)# end
Rtr2# copy nvram:startup-config rcp:
Remote host[]? 172.16.101.101
Name of configuration file to write [rtr2-confg]?
Write file rtr2-confg on host 172.16.101.101?[confirm]
![OK]
```

Copying a Configuration File from the Router to an FTP Server

You can copy a configuration file from the router to an FTP server.

Understanding the FTP Username and Password

The FTP protocol requires a client to send a remote username and password on each FTP request to a server. When you copy a configuration file from the router to a server using FTP, the Cisco IOS software sends the first valid username it encounters in the following sequence:

1. The username specified in the **copy** EXEC command, if a username is specified.
2. The username set by the **ip ftp username** global configuration command, if the command is configured.
3. Anonymous.

The router sends the first valid password it encounters in the following sequence:

1. The password specified in the **copy** command, if a password is specified.
2. The password set by the **ip ftp password** command, if the command is configured.
3. The router forms a password *username@routername.domain*. The variable *username* is the username associated with the current session, *routername* is the configured host name, and *domain* is the domain of the router.

The username and password must be associated with an account on the FTP server. If you are writing to the server, the FTP server must be properly configured to accept the FTP write request from the user on the router.

If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the username on the server. For example, if the system image resides in the home directory of a user on the server, specify that user name as the remote username.

Refer to the documentation for your FTP server for more information.

Use the **ip ftp username** and **ip ftp password** global configuration commands to specify a username and password for all copies. Include the username in the **copy** EXEC command if you want to specify a username for that copy operation only.

Copying a Configuration File from the Router to the FTP Server

To copy a startup configuration file or a running configuration file from the router to an FTP server, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	(Optional) Enters configuration mode from the terminal. This step is required only if you want to override the default remote username or password (see Steps 2 and 3).
Step 2	`Router(config)# ip ftp username` *username*	(Optional) Specifies the default remote username.
Step 3	`Router(config)# ip ftp password` *password*	(Optional) Specifies the default password.

	Command	Purpose
Step 4	`Router(config)# end`	(Optional) Exits global configuration mode. This step is required only if you override the default remote username or password (see Steps 2 and 3).

Step 5	`Router# copy system:running-config ftp:[[[//[username[:password]@]location] /directory]/filename]`
	
	or
	
	`Router# copy nvram:startup-config ftp:[[[//[username[:password]@]location] /directory]/filename]`
		Copies the running configuration or startup configuration file to an FTP server.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Storing a Running Configuration File on an FTP Server Example

The following example copies the running configuration file named rtr2-confg to the netadmin1 directory on the remote host with an IP address of 172.16.101.101:

```
Router# copy system:running-config ftp://netadmin1:mypass@172.16.101.101/Rtr2-confg
Write file rtr2-confg on host 172.16.101.101?[confirm]
Building configuration...[OK]
Connected to 172.16.101.101
Router#
```

Storing a Startup Configuration File on an FTP Server Example

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

```
Rtr2# configure terminal
Rtr2(config)# ip ftp username netadmin2
Rtr2(config)# ip ftp password mypass
Rtr2(config)# end
Rtr2# copy nvram:startup-config ftp:
Remote host[]? 172.16.101.101
Name of configuration file to write [rtr2-confg]?
Write file rtr2-confg on host 172.16.101.101?[confirm]
![OK]
```

Copying Configuration Files from a Network Server to the Router

You can copy configuration files from a TFTP, rcp, or FTP server to the running configuration or startup configuration of the router. You may want to perform this function for one of the following reasons:

- To restore a backed-up configuration file.
- To use the configuration file for another router. For example, you may add another router to your network and want it to have a similar configuration to the original router. By copying the file to the new router, you can change the relevant parts rather than re-creating the whole file.
- To load the same configuration commands on to all the routers in your network so that all the routers have similar configurations.

The **copy** {**ftp:** | **rcp:** | **tftp:**} **system:running-config** EXEC command loads the configuration files into the router as if you were typing the commands in at the command line. The router does not erase the existing running configuration before adding the commands. If a command in the copied configuration file replaces a command in the existing configuration file, the existing command will be erased. For example, if the copied configuration file contains a different IP address in a particular command than the existing configuration, the IP address in the copied configuration will be used. However, some commands in the existing configuration may not be replaced or negated. In this case, the resulting configuration file will be a mixture of the existing configuration file and the copied configuration file, with the copied configuration file having precedence.

In order to restore a configuration file to an exact copy of a file stored on a server, you need to copy the configuration file directly to the startup configuration (using the **copy** {**ftp:** | **rcp:** | **tftp:**} **nvram:startup-config** command) and reload the router.

To copy configuration files from a server to a router, perform the tasks described in the following sections:

- Copying a Configuration File from a TFTP Server to the Router
- Copying a Configuration File from an rcp Server to the Router
- Copying a Configuration File from an FTP Server to the Router

The protocol you use depends on which type of server you are using. The FTP and rcp transport mechanisms provide faster performance and more reliable delivery of data than TFTP. These improvements are possible because the FTP and rcp transport mechanisms are built on and use the TCP/IP stack, which is connection-oriented.

Copying a Configuration File from a TFTP Server to the Router

To copy a configuration file from a TFTP server to the router, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# **copy tftp:**[[[**//***location*]**/***directory*]**/***filename*] **system:running-config**	Copies a configuration file from a TFTP server to the running configuration.
Router# **copy tftp:**[[[**//***location*]**/***directory*]**/***filename*] **nvram:startup-config**	Copies a configuration file from a TFTP server to the startup configuration.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

In the following example, the software is configured from the file named tokyo-config at IP address 172.16.2.155:

```
Router1# copy tftp://172.16.2.155/tokyo-confg system:running-config
Configure using tokyo-confg from 172.16.2.155? [confirm] y
Booting tokyo-confg from 172.16.2.155:!!! [OK - 874/16000 bytes]
```

Copying a Configuration File from an rcp Server to the Router

You can copy configuration files from an rcp server to the router.

Understanding the rcp Username

The rcp protocol requires a client to send a remote username on each rcp request to a server. When you copy a configuration file from the router to a server using rcp, the Cisco IOS software sends the first valid username it encounters in the following sequence:

1. The username specified in the **copy** EXEC command, if a username is specified.
2. The username set by the **ip rcmd remote-username** global configuration command, if the command is configured.
3. The remote username associated with the current tty (terminal) process. For example, if the user is connected to the router through Telnet and was authenticated through the **username** command, the router software sends the Telnet username as the remote username.
4. The router host name.

For the rcp copy request to execute, an account must be defined on the network server for the remote username. If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the remote username on the server. For example, if the system image resides in the home directory of a user on the server, specify that user name as the remote username.

Copying a Configuration File from the rcp Server to the Router

To copy a configuration file from an rcp server to the running configuration or startup configuration, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	(Optional) Enters configuration mode from the terminal. This step is required only if you override the default remote username (see Step 2).
Step 2	`Router(config)# ip rcmd remote-username username`	(Optional) Specifies the remote username.
Step 3	`Router(config)# end`	(Optional) Exits global configuration mode. This step is required only if you override the default remote username (see Step 2).

Step 4	`Router# copy rcp:[[[//[username@]location]/directory]/filename] system:running-config`
	or
	`Router# copy rcp:[[[//[username@]location]/directory]/filename] nvram:startup-config`
		Copies the configuration file from a rcp server to the running configuration or startup configuration.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copy rcp Running-Config Example

The following example copies a configuration file named host1-confg from the netadmin1 directory on the remote server with an IP address of 172.16.101.101, and loads and runs those commands on the router:

```
Router# copy rcp://netadmin1@172.16.101.101/host1-confg system:running-config
Configure using host1-confg from 172.16.101.101? [confirm]
Connected to 172.16.101.101
Loading 1112 byte file host1-confg:![OK]
Router#
%SYS-5-CONFIG: Configured from host1-config by rcp from 172.16.101.101
```

Copy rcp Startup-Config Example

The following example specifies a remote username of netadmin1. Then it copies the configuration file named host2-confg from the netadmin1 directory on the remote server with an IP address of 172.16.101.101 to the startup configuration.

```
Rtr2# configure terminal
Rtr2(config)# ip rcmd remote-username netadmin1
Rtr2(config)# end
Rtr2# copy rcp: nvram:startup-config
Address of remote host [255.255.255.255]? 172.16.101.101
Name of configuration file[rtr2-confg]? host2-confg
Configure using host2-confg from 172.16.101.101?[confirm]
Connected to 172.16.101.101
Loading 1112 byte file host2-confg:![OK]
[OK]
Rtr2#
%SYS-5-CONFIG_NV:Non-volatile store configured from host2-config by rcp from
172.16.101.101
```

Copying a Configuration File from an FTP Server to the Router

You can copy configuration files from an FTP server to the router.

Understanding the FTP Username and Password

The FTP protocol requires a client to send a remote username and password on each FTP request to a server. When you copy a configuration file from the router to a server using FTP, the Cisco IOS software sends the first valid username it encounters in the following sequence:

1. The username specified in the **copy** EXEC command, if a username is specified.
2. The username set by the **ip ftp username** global configuration command, if the command is configured.
3. Anonymous.

The router sends the first valid password it encounters in the following sequence:

1. The password specified in the **copy** EXEC command, if a password is specified.
2. The password set by the **ip ftp password** global configuration command, if the command is configured.
3. The router forms a password *username@routername.domain*. The variable *username* is the username associated with the current session, *routername* is the configured host name, and *domain* is the domain of the router.

The username and password must be associated with an account on the FTP server. If you are writing to the server, the FTP server must be properly configured to accept the FTP write request from the user on the router.

If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the username on the server. For example, if the system image resides in the home directory of a user on the server, specify that user name as the remote username.

Refer to the documentation for your FTP server for more information.

Use the **ip ftp username** and **ip ftp password** global configuration commands to specify a username and password for all copies. Include the username in the **copy** command if you want to specify a username for that copy operation only.

Copying a Configuration File from an FTP Server to the Router

To copy a configuration file from an FTP server to the running configuration or startup configuration, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	(Optional) Allows you to enter global configuration mode. This step is required only if you want to override the default remote username or password (see Steps 2 and 3).
Step 2	`Router(config)# ip ftp username username`	(Optional) Specifies the default remote username.
Step 3	`Router(config)# ip ftp password password`	(Optional) Specifies the default password.
Step 4	`Router(config)# end`	(Optional) Exits global configuration mode. This step is required only if you override the default remote username or password (see Steps 2 and 3).

Step 5	`Router# copy ftp:[[[//[username[:password]@]location]/directory]/filename] system:running-config`
	or
	`Router# copy ftp:[[[//[username[:password]@]location]/directory]/filename] nvram:startup-config`
		Using FTP, copies the configuration file from a network server to running memory or the startup configuration.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copy FTP Running-Config Example

The following example copies a host configuration file named host1-confg from the netadmin1 directory on the remote server with an IP address of 172.16.101.101, and loads and runs those commands on the router:

```
Router# copy rcp://netadmin1:mypass@172.16.101.101/host1-confg system:running-config
Configure using host1-confg from 172.16.101.101? [confirm]
Connected to 172.16.101.101
Loading 1112 byte file host1-confg:![OK]
Router#
%SYS-5-CONFIG: Configured from host1-config by ftp from 172.16.101.101
```

Copy FTP Startup-Config Example

The following example specifies a remote username of netadmin1. Then it copies the configuration file named host2-confg from the netadmin1 directory on the remote server with an IP address of 172.16.101.101 to the startup configuration.

```
Rtr2# configure terminal
Rtr2(config)# ip ftp username netadmin1
Rtr2(config)# ip ftp password mypass
Rtr2(config)# end
Rtr2# copy ftp: nvram:startup-config
Address of remote host [255.255.255.255]? 172.16.101.101
Name of configuration file[rtr2-confg]? host2-confg
Configure using host2-confg from 172.16.101.101?[confirm]
Connected to 172.16.101.101
Loading 1112 byte file host2-confg:![OK]
[OK]
Rtr2#
%SYS-5-CONFIG_NV:Non-volatile store configured from host2-config by ftp from
172.16.101.101
```

Maintaining Configuration Files Larger than NVRAM

To maintain a configuration file that exceeds size of NVRAM, perform the tasks described in the following sections:

- Compressing the Configuration File
- Storing the Configuration in Flash Memory on Class A Flash File Systems
- Loading the Configuration Commands from the Network

Compressing the Configuration File

The **service compress-config** global configuration command specifies that the configuration file be stored compressed in NVRAM. Once the configuration file has been compressed, the router functions normally. When the system is booted, it recognizes that the configuration file is compressed, expands it, and proceeds normally. The **more nvram:startup-config** EXEC command expands the configuration before displaying it.

Before you compress configuration files, refer to the appropriate hardware installation and maintenance publication. Verify that your system's ROMs support file compression. If not, you can install new ROMs that support file compression.

To compress configuration files, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# service compress-config`	Specifies that the configuration file be compressed.
Step 2	`Router(config)# end`	Exits global configuration mode.

	Command	Purpose
Step 3	Use FTP, rcp, or TFTP to copy the new configuration. If you try to load a configuration that is more than three times larger than the NVRAM size, the following error message is displayed: "[buffer overflow - *file-size*/*buffer-size* bytes]." or `Router# configure terminal`	Enters the new configuration.
Step 4	`Router(config)# copy system:running-config nvram:startup-config`	When you have finished changing the running-configuration, saves the new configuration.

The size of the configuration must not exceed three times the NVRAM size. For a 128-KB size NVRAM, the largest expanded configuration file size is 384 KB.

The **service compress-config** global configuration command works only if you have Cisco IOS software Release 10 or later release boot ROMs. Installing new ROMs is a one-time operation and is necessary only if you do not already have Cisco IOS Release 10 in ROM. If the boot ROMs do not recognize a compressed configuration, the following message is displayed:

```
Boot ROMs do not support NVRAM compression Config NOT written to NVRAM
```

The following example compresses a 129-KB configuration file to 11 KB:

```
Router# configure terminal
Router(config)# service compress-config
Router(config)# end
Router# copy tftp://172.16.2.15/tokyo-confg system:running-config
Configure using tokyo-confg from 172.16.2.155? [confirm] y
Booting tokyo-confg from 172.16.2.155:!!! [OK - 874/16000 bytes]
Router# copy system:running-config nvram:startup-config
Building configuration...
Compressing configuration from 129648 bytes to 11077 bytes
[OK]
```

Storing the Configuration in Flash Memory on Class A Flash File Systems

On Class A Flash file system routers, you can store the startup configuration in Flash memory by setting the CONFIG_FILE environment variable to a file in internal Flash memory or Flash memory in a PCMCIA slot.

To store the startup configuration in Flash memory, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# copy nvram:startup-config` *flash-filesystem:filename*	Copies the current startup configuration to the new location to create the configuration file.
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# boot config flash-filesystem:`*filename*	Specifies that the startup configuration file be stored in Flash memory by setting the CONFIG_FILE variable.

	Command	Purpose
Step 4	`Router(config)# end`	Exits global configuration mode.

Step 5	`Use FTP, rcp, or TFTP to copy the new configuration. If you try to load a configuration that is more than three times larger than the NVRAM size, the following error message is displayed:`
	`"[buffer overflow - file-size/buffer-size bytes]."`
	
	or
	
	`Router# configure terminal`
		Enters the new configuration.

Step 6	`Router# copy system:running-config nvram:startup-config`	When you have finished changing the running-configuration, saves the new configuration.

See the "Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems" section for more information.

The following example stores the configuration file in slot 0:

```
Router# copy nvram:startup-config slot0:router-config
Router# configure terminal
Router(config)# boot config slot0:router-config
Router(config)# end
Router# copy system:running-config nvram:startup-config
```

Care must be taken when editing or changing a large configuration. Flash memory space is used every time a **copy system:running-config nvram:startup-config** EXEC command is issued. Because file management for Flash memory, such as optimizing free space, is not done automatically, you must pay close attention to available Flash memory. Use the **squeeze** command to reclaim used space. We recommend that you use a large-capacity Flash card of at least 20 MB.

Loading the Configuration Commands from the Network

You can also store large configurations on FTP, rcp, or TFTP servers and download them at system startup. To use a network server to store large configurations, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# copy system:running-config {ftp: | rcp: | tftp:}`	Saves the running configuration to an FTP, rcp, or TFTP server.
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# boot network {ftp:[[[//[username[:password]@]location]/directory]/filename] | rcp:[[[//[username@]location]/directory]/filename] | tftp:[[[//location]/directory]/filename]}`	Specifies that the startup configuration file be loaded from the network server at startup.
Step 4	`Router(config)# service config`	Enables the router to download configuration files at system startup.
Step 5	`Router(config)# end`	Exits global configuration mode.
Step 6	`Router# copy system:running-config nvram:startup-config`	Saves the configuration.

See the "Copying Configuration Files from the Router to a Network Server" and "Configuring the Router to Download Configuration Files" sections for more information on these commands.

Controlling the Parser Cache

The Cisco IOS command-line parser in the Cisco IOS software performs the translation and execution (parsing) of command lines. The Parser Cache feature was developed to rapidly process large configuration files, thereby dramatically improving load time.

The Parser Cache feature allows the rapid recognition and translation of configuration lines in a configuration file that differ slightly from previously used configuration lines (for example, pvc 0/100, pvc 0/101, and so on) by dynamically creating, caching, and reusing simplified parse graphs. This improvement is useful primarily for configuration files that repeat similar commands hundreds or thousands of times, such as cases in which thousands of virtual circuits must be configured for subinterfaces, or hundreds of access lists must be configured. Performance will improve the most for those files in which the same commands are used repeatedly but the numerical arguments change from command to command.

The Parser Cache is enabled by default on all platforms using Cisco IOS Release 12.1(5)T and later releases. However, users with Cisco devices that do not require large configuration files may want to disable the Parser Cache to free the resources used by this feature. (Memory used by this feature depends on the size of the configuration files parsed, but is generally less than 512 KB.)

To control the Parser Cache feature, perform the tasks described in the following sections. All of these tasks are optional:

- Clearing the Parser Cache
- Disabling the Parser Cache
- Reenabling the Parser Cache
- Monitoring the Parser

Clearing the Parser Cache

To free resources or to reset the parser cache memory, you may wish to clear the parse entries and hit/miss statistics stored by the Parser Cache feature. To clear the information stored by the Parser Cache feature, use the following command in privileged EXEC mode:

Command	Purpose
`Router# clear parser cache`	Clears the parse cache entries and hit/miss statistics stored for the Parser Cache feature.

Disabling the Parser Cache

The Parser Cache feature is enabled by default. To disable the Parser Cache feature, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no parser cache`	Disables the Parser Cache feature.

When the parser cache is disabled, the **no parser cache** command line is written to the running configuration file.

Tips If you wish to disable the parser cache to free system resources, you should clear the parser cache before issuing the **no parser cache** command. You will not be able to clear the parser cache after disabling it.

Reenabling the Parser Cache

To reenable the Parser Cache feature after disabling it, use the following command in global configuration mode:

Command	Purpose
`Router(config)# parser cache`	Enables the Parser Cache feature.

Monitoring the Parser

Statistics about the last configuration file parsed are kept in the system memory, along with hit/miss statistics on the commands parsed by the Parser Cache feature. “Hits” and “misses” refer to the matches that the parser cache was able to make to similar commands used previously in the configuration session. Those commands that are matched (“hits”) be parsed more efficiently. The parser cache cannot improve the parse time for those commands it was unable to match (“misses”).

To display the parser statistics, use the following command in privileged EXEC mode:

Command	Purpose
`Router# show parser statistics`	Displays statistics about the last configuration file parsed and the status of the Parser Cache feature.

The following example shows sample output from the **show parser statistics** command:

```
Router# show parser statistics
Last configuration file parsed:Number of Commands:1484, Time:1272 ms

Parser cache:disabled, 0 hits, 0 misses
```

The **show parser statistics** command displays two sets of data, as follows:

- The number of commands in the configuration file that was last copied into the running configuration, and the time it took for the system to parse them (a configuration file can be loaded into the running configuration at system startup, or by issuing commands such as the **copy** *source* **running-config** EXEC command).

- The status of the parser cache (enabled or disabled) and the number of command matches (hits or misses) since the system was started or since the parser cache was cleared.

In the example shown, the hit/miss statistics (0/0) do not match the number of commands in the last configuration file parsed (1484), which indicates that the last configuration file was loaded while the parser cache was disabled.

Copying Configuration Files Between Different Locations

On many platforms, you can copy configuration files from one Flash memory device, such as internal Flash memory or a Flash memory card in a PCMCIA slot, to other locations. You also can copy configuration files from an FTP, rcp, or TFTP server to Flash memory.

Copying Configuration Files from Flash Memory to the Startup or Running Configuration

To copy a configuration file from Flash memory directly to your startup configuration in NVRAM or your running configuration, enter one following commands in EXEC mode, as needed:

Command	Purpose
Router> **copy** *filesystem*:[*partition-number*:][*filename*] **nvram:startup-config**	Loads a configuration file directly into NVRAM.
Router> **copy** *filesystem*:[*partition-number*:][*filename*] **system:running-config**	Copies a configuration file to your running configuration.

The following example copies the file named ios-upgrade-1 from partition 4 of the Flash memory PC Card in slot 0 to the router startup configurations:

```
Router# copy slot0:4:ios-upgrade-1 nvram:startup-config

Copy 'ios-upgrade-1' from flash device
  as 'startup-config' ? [yes/no] yes
[OK]
```

Copying Configuration Files Between Flash Memory File Systems

On platforms with multiple Flash memory file systems, you can copy files from one Flash memory file system, such as internal Flash memory or a Flash memory card in a PCMCIA slot, to another Flash memory file system. Copying files to different Flash memory file systems lets you create backup copies of working configurations and duplicate configurations for other routers.

To copy a configuration file between Flash memory file systems, use the following commands in EXEC mode:

	Command	Purpose
Step 1	`Router> show source-filesystem:`	Displays the layout and contents of Flash memory to verify the filename.
Step 2	`Router> copy source-filesystem:[partition-number:][filename] dest-filesystem:[partition-number:][filename]`	Copies a configuration file between Flash memory devices.
Step 3	`Router> verify dest-filesystem:[partition-number:][filename]`	Verifies the checksum of the file you copied.

Note The source device and the destination device cannot be the same. For example, the **copy slot1: slot1:** command is invalid.

Copying a Configuration File Between Local Flash Memory Devices Example

The following example copies the file named running-config from partition 1 of internal Flash memory to partition 1 of slot 1 on a Cisco 3600 series router. In this example, the source partition is not specified, so the router prompts for the partition number.

```
Router# copy flash: slot1:

System flash

Partition   Size    Used      Free      Bank-Size  State          Copy Mode
  1          4096K   3070K     1025K     4096K      Read/Write     Direct
  2         16384K   1671K    14712K     8192K      Read/Write     Direct

[Type ?<no> for partition directory; ? for full directory; q to abort]
Which partition? [default = 1]

System flash directory, partition 1:
File  Length   Name/status
  1   3142748  dirt/network/mars-test/c3600-j-mz.latest
  2   850      running-config
[3143728 bytes used, 1050576 available, 4194304 total]

PCMCIA Slot1 flash directory:
File  Length   Name/status
  1   1711088  dirt/gate/c3600-i-mz
  2   850      running-config
[1712068 bytes used, 2482236 available, 4194304 total]

Source file name? running-config

Destination file name [running-config]?
Verifying checksum for 'running-config' (file # 2)...  OK

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

Copy 'running-config' from flash: device
  as 'running-config' into slot1: device WITH erase? [yes/no] yes
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
...erased
!
 [OK - 850/4194304 bytes]
```

```
Flash device copy took 00:00:30 [hh:mm:ss]
Verifying checksum...  OK (0x16)
```

Copying a Configuration File from a Server to Flash Memory Devices

To copy a configuration file from an FTP server to a Flash memory device, use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	(Optional) Enters global configuration mode. This step is required only if you override the default remote username or password (see Steps 2 and 3).
Step 2	`Router(config)# ip ftp username username`	(Optional) Specifies the remote username.
Step 3	`Router(config)# ip ftp password password`	(Optional) Specifies the remote password.
Step 4	`Router(config)# end`	(Optional) Exits configuration mode. This step is required only if you override the default remote username (see Steps 2 and 3).
Step 5	`Router# copy ftp:` `[[[//[username:password@]location]/directory]/file name]` `flash-filesystem:[partition-number:][filename]`	Copies the configuration file from a network server to the Flash memory device using FTP.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

To copy a configuration file from an rcp server to a Flash memory device, use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	(Optional) Enters global configuration mode. This step is required only if you override the default remote username (see Step 2).
Step 2	`Router(config)# ip rcmd remote-username username`	(Optional) Specifies the remote username.
Step 3	`Router(config)# end`	(Optional) Exits global configuration mode. This step is required only if you override the default remote username (see Step 2).
Step 4	`Router# copy` `rcp:[[[//[username@]location]/directory]/filename]` `flash-filesystem:[partition-number:][filename]`	Copies the configuration file from a network server to the Flash memory device using rcp. Reply to any router prompts for additional information or confirmation. The prompting will depending on how much information you provide in the **copy** command and the current setting of the **file prompt** command.

To copy a configuration file from a TFTP server to the router, use the following command in EXEC mode:

Command	Purpose
Router> **copy tftp:**[[[**//***location*]**/***directory*]**/***filename*] *flash-filesystem***:**[*partition-number***:**][*filename*]	Copies the file from a TFTP server to the Flash memory device. Reply to any router prompts for additional information or confirmation. The prompting will depending on how much information you provide in the **copy** command and the current setting of the **file prompt** command.

The following example shows the copying of the configuration file named router-config from a TFTP server to the Flash memory card inserted in slot 0 of the Network Processing Engine (NPE) or Route Switch Processor (RSP) card of a Cisco 7500 series router. The copied file is renamed new-config.

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

Reexecuting the Configuration Commands in the Startup Configuration File

To reexecute the commands located in the startup configuration file, use the following command in privileged EXEC m ode:

Command	Purpose
Router# **configure memory**	Reexecutes the configuration commands located in the startup configuration file.

Clearing Configuration Information

You can clear the configuration information from the startup configuration. If you reboot the router with no startup configuration, the router will enter the Setup command facility so that you can configure the router from scratch.

Clearing the Startup Configuration

To clear the contents of your startup configuration, use the following command in EXEC mode:

Command	Purpose
Router> **erase nvram:**	Clears the contents of your startup configuration.

For all platforms except the Class A Flash file system platforms, this command erases NVRAM. The startup configuration file cannot be restored once it has been deleted.

On Class A Flash file system platforms, when you use the **erase startup-config** EXEC command, the router erases or deletes the configuration pointed to by CONFIG_FILE environment variable. If this variable points to NVRAM, the router erases NVRAM. If the CONFIG_FILE environment variable

specifies a Flash memory device and configuration filename, the router deletes the configuration file. That is, the router marks the file as "deleted," rather than erasing it. This feature allows you to recover a deleted file.

Deleting a Specified Configuration File

To delete a specified configuration on a specific Flash device, use the following command in EXEC mode:

Command	Purpose
`Router> delete flash-filesystem:filename`	Deletes a specified configuration file on a specified Flash device.

On Class A and B Flash file systems, when you delete a specific file in Flash memory, the system marks the file as deleted, allowing you to later recover a deleted file using the **undelete** EXEC command. Erased files cannot be recovered. To permanently erase the configuration file, use the **squeeze** EXEC command.

On Class C Flash file systems, you cannot recover a file that has been deleted.

If you attempt to erase or delete the configuration file specified by the CONFIG_FILE environment variable, the system prompts you to confirm the deletion.

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

```
Router# delete slot0:myconfig
```

Specifying the Startup Configuration File

Normally, the router uses the startup configuration file in NVRAM or the Flash file system specified by the CONFIG_FILE environment variable (Class A Flash file systems only) at startup. See the "Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems" section for more information on setting the CONFIG_FILE variable.

You can also configure the router to automatically request and receive two configuration files from the network server at startup. See the "Configuring the Router to Download Configuration Files" section for more information.

Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems

On Class A Flash file systems, you can configure the Cisco IOS software to load the startup configuration file specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM. To change the CONFIG_FILE environment variable, use the following commands beginning in EXEC mode:

	Command	Purpose
Step 1	`Router> copy [flash-url | ftp-url | rcp-url | tftp-url | system:running-config | nvram:startup-config] dest-flash-url`	Copies the configuration file to the Flash file system from which the router will load the file upon restart.
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# boot config dest-flash-url`	Sets the CONFIG_FILE environment variable. This step modifies the runtime CONFIG_FILE environment variable.
Step 4	`Router(config)# end`	Exits global configuration mode.
Step 5	`Router> copy system:running-config nvram:startup-config`	Saves the configuration performed in Step 3 to the startup configuration.
Step 6	`Router> show bootvar`	(Optional) Allows you to verify the contents of the CONFIG_FILE environment variable.

After you specify a location for the startup configuration file, the **nvram:startup-config** command is aliased to the new location of the startup configuration file. The **more nvram:startup-config** EXEC command will display the startup configuration, regardless of its location. The **erase nvram:startup-config** EXEC command will erase the contents of NVRAM and delete the file pointed to by the CONFIG_FILE environment variable.

When you save the configuration using the **copy system:running-config nvram:startup-config** command, the router saves a complete version of the configuration file to the location specified by the CONFIG_FILE environment variable and a distilled version to NVRAM. A distilled version is one that does not contain access list information. If NVRAM contains a complete configuration file, the router prompts you to confirm your overwrite of the complete version with the distilled version. If NVRAM contains a distilled configuration, the router does not prompt you for confirmation and proceeds with overwriting the existing distilled configuration file in NVRAM.

Note If you specify a file in a Flash device as the CONFIG_FILE environment variable, every time you save your configuration file with the **copy system:running-config nvram:startup-config** command, the old configuration file is marked as "deleted," and the new configuration file is saved to that device. Eventually, Flash memory will be full, because the old configuration files still take up memory. Use the **squeeze** EXEC command to permanently delete the old configuration files and reclaim the space.

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

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

Configuration register is 0x010F
```

Configuring the Router to Download Configuration Files

You can configure the router to load one or two configuration files at system startup. The configuration files are loaded into memory and read in as if you were typing the commands at the command line. Thus, the configuration for the router will be a mixture of the original startup configuration and the one or two downloaded configuration files.

Network Versus Host Configuration Files

For historical reasons, the first file the router downloads is called the network configuration file. The second file the router downloads is called the host configuration file. Two configuration files can be used when all of the routers on a network use many of the same commands. The network configuration file contains the standard commands used to configure all of the routers. The host configuration files contain the commands specific to one particular host. If you are loading two configuration files, the host configuration file should be the configuration file you want to have precedence over the other file. Both the network and host configuration files must reside on a network server reachable via TFTP, rcp, or FTP, and must be readable.

Understanding the rcp Username

The rcp protocol requires a client to send a remote username on each rcp request to a server. When you copy a configuration file from the router to a server using rcp, the Cisco IOS software sends the first valid username it encounters in the following sequence:

1. The username specified in the **boot network** or **boot host** global configuration command, if a username is specified.
2. The username set by the **ip rcmd remote-username** global configuration command, if the command is configured.
3. The remote username associated with the current tty (terminal) process. For example, if the user is connected to the router through Telnet and was authenticated through the **username** command, the router software sends the Telnet username as the remote username.
4. The router host name.

For the rcp copy request to execute, an account must be defined on the network server for the remote username. If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the remote username on the server. For example, if the system image resides in the home directory of a user on the server, specify that user name as the remote username.

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

Understanding the FTP Username and Password

The FTP protocol requires a client to send a remote username and password on each FTP request to a server. When you copy a configuration file from the router to a server using FTP, the Cisco IOS software sends the first valid username it encounters in the following sequence:

1. The username specified in the **copy** EXEC command, if a username is specified.
2. The username set by the **ip ftp username** global configuration command, if the command is configured.
3. Anonymous.

The router sends the first valid password in the following list:

1. The password specified in the **copy** command, if a password is specified.
2. The password set by the **ip ftp password** command, if the command is configured.
3. The router forms a password *username@routername.domain*. The variable *username* is the username associated with the current session, *routername* is the configured host name, and *domain* is the domain of the router.

The username and password must be associated with an account on the FTP server. If you are writing to the server, the FTP server must be properly configured to accept the FTP write request from the user on the router.

If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the username on the server. For example, if the system image resides in the home directory of a user on the server, specify that user name as the remote username.

Refer to the documentation for your FTP server for more information.

Use the **ip ftp username** and **ip ftp password** commands to specify a username and password for all copies. Include the username in the **copy** command if you want to specify a username for that copy operation only.

Configuring the Router to Download Configuration Files

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

To configure the router to download configuration files at system startup, perform at least one of the tasks described in the following sections:

- Configuring the Router to Download the Network Configuration File
- Configuring the Router to Download the Host Configuration File

If the router fails to load a configuration file during startup, it tries again every 10 minutes (the default setting) until a host provides the requested files. With each failed attempt, the router displays the following message on the console terminal:

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

Refer to the *Internetwork Troubleshooting Guide* for troubleshooting procedures.

If there are any problems with the startup configuration file, or if the configuration register is set to ignore NVRAM, the router enters the Setup command facility. See the "Using the Setup Command Facility for Configuration Changes" chapter in this publication for details on the Setup command facility.

Configuring the Router to Download the Network Configuration File

To configure the Cisco IOS software to download a network configuration file from a server at startup, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# boot network {ftp:[[[//[username[:password]@]location]/directory]/filename] | rcp:[[[//[username@]location]/directory]/filename] | tftp:[[[//location]/directory]/filename]}`	Specifies the network configuration file to download at startup, and the protocol to be used (TFTP, rcp, or FTP).
Step 3	`Router(config)# service config`	Enables the system to automatically load the network file upon restart.
Step 4	`Router(config)# end`	Exits global configuration mode.
Step 5	`Router# copy system:running-config nvram:startup-config`	Saves the running configuration to the startup configuration file.

For Step 2, if you do not specify a network configuration filename, the Cisco IOS software uses the default filename network-confg. If you omit the address, the router uses the broadcast address.

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

Configuring the Router to Download the Host Configuration File

To configure the Cisco IOS software to download a host configuration file from a server at startup, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# boot host {ftp:[[[//[username[:password]@]location]/directory]/filename] | rcp:[[[//[username@]location]/directory]/filename] | tftp:[[[//location]/directory]/filename] }`	Specifies the host configuration file to download at startup, and the protocol to be used (FTP, rcp, or TFTP).
Step 3	`Router(config)# service config`	Enables the system to automatically load the host file upon restart.
Step 4	`Router(config)# end`	Exits global configuration mode.
Step 5	`Router# copy system:running-config nvram:startup-config`	Saves the running configuration to the startup configuration file.

If you do not specify a host configuration filename, the router uses its own name to form a host configuration filename by converting the name to all lowercase letters, removing all domain information, and appending "-confg." If no host name information is available, the software uses the default host configuration filename router-confg. If you omit the address, the router uses the broadcast address.

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

Configuring the Router to Download Configuration Files at System Startup Example

In the following example, a router is configured to download the host configuration file named hostfile1 and the network configuration file named networkfile1. The router uses TFTP and the broadcast address to obtain the file.

```
Router# configure terminal
Router(config)# boot host tftp:hostfile1
Router(config)# boot network tftp:networkfile1
Router(config)# service config
Router(config)# end
Router# copy system:running-config nvram:startup-config
```

Loading and Maintaining System Images

This document applies to: Cisco IOS Software, Release 12.2

Published: 04/30/2001

This chapter describes how to load and maintain system images and microcode. System images contain the system software. Microcode typically contains system images or hardware-specific software that can be loaded directly on to various hardware devices.

For a complete description of the system image and microcode commands mentioned in this chapter, refer to the "System Image and Microcode Commands" chapter in the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Understanding Images

System images contain the Cisco IOS software. Your router already has an image on it when you receive it. However, you may want to load a different image onto the router at some point. For example, you may want to upgrade your software to the latest release, or use the same version of the software for all the routers in a network. Different system images contain different sets of Cisco IOS features. To determine which version (release number) of Cisco IOS is currently running on your system, and the filename of the system image, use the **show version** command in Exec mode. For example, "Version 12.2" indicates Cisco IOS Release 12.2, and "c7200-js-mz" indicates the system image for a Cisco 7200 series router (c7200) containing the "enterprise" feature set (jz).

Types of Images

The following are the two main types of image your router may use:

- System image—The complete Cisco IOS software. This image is loaded when your router boots and is used most of the time.

On most platforms, the image is located in Flash memory. On platforms with multiple Flash memory file systems (Flash, boot flash, slot 0, or slot 1), the image can be located in any existing Flash file system. Use the **show file systems** EXEC command to determine which file systems your router supports. Refer to your hardware documentation for information about where these images are located by default.

- Boot image—A subset of the Cisco IOS software. This image is used to perform network booting or to load Cisco IOS images onto the router. This image is also used if the router cannot find a valid system image. Depending on your platform, this image may be called xboot image, rxboot image, bootstrap image, or boot loader/helper image.

  On some platforms, the boot image is contained in ROM. In others, the boot image can be stored in Flash memory. On these platforms, you can specify which image should be used as the boot image using the **boot bootldr** global configuration command. Refer to your hardware documentation for information about the boot image used on your router.

Image Naming Conventions

You can identify the platform, features, and image location by the name of the image. The naming convention is *platform-features-type* for images that are stored on a UNIX system

The *platform* variable indicates which platforms can use this image. Examples of *platform* variables include rsp (Cisco 7000 series with RSP7000 and Cisco 7500 series), c1600 (Cisco 1600 series), and c1005 (Cisco 1005).

The *features* variable identifies the feature sets supported by the image.

The *type* field can contain the following characters:

- f—The image runs from Flash memory.
- m—The image runs from RAM.
- r—The image runs from ROM.
- l—The image is relocatable.
- z—The image is zip compressed.
- x—The image is mzip compressed.

General Output Conventions for Copy Operations

During a copy operation, any of the following characters may be printed to the screen:

- A pound sign (#) generally means that a Flash memory device is being cleared and initialized. (Different platforms use different ways of indicating that Flash is being cleared.)
- An exclamation point (!) means that ten packets have been transferred.
- A series of “V” characters means that a checksum verification of the file is occurring after the file is written to Flash memory.
- An “O” means an out-of-order packet.
- A period (.) means a timeout.

The last line in the output indicates whether the copy was successful.

System Images Task List

To manage system images, perform any of the tasks in the following sections:

- Displaying System Image Information
- Copying Images from Flash Memory to a Network Server
- Copying Images from a Network Server to Flash Memory
- Copying Images Between Local Flash Memory Devices
- Specifying the Startup System Image in the Configuration File
- Recovering a System Image Using Xmodem or Ymodem
- Loading and Displaying Microcode Images

Displaying System Image Information

Use the following commands in EXEC mode to display information about system software:

Command	Purpose
`Router# show bootvar`	Lists the contents of the BOOT environment variable, the name of the configuration file pointed to by the CONFIG_FILE environment variable, and the contents of the BOOTLDR environment variable.
`Router# show flash-filesystem: [partition number] [all | chips | detailed | err | summary]`	Lists information about Flash memory for Class B file systems.
`Router# show flash-filesystem: [all | chips | filesys]`	Lists information about Flash memory for Class A file systems.
`Router# show flash-filesystem:`	Lists information about Flash memory for Class C file systems.
`Router# show microcode`	Displays microcode information.
`Router# show version`	Lists the currently running system image filename, and the system software release version, the configuration register setting, and other information.

Refer to the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference* for examples of these commands.

Copying Images from Flash Memory to a Network Server

You can copy system images from Flash memory to an File Transfer Protocol (FTP), remote copy protocol (rcp), or Trivial File Transfer Protocol (TFTP) server. You can use this server copy of the system image as a backup copy, or you can use it to verify that the copy in Flash is the same as the original file on disk. The following sections describe these tasks:

- Copying an Image from Flash Memory to a TFTP Server
- Copying an Image from Flash Memory to an rcp Server
- Copying an Image from Flash Memory to an FTP Server

The protocol you use depends on which type of server you are using. The FTP and rcp transport mechanisms provide faster performance and more reliable delivery of data than TFTP. These improvements are possible because the FTP and rcp transport mechanisms are built on and use the TCP/IP stack, which is connection-oriented.

To stop the copy process, press **Ctrl**-**^** or **Ctrl-Shift-6**.

In the output, an exclamation point (!) indicates that the copy process is taking place. Each exclamation point (!) indicates that ten packets have been transferred.

Refer to the *Internetwork Troubleshooting Guide* publication for procedures on how to resolve Flash memory problems.

Copying an Image from Flash Memory to a TFTP Server

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

To copy a system image to a TFTP network server, use the following commands in EXEC mode:

	Command	Purpose
Step 1	`Router# show flash-filesystem:`	(Optional) Displays the system image filename in Flash memory. Use this command to verify the url-path of the file and the exact spelling of the system image filename for use in the next command.
Step 2	`Router# copy flash-url tftp:[[[//location]/directory]/filename]`	Copies the system image from Flash memory to a TFTP server. Specify the file location and filename as the *flash-url* argument.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copying an Image from Flash Memory to a TFTP Server Example

The following example uses the **show flash:** EXEC command to learn the name of the system image file and the **copy flash: tftp:** EXEC command to copy the system image to a TFTP server:

```
RouterB# show flash:
```

```
System flash directory:
File  Length   Name/status
  1   4137888  c3640-c2is-mz.Feb24
[4137952 bytes used, 12639264 available, 16777216 total]
16384K bytes of processor board System flash (Read/Write)\

Router# copy flash: tftp:
IP address of remote host [255.255.255.255]? 172.16.13.110
filename to write on tftp host? c3640-c2is-mz.Feb24
writing c3640-c2is-mz.Feb24 !!!!...
successful tftp write.
```

Copying an Image from Partitioned Flash Memory to a TFTP Server Example

In this example, the file named your-ios is copied from partition 1 of the Flash memory PC card in slot 0 to the TFTP server at 172.23.1.129. The file will be saved with the name your-ios in the dirt/sysadmin directory relative to the directory of the remote username.

```
Router# copy slot0:1:your-ios tftp://172.23.1.129/dirt/sysadmin/your-ios
Verifying checksum for 'your-ios' (file # 1)...  OK
Copy 'your-ios' from Flash to server
  as 'dirt/sysadmin/ios-2'? [yes/no] yes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Upload to server done
Flash device copy took 00:00:23 [hh:mm:ss]
```

Copying an Image from Flash Memory to an rcp Server

You can copy a system image from Flash memory to an rcp network server.

If you copy the configuration file to a PC used as a file server, the computer must support remote shell protocol (rsh).

The rcp protocol requires a client to send a remote username on each rcp request to a server. When you copy an image from the router to a server using rcp, the Cisco IOS software sends the first valid username it encounters in the following list:

1. The remote username specified in the **copy** EXEC command, if one if specified.
2. The username set by the **ip rcmd remote-username** global configuration command, if the command is configured.
3. The remote username associated with the current tty (terminal) process. For example, if the user is connected to the router through Telnet and was authenticated through the **username** global configuration command, the router software sends the Telnet username as the remote username.
4. The router host name.

For the rcp copy request to execute, an account must be defined on the network server for the remote username. If the server has a directory structure, the configuration file or image is written or copied relative to the directory associated with the remote username on the server. The path for all files and images to be copied begins at the remote user's home directory. For example, if the system image resides in the home directory of a user on the server, specify that user's name as the remote username.

If you are writing to the server, the rcp server must be properly configured to accept the rcp write request from the user on the router. For UNIX systems, you must add an entry to the .rhosts file for the remote user on the rcp server. For example, suppose the router contains the following configuration lines:

```
hostname Rtr1
ip rcmd remote-username User0
```

If the router's IP address translates to Router1.domain.com, then the .rhosts file for User0 on the rcp server should contain the following line:

```
Router1.domain.com Rtr1
```

Refer to the documentation for your rcp server for more information.

To copy a system image from Flash memory to a network server, use the following commands:

	Command	Purpose
Step 1	`Router# show flash-filesystem:`	(Optional) Displays the system image filename in Flash memory. Use this command to verify the *url-path* of the file and the exact spelling of the system image filename for use in the **copy** EXEC command.
Step 2	`Router# configure terminal`	(Optional) Enters global configuration mode from the terminal. This step is required only if you want to change the default remote username (see Step 3).
Step 3	`Router(config)# ip rcmd remote-username username`	(Optional) Configures the remote username.
Step 4	`Router(config)# end`	(Optional) Exits global configuration mode. This step is required only if you want to change the default remote username (see Step 3).
Step 5	`Router# copy flash-url rcp:[[[//[username@]location]/directory]/filename]`	Copies the system image from Flash memory to a network server using rcp.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copy from Flash to RCP Server Example

The following example copies the system image named c5200-ds-l to the network server at 172.16.1.111 using rcp and a username of netadmin:

```
Router# copy flash:c5200-ds-l rcp:netadmin1@172.16.1.111/c5200-ds-l
Verifying checksum for 'c5200-ds-l' (file # 1)...[OK]
Writing c5200-ds-l -
```

Copy from Slot1 to RCP Server Example

The following example copies a system image file named test from the second PCMCIA slot to a network server using rcp. The remote username is netadmin1. Because the destination address and filename are not specified, the router prompts for this information.

```
Router# configure terminal
Router(config)# ip rcmd remote-username netadmin1
Router(config)# end
Router# copy slot1:test rcp:
Address or name of remote host [UNKNOWN]? 172.16.1.111
File name to write to? test
Verifying checksum for 'test' (file # 1)...[OK]
```

```
Writing test
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Upload to server done
Flash device copy took 00:00:08 [hh:mm:ss]
```

Copying an Image from Flash Memory to an FTP Server

You can copy a system image to an FTP network server.

Understanding the FTP Username and Password

The FTP protocol requires a client to send a remote username and password on each FTP request to a server. When you copy a configuration file from the router to a server using FTP, the Cisco IOS software sends the first valid username it encounters in the following list:

1. The username specified in the **copy** EXEC command, if a username is specified.
2. The username set by the **ip ftp username** global configuration command, if the command is configured.
3. Anonymous.

The router sends the first valid password it encounters in the following list:

1. The password specified in the **copy** EXEC command, if a password is specified.
2. The password set by the **ip ftp password** global configuration command, if the command is configured.
3. The router forms a password *username@routername.domain*. The variable *username* is the username associated with the current session, *routername* is the configured host name, and *domain* is the domain of the router.

The username and password must be associated with an account on the FTP server. If you are writing to the server, the FTP server must be properly configured to accept the FTP write request from the user on the router.

If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the username on the server. For example, if the system image resides in the home directory of a user on the server, specify that user's name as the remote username.

Refer to the documentation for your FTP server for more information.

Use the **ip ftp username** and **ip ftp password** commands to specify a username and password for all copies. Include the username in the **copy** command if you want to specify a username for that copy operation only.

Copying from Flash Memory to an FTP Server Tasks

To copy a system image to an FTP network server, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	(Optional) Enters global configuration mode. This step is required only if you override the default remote username or password (see Steps 2 and 3).
Step 2	`Router(config)# ip ftp username username`	(Optional) Changes the default remote username.
Step 3	`Router(config)# ip ftp password password`	(Optional) Changes the default password.
Step 4	`Router(config)# end`	(Optional) Exits global configuration mode. This step is required only if you override the default remote username or password (see Steps 2 and 3).
Step 5	`Router# show flash-filesystem:`	(Optional) Displays the system image file in the specified Flash directory. If you do not already know it, note the exact spelling of the system image filename in Flash memory.
Step 6	`Router# copy flash-filesystem:filename ftp:[[[//[username [:password]@]location]/directory]/filename]`	Copies the image to the FTP server.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copying from Flash Memory to an FTP Server Example

The following example uses the **show flash:** EXEC command to learn the name of the system image file and the **copy flash: tftp:** EXEC command to copy the system image (c3640-2is-mz) to a TFTP server. The router uses the default username and password.

```
Router# show flash:

System flash directory:
File  Length   Name/status
  1   4137888  c3640-c2is-mz
[4137952 bytes used, 12639264 available, 16777216 total]
16384K bytes of processor board System flash (Read/Write)\

Router# copy flash: tftp:
IP address of remote host [255.255.255.255]? 172.16.13.110
filename to write on tftp host? c3600-c2is-mz
writing c3640-c2is-mz !!!!...
successful ftp write.
```

Copying from Slot1 to an FTP Server Example

The following example uses the **show slot1:** EXEC command to display the name of the system image file in the second PCMCIA slot, and copies the file (test) to an FTP server.

```
Router# show slot1:

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

Router# copy slot1:test ftp://thisuser:thatpass@172.16.13.110/test
writing test!!!!...
successful ftp write.
```

Copying from Partitioned Flash to an FTP Server Example

In this example, the file named your-ios is copied from partition 1 of the Flash memory PC card in slot 0 to the TFTP server at 172.23.1.129. The file will be saved with the name your-ios in the dirt/sysadmin directory relative to the directory of the remote username.

```
Router# show slot0: partition 1

PCMCIA Slot0 flash directory, partition 1:
File  Length   Name/status
  1   1711088 your-ios
[1711152 bytes used, 2483152 available, 4194304 total]

Router# copy slot0:1:your-ios ftp://myuser:mypass@172.23.1.129/dirt/sysadmin/your-ios

Verifying checksum for 'your-ios' (file # 1)...  OK
Copy 'your-ios' from Flash to server
  as 'dirt/sysadmin/ios-2'? [yes/no] yes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Upload to server done
Flash device copy took 00:00:23 [hh:mm:ss]
```

Copying Images from a Network Server to Flash Memory

You can copy system images or boot image from a TFTP, rcp, or FTP server to a Flash memory file system to upgrade or change the Cisco IOS software or boot image on your router.

The protocol you use depends on which type of server you are using. The FTP and rcp transport mechanisms provide faster performance and more reliable delivery of data than TFTP. These improvements are possible because the FTP and rcp transport mechanisms are built on and use the TCP/IP stack, which is connection-oriented.

The following sections describe the copying tasks. The first two tasks and the last task are required. If you have a run-from-Flash system, the third section is required. Perform one of the remaining tasks, depending on which file transfer protocol you use.

- Restrictions on Naming Files
- Understanding Flash Memory Space Considerations
- Output for Image Downloading Process
- Copying to Flash Memory for Run-from-Flash Systems
- Copying an Image from a TFTP Server to a Flash Memory File System
- Copying an Image from an rcp Server to a Flash Memory File System
- Copying an Image from an FTP Server to a Flash Memory File System
- Verifying the Image in Flash Memory

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

Restrictions on Naming Files

Filenames in Flash memory can be up to 63 characters long; they are not case-sensitive and are always converted to lowercase.

Note The destination filename must be an alphanumeric expression (contains all letters or a combination of letters and numberals). For example, "1" is an invalid filename.

The filename can be in either lowercase or uppercase; the system ignores case. If more than one file of the same name is copied to Flash, regardless of case, the last file copied becomes the valid file.

Understanding Flash Memory Space Considerations

Be sure that enough space is available before copying a file to Flash memory. Use the **show** *flash-filesystem***:** EXEC command, and compare the size of the file you want to copy to the amount of Flash memory available. If the space available is less than the amount needed, the **copy** EXEC command will be partially executed, but the entire file will not be copied into Flash memory. The failure message "buffer overflow - *xxxx*/*xxxx*" will appear, where *xxxx*/*xxxx* is the number of bytes read from the source file and the number of bytes available on the destination device.

Caution Do not reboot the router if no valid image is in Flash memory.

Note For the Cisco 3600 series routers, if you do not have access to a network server and need to download a system image, you can copy an image from a local or remote computer (such as a PC, UNIX workstation, or Macintosh) using the Xmodem or Ymodem protocols. See the section "Recovering a System Image Using Xmodem or Ymodem" later in this chapter.

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

On Class B Flash file systems, the router gives you the option of erasing the existing contents of Flash memory before writing to it. If no free Flash memory is available, or if no files have ever been written to Flash memory, the erase routine is required before new files can be copied. If there is enough free Flash memory, the router gives you the option of erasing the existing Flash memory before writing to it. The system will inform you of these conditions and prompt you for a response.

Note If you enter **n** after the "Erase flash before writing?" prompt, the copy process continues. If you enter **y** and confirm the erasure, the erase routine begins. Be sure to have ample Flash memory space before entering **n** at the erasure prompt.

If you attempt to copy a file into Flash memory that is already there, a prompt informs you that a file with the same name already exists. This file is "deleted" when you copy the new file into Flash.

- On Class A and B Flash file systems, the first copy of the file still resides within Flash memory, but it is rendered unusable in favor of the newest version and is listed with the "deleted" tag when you use the **show** *flash-filesystem***:** EXEC command. If you terminate the copy process, the newer file is marked "deleted" because the entire file was not copied and is not valid. In this case, the original file in Flash memory is valid and available to the system.
- On Class C Flash file systems, the first copy of the file is erased.

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

On some platforms, the Flash security jumper must be installed in order to write to Flash memory. In addition, some platforms have a write protect switch which must be set to *unprotected* in order to write to Flash memory.

Output for Image Downloading Process

The output and dialog varies depending on the platform.

Output for Partitioned Flash Memory

One of the following prompts will be displayed after the command is entered to indicate how a file can be downloaded:

- None—The file cannot be copied.
- RXBOOT-Manual—You must manually reload to the rxboot image in ROM to copy the image.
- RXBOOT-FLH—The copy is done automatically via the Flash load helper software in boot ROMs.
- Direct—The copy can be done directly.

If the file can be downloaded into more than one partition, you are prompted for the partition number. To obtain help, enter any of the following characters at the partition number prompt:

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

Copying to Flash Memory for Run-from-Flash Systems

You cannot run the system from Flash memory and copy to it at the same time. Therefore, for systems that run from Flash, preform either of the following tasks before copying to Flash:

- Partition Flash memory or use Flash load helper to allow the system to run from Flash memory while you copy to it.
- Reload the system to use a system image from boot ROMs.

See the "Understanding Memory Types and Functions" section in the "Maintaining System Memory" chapter of this document for more information on run-from-Flash systems.

Refer to the appropriate hardware installation and maintenance publication for information about the jumper settings required for your configuration.

Copying an Image from a TFTP Server to a Flash Memory File System

Before you copy a system image or boot image to Flash memory, you should make a backup copy of the current software or bootstap image. See the “Copying Images from Flash Memory to a Network Server” section on page 176 for details.

To copy a system image from a TFTP server to a Flash memory file system, use the following command in EXEC mode:

Command	Purpose
Router# **copy tftp:** [[[**//***location*]**/***directory*]**/***filename*] *flash-filesystem***:**[*filename*]	Copies a system image or a boot image to Flash memory.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copying from a TFTP Server to Flash Memory Example

In the following example, a file is copied from a TFTP server to slot1:

```
Router# copy tftp://theserver/tftpboot/kristen/ken/c7200-js-mz slot1:
Destination filename [c7200-js-mz]?
Accessing tftp://theserver/tftpboot/kristen/ken/c7200-js-mz...Translating
"theserver"...domain server (192.168.2.132) [OK]

Loading tftpboot/kristen/ken/c7200-js-mz from 192.168.2.132 (via Ethernet3/0):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 4823492 bytes]

4823492 bytes copied in 264.312 secs (18270 bytes/sec)
```

The following example copies a system image named igs-p-l from a TFTP server to a Class B Flash file system when Flash memory is too full to copy the file:

```
Router# copy tftp: flash:
IP address or name of remote host [255.255.255.255]? dirt
Translating “DIRT”...domain server (255.255.255.255) [OK]

Name of file to copy? igs-p-l
Copy igs-p-l from 172.16.13.111 into flash memory? [confirm]
Flash is filled to capacity.
Erasure is needed before flash may be written.
Erase flash before writing? [confirm]
Erasing flash EPROMs bank 0

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

```
Erasing flash EPROMs bank 1

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

Erasing flash EPROMs bank 2

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

Erasing flash EPROMs bank 3

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

Loading from 172.16.1.111:!!!!...
 [OK - 1906676 bytes]
Verifying via checksum...
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvv
Flash verification successful. Length = 1906676, checksum = 0x12AD
```

Copying from a TFTP Server to Flash Example When File by the Same Name Already Exists Example

The following example shows how to copy a system image named igs-p-l into the current Flash configuration in which a file named igs-p-l already exists:

```
Router# copy tftp://172.16.13.111/igs-p-l flash:igs-p-l
File igs-p-l already exists; it will be invalidated!
Copy igs-p-l from 172.16.13.111 into flash memory? [confirm]
2287500 bytes available for writing without erasure.
Erase flash before writing? [confirm]n
Loading from 172.16.1.111:!!!!...
[OK - 1906676 bytes]
Verifying via checksum...
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvv
Flash verification successful. Length = 1902192, checksum = 0x12AD
```

Copying from TFTP Server to Flash Example Without Security Jumper Installed Example

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

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

Copying from TFTP Server to Partitioned Flash Example

In the following example, the file named c3600-i-mz on the TFTP server at 172.23.1.129 is copied to the first partition of internal Flash Memory:

```
Router# copy tftp://172.23.1.129/c3600-i-mz flash:1:c3600-i-mz/c3600-i-mz
Accessing file 'c3600-i-mz' on 172.23.1.129...
Loading c3600-i-mz from 172.23.1.129 (via Ethernet1/0): ! [OK]
Erase flash device before writing? [confirm]
Flash contains files. Are you sure you want to erase? [confirm]
Copy 'c3600-i-mz' from server
  as 'c3600-i-mz' into Flash WITH erase? [yes/no] yes
Erasing device... eeeeeeeeeeeeeeee ...erased
Loading c3600-i-mz from 172.23.1.129 (via Ethernet1/0):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 1711088 bytes]

Verifying checksum...  OK (0xF89A)
Flash device copy took 00:00:17 [hh:mm:ss]
```

Copying an Image from an rcp Server to a Flash Memory File System

You can copy a system image from an rcp network server to a Flash memory file system.

If you copy the configuration file to a PC used as a file server, the computer must support rsh.

Understanding the rcp Username

The rcp protocol requires a client to send a remote username on each rcp request to a server. When you copy an image from the router to a server using rcp, the Cisco IOS software sends the first valid username it encounters in the following list:

1. The remote username specified in the **copy** EXEC command, if one if specified.
2. The username set by the **ip rcmd remote-username** global configuration command, if the command is configured.
3. The remote username associated with the current tty (terminal) process. For example, if the user is connected to the router through Telnet and was authenticated through the **username** global configuration command, the router software sends the Telnet username as the remote username.
4. The router host name.

For the rcp copy request to execute, an account must be defined on the network server for the remote username. If the server has a directory structure, the configuration file or image is written or copied relative to the directory associated with the remote username on the server. The path for all files and images to be copied begins at the remote user's home directory. For example, if the system image resides in the home directory of a user on the server, specify that user's name as the remote username.

Copying from an rcp Server to Flash Memory

To copy an image from an rcp server to Flash memory, use the following command, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	See the instructions in the section "Copying Images from Flash Memory to a Network Server."	Make a backup copy of the current system or bootstrap software image.
Step 2	Router# **configure terminal**	(Optional) Enters global configuration mode from the terminal. This step is required only if you override the default remote username (see Step 3).
Step 3	Router(config)# **ip rcmd remote-username** *username*	(Optional) Specifies the remote username.
Step 4	Router# **end**	(Optional) Exits global configuration mode. This step is required only if you override the default remote username (see Step 3).
Step 5	Router# **copy rcp:** [[[**//**[*username***@**]*location*]**/***directory*] **/***filename*] *flash-filesystem***:**[*filename*]	Copies the image from an rcp server to a Flash memory file system.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copying from an rcp Server to Flash Example

The following example copies a system image named mysysim1 from the netadmin1 directory on the remote server named SERVER1.CISCO.COM with an IP address of 172.16.101.101 to Flash memory. To ensure that enough Flash memory is available to accommodate the system image to be copied, the Cisco IOS software allows you to first erase the contents of Flash memory.

```
Router1# configure terminal
Router1(config)# ip rcmd remote-username netadmin1
Router1(config)# end
Router# copy rcp: flash:

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

Address or name of remote host[UNKNOWN]? 172.16.101.101
Name of file to copy? mysysim1
Copy mysysim1 from SERVER1.CISCO.COM?[confirm]

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

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

Connected to 172.16.101.101

Loading 2076007 byte file mysysim1:!!!!!...
[OK]

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

Copying from an rcp Server to Partitioned Slot0 Example

In the following example, the file named c3600-i-mz on the rcp server at the IP address 172.23.1.129 is copied to partition 3 in slot 0. Because no username is specified, the router uses the default rcp remote username.

```
Router# show slot0: partition 3

PCMCIA Slot0 flash directory, partition 3:
File  Length   Name/status
  1   426      running-config
[492 bytes used, 4193812 available, 4194304 total]

Router# copy rcp://172.23.1.129/tftpboot/gate/c3600-i-mz slot0:3:/tftpboot/gate/c3600-i-mz
Accessing file '/tftpboot/gate/c3600-i-mz' on 172.23.1.129...
Connected to 172.23.1.129
Loading 1711088 byte file c3600-i-mz: ! [OK]

Erase flash device before writing? [confirm]
Flash contains files. Are you sure you want to erase? [confirm]
Copy '/tftpboot/gate/c3600-i-mz' from server
  as '/tftpboot/gate/c3600-i-mz' into Flash WITH erase? [yes/no] yes
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
Connected to 172.23.1.129
Loading 1711088 byte file c3600-i-mz:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! [OK]

Verifying checksum...  OK (0xF89A)
Flash device copy took 00:00:16 [hh:mm:ss]
```

Copying an Image from an FTP Server to a Flash Memory File System

You can copy a system image from an FTP server to a Flash memory file system.

Understanding the FTP Username and Password

The FTP protocol requires a client to send a remote username and password on each FTP request to a server. When you copy a configuration file from the router to a server using FTP, the Cisco IOS software sends the first valid username it encounters in the following list:

1. The username specified in the **copy** EXEC command, if a username is specified.
2. The username set by the **ip ftp username** global configuration command, if the command is configured.
3. Anonymous.

The router sends the first valid password it encounters in the following list:

1. The password specified in the **copy** EXEC command, if a password is specified.
2. The password set by the **ip ftp password** command, if the command is configured.
3. The router forms a password *username@routername.domain*. The variable *username* is the username associated with the current session, *routername* is the configured host name, and *domain* is the domain of the router.

The username and password must be associated with an account on the FTP server. If you are writing to the server, the FTP server must be properly configured to accept the FTP write request from the user on the router.

If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the username on the server. For example, if the system image resides in the home directory of a user on the server, specify that user's name as the remote username.

Refer to the documentation for your FTP server for more information.

Use the **ip ftp username** and **ip ftp password** commands to specify a username and password for all copies. Include the username in the **copy** command if you want to specify a username for that copy operation only.

Copying from an FTP Server to Flash Memory

To copy a system image from an FTP server to a Flash memory file system, use the following command, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`See the instructions in the section "Copying Images from Flash Memory to a Network Server."`	Make a backup copy of the current software image or bootstrap image.
Step 2	`Router# configure terminal`	(Optional) Enters global configuration mode from the terminal. This step is required only if you want to override the default remote username or password (see Steps 3 and 4).
Step 3	`Router(config)# ip ftp username username`	(Optional) Changes the default remote username.
Step 4	`Router(config)# ip ftp password password`	(Optional) Changes the default password.
Step 5	`Router(config)# end`	(Optional) Exits global configuration mode. This step is required only if you override the default remote username or password (see Steps 3 and 4).
Step 6	`Router# copy ftp: [[[//[username[:password]@]location] /directory]/filename] flash-filesystem:[filename]`	Copies the configuration file from a network server to running memory or the startup configuration using rcp.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copy from FTP Server to Flash Memory Example

The following example copies a the file named c7200-js-mz from the FTP server the server using a username of myuser and a password of mypass:

```
Router# copy ftp://myuser:mypass@theserver/tftpboot/ken/c7200-js-mz slot1:c7200-js-mz
Accessing ftp://theserver/tftpboot/ken/c7200-js-mz...Translating "theserver"...domain
server (192.168.2.132) [OK]

Loading c7200-js-mz from 192.168.2.132 (via Ethernet3/0):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 4823492 bytes]

4823492 bytes copied in 264.312 secs (18270 bytes/sec)
```

Verifying the Image in Flash Memory

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

Caution If the checksum value does not match the value in the README file, do not reboot the router. Instead, issue the **copy** command and compare the checksums again. If the checksum repeatedly is incorrect, copy the original system software image back into Flash memory *before* you reboot the router from Flash memory. If you have a corrupted image in Flash memory and try to boot from Flash, the router will start the system image contained in ROM (assuming that booting from a network server is not configured). If ROM does not contain a fully functional system image, the router will not function and must be reconfigured through a direct console port connection.

The Flash memory content listing does not include the checksum of individual files. To recompute and verify the image checksum after an image is copied into Flash memory or a Flash memory device, use the following command in EXEC mode:

Command	Purpose
`Router#` **`verify`** *`flash-filesystem`***`:`**`[`*`partition-number`***`:`**`]` `[`*`filename`*`]`	Recomputes and verifies the image checksum after the image is copied into Flash memory.

If you do not provide the filename in the command, the router prompts you. By default, it prompts for the last (most recent) file in Flash. Press Return to recompute the default file checksum, or enter the filename of a different file at the prompt. Note that the checksum for microcode images is always 0x0000.

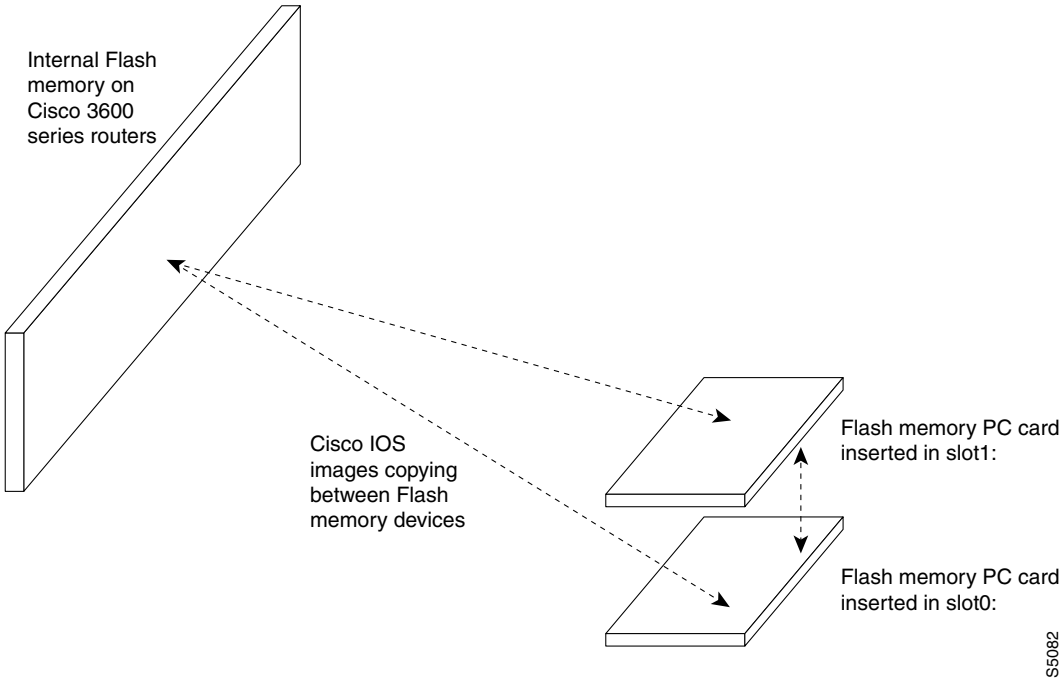
The following example verifies the image named c7200-js-mz in slot0:

```
Router# verify slot0:c7200-js-mz
Verified slot0:c7200-js-mz
```

Copying Images Between Local Flash Memory Devices

On routers with multiple Flash memory file systems, you can copy images from one Flash memory file system, such as internal Flash memory or a Flash memory card in a PCMCIA slot, to another Flash memory file system, as shown in Figure 9. One reason to copy the image to a different Flash device is to make a backup copy of it.

Figure 9 Copying Images Between Flash Memory File Systems



Caution Before copying to a new Flash device, you must first format that device.
All new media should be formatted. Memory media used in Cisco devices does not typically come pre-formatted. Even if pre-formatted, an inital format using the Cisco filesystem may help to prevent potential problems with incompatible formatting.
Attempts to copy images to unformatted or improperly formatted Flash devices may not generate failure messages on some devices. For this reason, the **show** and **verify** steps below are strongly recommended.
For instructions on formatting your flash device, see the "Maintaining Router Memory" chapter.

To copy an image between Flash memory file systems, use the following commands in EXEC mode:

	Command	Purpose
Step 1	`Router# show flash-filesystem:`	Displays the layout and contents of Flash memory.
Step 2	`Router# copy source-url destination-url`	Copies an image between Flash memory devices.
Step 3	`Router# verify flash-filesystem:filename`	Verifies the checksum of the image you copied. (You can get the MD5 checksum for your image from Cisco.com).

Note The source device and the destination device cannot be the same. For example, the **copy slot1: slot1:** command is invalid.

Copying a File Between Local Flash Memory Devices Example

The following example copies the file named new-ios from partition 1 of internal Flash memory to slot 0:

```
Router# show flash: partition 1

System flash directory, partition 1:
File  Length   Name/status
  1   3142748 admin/images/new-ios
[3142812 bytes used, 1051492 available, 4194304 total]

Router# show slot0:

PCMCIA Slot0 flash directory
File  Length   Name/status
  1   1711088  /tftpboot/gate/c3600-i-mz
[1711152 bytes used, 2483152 available, 4194304 total]

Router# copy flash:1:admin/images/new-ios slot0:admin/images/new-ios
Verifying checksum for 'admin/images/new-ios' (file # 1)...  OK

Erase flash device before writing? [confirm]
Flash contains files. Are you sure you want to erase? [confirm]
Copy 'admin/images/new-ios' from flash: device
  as 'admin/images/new-ios' into slot0: device WITH erase? [yes/no] yes
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 [OK - 3142748 bytes]

Flash device copy took 00:00:50 [hh:mm:ss]
Verifying checksum...  OK (0xB732)

Router# show slot0:

PCMCIA Slot0 flash directory
File  Length   Name/status
  1   3142748 admin/images/new-ios
[3142812 bytes used, 1051492 available, 4194304 total]

Router# verify slot0:
Verify filename []? new-ios
! long pause ...
Verifying file integrity of slot0:new-ios.....................!
Embedded Hash   MD5 : E1A04D4DE1ED00407E6E560B315DA505
Computed Hash   MD5 : E1A04D4DE1ED00407E6E560B315DA505
CCO Hash        MD5 : C03EC4564F86F9A24201C88A9DA67317

Signature Verified
Verified slot0:

Router#
```

Specifying the Startup System Image in the Configuration File

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

- From Flash memory—Flash memory allows you to copy new system images without changing ROM. Information stored in Flash memory is not vulnerable to network failures that might occur when loading system images from servers.
- From a network server—In case Flash memory becomes corrupted, you can specify that a system image be loaded from a network server using Maintenance Operation Protocol (MOP), TFTP, rcp, or FTP as a backup boot method. For some platforms, you can specify a boot image to be loaded from a network server using TFTP, rcp, or FTP.
- From ROM—In case of both Flash memory corruption and network failure, specifying a system image to be loaded from ROM provides a final backup boot method. System images stored in ROM may not always be as current as those stored in Flash memory or on network servers.

Note Some platforms cannot boot from ROM.

You can enter the different types of boot commands in any order in the startup configuration file or in the BOOT environment variable. If you enter multiple boot commands, the Cisco IOS software tries them in the order they are entered.

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

Loading the System Image from Flash Memory

Use the tasks described in the following sections to configure your router to boot from Flash memory. Flash memory can reduce the effects of network failure by reducing dependency on files that can only be accessed over the network.

Flash Memory Configuration Process

To configure the router to load a system image in Flash memory, perform the following steps:

	Task
Step 1	(Optional) Copy a system image or boot image to Flash memory using TFTP, rcp, and FTP. See the "Copying Images from a Network Server to Flash Memory" section for more information on performing this step.
Step 2	Configure the system to automatically boot from the desired file and location in Flash memory or boot flash memory. See the "Configuring the Router to Automatically Boot from an Image in Flash Memory" section.
Step 3	(Optional) Depending on the current configuration register setting, change the configuration register value. See the "Configuring the Router to Automatically Boot from an Image in Flash Memory" section for more information on modifying the configuration register.
Step 4	(Optional) For some platforms, set the BOOTLDR environment variable to change the location of the boot image.

	Task
Step 5	Save your configuration.
Step 6	Power-cycle and reboot your system to ensure that all is working as expected.

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

To configure a router to automatically boot from an image in Flash memory, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode from the terminal.
Step 2	`Router(config)# boot system flash` [*flash-filesystem:*] [*partition-number:*] *filename*	Specifies the filename of an image stored in Flash memory that should be used for booting.
Step 3	`Router(config)# config-register` *value*	Sets the configuration register to enable loading of the system image specified in the configuration file.
Step 4	`Router(config)# end`	Ends your configuration session and exits global configuration mode.
Step 5	`Router# copy system:running-config nvram:startup-config`	Saves the system running configurattion as the device startup configuration (startup-config file).
Step 6	`Router# more nvram:startup-config`	(Optional) Allows verification of the contents of the startup configuration.
Step 7	`Router# reload`	Reboots the system.

For routers that are partitioned, if you do not specify a partition, the router boots from the first partition. If you do not specify a filename, the router boots from the first valid image found in the partition.

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

To remove a filename from the configuration file, enter the **no boot system flash** global configuration command and specify the file location.

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

Configuring the Router to Boot from Flash Memory Example

The following example shows a router configured to automatically boot from an image in Flash memory:

```
Router# configure terminal
Router(config)# boot system flash new-image
Router(config)# config-register 0x010F
Router(config)# end
Router# copy system:running-config nvram:startup-config
[ok]
Router# reload
[confirm]
```

```
%SYS-5-RELOAD: Reload requested
System Bootstrap, Version 12.0(7)W5(15) RELEASE SOFTWARE
     Copyright (c) 1986-2001 by Cisco Systems, Inc.
RP1 processor with 16384 Kbytes of memory
F3: 1871404+45476+167028 at 0x1000

Booting new-image from flash memory RRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR [OK - 1916912 bytes]
F3: 1871404+45476+167028 at 0x1000

              Restricted Rights Legend
.
.
.
```

Loading the System Image from a Network Server

You can configure the Cisco IOS software to load a system image file from a network server using FTP, TFTP, rcp, or MOP.

If you do not boot from a network server using MOP and you do not specify either FTP, TFTP, or rcp, by default the system image that you specify is booted from a network server via TFTP.

Note If you are using a Sun workstation as a network server and TFTP to transfer the file, configure the workstation to enable verification and generation of User Datagram Protocol (UDP) checksums. See Sun documentation for details.

For increased performance and reliability, use rcp to boot a system image from a network server. The rcp implementation uses the TCP, which ensures reliable delivery of data.

You cannot explicitly specify a remote username when you issue the **boot** ROM monitor command. Instead, the host name of the router is used. If the remote server has a directory structure, as do UNIX systems, and you boot the router from a network server using rcp, the Cisco IOS software searches for the system image on the server relative to the directory of the remote username.

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

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

To specify the loading of a system image from a network server, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.

Step 2	`Router(config)# boot system [rcp | tftp] filename [ip-address]`
	or
	`Router(config)# boot system mop filename [mac-address] [interface]`
		Specifies the system image file to be booted from a network server using rcp, TFTP, or MOP.

Step 3	`Router(config)# config-register value`	Sets the configuration register to enable loading of the image specified in the configuration file.
Step 4	`Router(config)# exit`	Exits configuration mode.

Step 5	`Router# copy system:running-config nvram:startup-config`
	or
	`Router# copy run start`
		Saves the configuration file to your startup configuration.

In the following example, a router uses rcp to boot from the testme5.tester system image file on a network server at IP address 172.16.0.1:

```
Router# configure terminal
Router(config)# boot system rcp testme5.tester 172.16.0.1
Router(config)# config-register 0x010F
Router(config)# ^Z
Router# copy system:running-config nvram:startup-config
```

The following section describes how to change request retry times and frequency if you have configured your system to boot using the **boot system mop** command.

Changing MOP Request Parameters

If you configure your router to boot from a network server using MOP (using the **boot system mop** ROM monitor command), the router will send a request for the configuration file to the MOP boot server during startup. By default, when the software sends a request that requires a response from a MOP boot server and the server does not respond, the message will be re-sent after 4 seconds. The message will be re-sent a maximum of eight times. The MOP device code is set to the Cisco device code by default.

If the MOP boot server and router are separated by a slow serial link, it may take longer than 4 seconds for the router to receive a response to its message. Therefore, you may want to configure the software to wait longer than 4 seconds before resending the message if you are using such a link. You may also want to change the maximum number of retries for the MOP request or the MOP device code.

To change the Cisco IOS software parameters for sending boot requests to a MOP server, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode from the terminal.
Step 2	`Router(config)# mop device-code {cisco | ds200} mop retransmit-timer seconds mop retries count`	Changes MOP server parameters.
Step 3	`Router(config)# end`	Exits configuration mode.
Step 4	`Router# copy running-config startup-config`	Saves the configuration file to your startup configuration.

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

```
Router# configure terminal
Router (config)# mop retransmit-timer 10
Router (config)# end
Router# copy running-config startup-config
```

Loading the System Image from ROM

To specify the use of the ROM system image as a backup to other boot instructions in the configuration file, use the following commands beginning in EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# boot system rom`	Specifies use of the ROM system image as a backup image.
Step 3	`Router(config)# config-register value`	Sets the configuration register to enable loading of the system image specified in the configuration file.
Step 4	`Router(config)# end`	Exits global configuration mode.
Step 5	`Router# copy system:running-config nvram:startup-config`	Saves the configuration file to your startup configuration.

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

```
Router# configure terminal
Router(config)# boot system rom
Router(config)# config-register 0x010F
Router(config)# end
Router# copy system:running-config nvram:startup-config
```

Note The Cisco 7000 series routers cannot load from ROM.

Using a Fault-Tolerant Booting Strategy

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

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

This boot order provides the most fault-tolerant booting strategy. Use the following commands beginning in EXEC mode to allow the router to boot first from Flash, then from a system file from a network server, and finally from ROM:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# boot system flash [flash-filesystem:][partition-number:] filename`	Configures the router to boot from Flash memory.
Step 3	`Router(config)# boot system [rcp | tftp] filename [ip-address]`	Configures the router to boot from a network server.
Step 4	`Router(config)# boot system rom`	Configures the router to boot from ROM.
Step 5	`Router(config)# config-register value`	Sets the configuration register to enable loading of the system image specified in the configuration file.
Step 6	`Router(config)# end`	Exits global configuration mode.
Step 7	`Router# copy system:running-config nvram:startup-config`	Saves the configuration file to your startup configuration.

In the following example, a router is configured to first boot an internal Flash image named *gsxx*. Should that image fail, the router will boot the configuration file *gsxx* from a network server. If that method should fail, then the system will boot from ROM.

```
Router# configure terminal
Router(config)# boot system flash gsxx
Router(config)# boot system gsxx 172.16.101.101
Router(config)# boot system rom
Router(config)# config-register 0x010F
Router(config)# end
Router# copy system:running-config nvram:startup-config
[ok]
```

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

Recovering a System Image Using Xmodem or Ymodem

If you do not have access to a network server and need to download a system image (to update it, or if all the system images in Flash memory somehow are damaged or erased), you can copy an image from a local or remote computer (such as a PC, UNIX workstation, or Macintosh) using the Xmodem or Ymodem protocols. This functionality primarily serves as a disaster recovery technique and is illustrated in Figure 10.

Note Recovering system images using Xmodem or Ymodem is performed only on the Cisco 1600 series and Cisco 3600 series routers.

Xmodem and Ymodem are common protocols used for transferring files and are included in applications such as Windows 3.1 (TERMINAL.EXE), Windows 95 (HyperTerminal), Windows NT 3.5x (TERMINAL.EXE), Windows NT 4.0 (HyperTerminal), and Linux UNIX freeware (minicom).

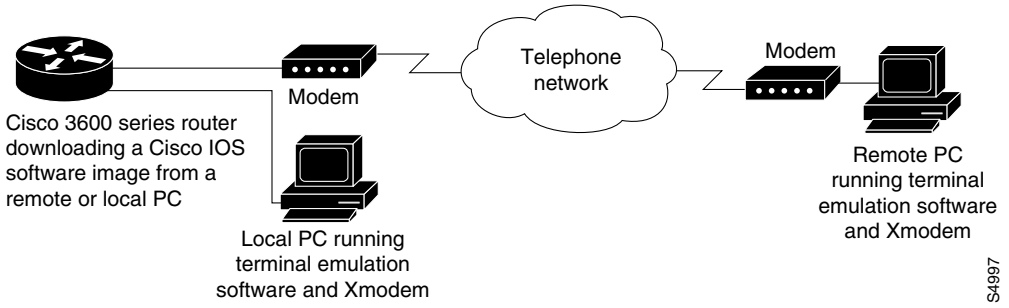
Cisco 3600 series routers do not support XBOOT functionality, a disaster recovery technique for Cisco IOS software, and do not have a separate boot helper (rxboot) image.

Xmodem and Ymodem downloads are slow, so you should use them only when you do not have access to a network server. You can speed up the transfer by setting the transfer port speed to 115200 bps.

On the Cisco 3600 series routers, you can perform the file transfer using Cisco IOS software or, if all local system images are damaged or erased, the ROM monitor. When you use Cisco IOS software for an Xmodem or Ymodem file transfer, the transfer can occur on either the AUX port or the console port. We recommend the AUX port, which supports hardware flow control. File transfers from the ROM monitor must use the console port.

On the Cisco 1600 series routers, you can only perform the file transfer from the ROM monitor over the console port.

Figure 10 Copying a System Image to a Cisco 3600 Series Router with Xmodem or Ymodem



To copy a Cisco IOS image from a computer or workstation to a router using the Xmodem or Ymodem protocol, use the following commands, as needed:

Command	Purpose
Router# **copy xmodem:** *flash-filesystem*:[*partition*:][*filename*] or Router# **copy ymodem:** *flash-filesystem*:[*partition*:][*filename*]	Copies a system image from a computer to Flash memory using Cisco IOS software in EXEC mode (Cisco 3600 series routers only).
ROMMON > **xmodem** [**-c**] [**-y**] [**-e**] [**-f**] [**-r**] [**-x**] [**-s** *data-rate*] [*filename*]	Copies a system image from a computer to Flash memory in ROM monitor mode for the Cisco 1600 series routers. The **-c** option provides CRC-16 checksumming; **-y** uses the Ymodem protocol; **-e** erases the first partition in Flash memory; **-f** erases all of Flash memory; **-r** downloads the image to DRAM (the default is Flash memory); **-x** prevents the image from executing after download; and **-s** sets the console port data rate.
ROMMON > **xmodem** [**-c** | **-y** | **-r** | **-x**] [*filename*]	Copies a system image from a computer to Flash memory in ROM monitor mode for the Cisco 3600 series routers.

The computer from which you transfer the Cisco IOS image must be running terminal emulation software and the Xmodem or Ymodem protocol.

For the Cisco 1600 series routers, if you include the **-r** option (download to DRAM), your router must have enough DRAM to hold the file being transferred. To run from Flash memory, an image must be positioned as the first file in Flash memory. If you are copying a new image to boot from Flash memory, erase all existing files first.

Xmodem Transfer Using the Cisco IOS Software Example

The following example shows a file transfer using Cisco IOS software and the Xmodem protocol. The Ymodem protocol follows a similar procedure, using the **copy ymodem:** EXEC command.

Note This functionality is enabled on Cisco 3600 series routers only.

To transfer a Cisco IOS image from a computer running terminal emulation software and the Xmodem protocol, perform the following steps:

Step 1 Place a Cisco IOS software image on the remote computer's hard drive. You can download an image from Cisco.com.

Step 2 To transfer from a remote computer, connect a modem to the AUX port of your Cisco 3600 series router and to the standard telephone network. The AUX port is set by default to a speed of 9600 bps, 2 stop bits, and no parity. The maximum speed is 115200 bps. Configure the router for both incoming and outgoing calls by entering the **modem inout** line configuration command.

Connect a modem to the remote computer and to the telephone network. The remote computer dials through the telephone network and connects to the router.

To transfer from a local computer, connect the router's AUX port to a serial port on the computer, using a null-modem cable. The AUX speed configured on the router must match the transfer speed configured on the local computer.

Step 3 At the EXEC prompt in the terminal emulator window of the computer, enter the **copy xmodem: flash:** EXEC command:

```
Router# copy xmodem: flash:
                          **** WARNING ****
x/ymodem is a slow transfer protocol limited to the current speed
settings of the auxiliary/console ports. The use of the auxiliary
port for this download is strongly recommended.
During the course of the download no exec input/output will be
available.
                          ---- ******* ----
```

Press **Enter** to continue.

Step 4 Specify whether to use cyclic redundancy check (CRC) block checksumming, which verifies that your data has been correctly transferred from the computer to the router. If your computer does not support CRC block checksumming, enter **no** at the prompt:

```
Proceed? [confirm]
Use crc block checksumming? [confirm] no
```

Step 5 Determine how many times the software should try to receive a bad block of data before it declares the copy operation a failure. The default is ten retries. A higher number may be needed for noisy telephone lines. You can configure an unlimited number of retries.

```
Max Retry Count [10]: 7
```

Step 6 Decide whether you want to check that the file is a valid Cisco 3600 series image:

```
Perform image validation checks? [confirm]
Xmodem download using simple checksumming with image validation
Continue? [confirm]
```

After the transfer has begun, and if the image is valid, the software determines whether enough Flash memory space exists on the router to accommodate the transfer:

```
System flash directory:
File  Length   Name/status
  1   1738244  images/c3600-i-mz
[1738308 bytes used, 2455996 available, 4194304 total]
```

Step 7 Enter the destination filename:

```
Destination file name ? new-ios-image
```

Step 8 If you do not want the contents of internal Flash memory erased before the file transfer, enter **no**:

```
Erase flash device before writing? [confirm] no

Copy '' from server
  as 'new-ios-image' into Flash WITHOUT erase? [yes/no] yes
Ready to receive file...........
```

Step 9 Start an Xmodem or Ymodem send operation with the terminal emulation software on the computer that is sending the system image to the router. See your emulation software application's documentation for instructions on how to execute a file transfer. Depending on the application you use, the emulation software may display the progress of the file transfer.

Xmodem Transfer Example Using the ROM Monitor

This example shows a file transfer using the ROM monitor and the Xmodem protocol. To send with the Ymodem protocol, use the **xmodem -y** ROM monitor command.

For the Cisco 3600 series routers, the router must have enough DRAM to hold the file being transferred, even if you are copying to Flash memory. The image is copied to the first file in internal Flash memory. Any existing files in Flash memory are erased. Copying files to Flash partitions or to the second-file position is not supported.

Caution A modem connection from the telephone network to your console port introduces security issues that you should consider before enabling the connection. For example, remote users can dial in to your modem and access the router's configuration settings.

Step 1 Place a Cisco IOS software image on the remote computer's hard drive. You can download an image from Cisco.com or from the Feature Pack (Cisco 1600 series routers only).

Step 2 To transfer from a remote computer, connect a modem to the console port of your router and to the standard telephone network. The modem and console port must communicate at the same speed, which can be from 9600 to 115200 bps (Cisco 3600 series routers) or from 1200 to 115200 bps (Cisco 1600 series routers), depending on the speed supported by your modem. Use the **confreg** ROM monitor command to configure the console port transmission speed for the router. For the Cisco 1600 series routers, you can also set the transmission speed with the **-s** option.

Connect a modem to the remote computer and to the telephone network. The remote computer dials through the telephone network and connects to the router.

To transfer from a local computer, connect the router's console port to a serial port on the computer, using a null-modem cable. The console port speed configured on the router must match the transfer speed configured on the local computer.

Note If you are transferring from a local computer, you may need to configure the terminal emulation program to ignore Request To Send (RTS)/data terminal ready (DTR) signals.

Step 3 You should see a ROM monitor prompt in the terminal emulation window:

```
rommon >
```

Enter the **xmodem** ROM monitor command, along with any desired copy options and, optionally, the filename of the Cisco IOS image. The image loads into Flash memory by default; to download to DRAM instead, use the **-r** option. The image is normally executed on completion of the file transfer; to prevent execution, use the **-x** option. The **-c** option specifies CRC-16 checksumming, which is more sophisticated and thorough than standard checksumming, if it is supported by the computer:

```
rommon > xmodem -c new-ios-image
Do not start the sending program yet...
         File size           Checksum   File name
   1738244 bytes (0x1a8604)   0xdd25 george-admin/c3600-i-mz


WARNING: All existing data in flash will be lost!
Invoke this application only for disaster recovery.
Do you wish to continue? y/n  [n]:  yes
Ready to receive file new-ios-image ...
```

Step 4 Start an Xmodem send operation, which is initiated from the terminal emulation software on the remote computer that is sending the system image to the router. See your emulation software application's documentation for instructions on how to execute an Xmodem file transfer.

Step 5 The Cisco IOS image is transferred and executed. If you are transferring from a remote computer, the computer maintains control of your console port even after the new Cisco IOS image is running. To release control to a local terminal, reconfigure the speed of the router's console port to match the speed of the local terminal by entering the **speed** *bps* line configuration command from the remote computer at the router prompt:

```
Router# configure terminal
Router(config)# line 0
Router(config-line)# speed 9600
```

The remote connection is broken, and you can disconnect the modem from the console port and reconnect the terminal line.

Loading and Displaying Microcode Images

On some Cisco routers, including Cisco 7200, 7500, and 12000 series GSRs, you can update microcode by loading it into peripheral components. This section provides information on loading, upgrading and verifying microcode images, as described in the following subsections:

- Understanding Microcode Images

- Specifying the Location of the Microcode Images
- Reloading the Microcode Image
- Displaying Microcode Image Information

Understanding Microcode Images

Microcode is stored on ROM and allows the addition of new machine instructions without requiring that they be designed into electronic circuits when new instructions are needed. Microcode images contain microcode software that runs on various hardware devices. For example, microcode can be updated in Channel Interface Processors (CIPs) on Cisco7500 series routers, or in Channel Port Adapters (CPAs) on Cisco 7200 series routers.

By default, the system loads the microcode bundled with the Cisco IOS system software image. This microcode is referred to as the default microcode image. However, you can configure the router to use microcode stored in Flash.

Cisco 7000 series routers with an RSP7000 and Cisco 7500 series routers each have a writable control store (WCS) that stores microcode. You can load updated microcode onto the WCS from boot flash or from a Flash memory card inserted in one of the PCMCIA slots of the RSP card.

You can update microcode without having physical access to the router by using the **copy** EXEC command to copy microcode to a Flash file system.

Specifying the Location of the Microcode Images

To specify the location from where the microcode should be loaded, use the following commands beginning in EXEC mode:

	Command	Purpose

Step 1	`Router# copy tftp: flash:`
	or
	`Router# copy tftp: file-id`
		(Optional) Copies microcode files into Flash. Perform this step only if you want to load the microcode from Flash.
		See the section "Copying Images from a Network Server to Flash Memory" for more information about how to copy images to Flash memory.

Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# microcode interface [flash-filesystem:filename [slot] | system [slot]]`	Configures the router to load microcode on a target interface from the specified memory location.
Step 4	`Router(config)# end`	Exits global configuration mode.
Step 5	`Router# copy system:running-config nvram:startup-config`	Saves the new configuration information.

If an error occurs when you are attempting to download microcode, the system loads the default system microcode image.

Note Microcode images cannot be compressed.

Reloading the Microcode Image

The configuration commands specifying the microcode to load are implemented following one of three events:

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

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

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

Command	Purpose
`Router(config)# microcode reload`	Reloads the microcode from the source specified in the configuration on to all interface and processor cards.

Immediately after you enter the **microcode reload** global configuration command and press Return, the system reloads all microcode. Global configuration mode remains enabled. After the reload is complete, enter the **exit** global configuration command to return to the EXEC prompt.

If Flash memory is busy because a card is being removed or inserted, or a **microcode reload** command is executed while Flash is locked, the files will not be available and the onboard ROM microcode will be loaded. Issue another **microcode reload** command when Flash memory is available, and the proper microcode will be loaded. The **show flash** EXEC command will reveal if another user or process has locked Flash memory.

Note The **microcode reload** command should not be used while Flash is in use. For example, do not use this command when a **copy** {**ftp:** | **rcp:** | **tftp:**} *flash-filesystem* or **show** *flash-filesystem***:** EXEC command is active.

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

In the following example, all controllers are reset, the specified microcode is loaded, and the CxBus complex is reinitialized according to the microcode configuration commands that have been written to memory:

```
Router# configure terminal
Router(config)# microcode reload
Router(config)# end
```

Displaying Microcode Image Information

To display microcode image information, use the following command in EXEC mode:

Command	Purpose
`Router# show microcode`	Displays microcode information.

Using Microcode on Specific Platforms

The commands for manipulating microcode vary by platform. This section refers you to specialized configuration information found in other Cisco IOS documents.

For information on downloading microcode (Modem Firmware and Portware) into modems on Cisco access servers (like the Cisco AS5800) using SPE, see the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

For specific information on loading CIP and CPA microcode into adapters on Cisco 7000, 7200, and 7500 series routers, see the "Configuring Cisco Mainframe Channel Connection Adapters" chapter in the "IBM Networking" part of the *Cisco IOS Bridging and IBM Networking Configuration Guide*.

Loading Microcode Images on the Cisco 12000 GSR

In addition to the Cisco IOS image that resides on the GRP, each line card on the Cisco 12000 series has a Cisco IOS image. When the router is reloaded, the specified Cisco IOS image is loaded onto the GRP, and that image is automatically downloaded to all the line cards.

Normally, you want the same Cisco IOS image on the GRP and all line cards. However, if you want to upgrade a line card with a new version of microcode for testing or to fix a defect, you may need to load a microcode system image that is different from the one on the line card. You may also need to load a new image on the line card to work around a problem that is affecting only one of the line cards.

To load a Cisco IOS image on a line card, first use the **copy tftp** EXEC command to download the Cisco IOS image to a slot on one of the PCMCIA Flash cards. After you have downloaded the Cisco IOS image on the Flash card, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# microcode {oc12-atm | oc12-pos | oc3-pos-4} flash file_id slot-number`	Specifies the type of line card, location of the microcode image, and the slot of the line card to download the image. If the slot number is omitted, the microcode image is downloaded to all line cards.
Step 2	`Router(config)# microcode reload slot-number`	Reloads the microcode on the specified line card.
Step 3	`Router(config)# exit`	Exits configuration mode.

Step 4	`Router# execute-on slot slot-number show version`
	or
	`Router# attach slot-number`
		Connects to the line card and verifies that the new Cisco IOS image is on the line card by checking the version number in the display output.

For further configuration information for Cisco 12000 series routers, see the documentation for Cisco IOS Release 11.2, Cisco IOS Release 12.0S, and Cisco IOS Release 12.2S, available on Cisco.com. For further platform specific documentation see http://www.cisco.com/univercd/cc/td/doc/product/core/.

This document first published April 30, 2001. Last updated February 15, 2005 (minor update).

Maintaining System Memory

This chapter describes how to maintain and use the different types of memory on your router. This document applies to Cisco IOS Release 12.2.

For a complete description of the memory commands mentioned in this chapter, refer to the "Router Memory Commands" chapter in the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Understanding Memory Types and Functions

Your router has many different locations where it can store images, configuration files, and microcode. Refer to your hardware documentation for details on which types of memory your routing device contains, where files can be stored (saved), and where images and boot images are located by default. This section provides information on the following memory types:

- DRAM
- EPROM
- NVRAM
- Flash Memory

DRAM

Dynamic random-access memory (DRAM) contains two types of memory:

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

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

DRAM often comes on dual in-line memory modules (DIMMs).

EPROM

Erasable programmable read-only memory (EPROM) is often referred to simply as ROM. On Cisco devices, the EPROM often contains the following:

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

NVRAM

Non-volatile random-access-memory (NVRAM) stores the following information:

- Startup configuration file for every platform except Class A Flash file system platforms (for Class A Flash file system platforms, the location of the startup configuration depends on the CONFIG_FILE Environment Variable).
- The software configuration register, which is used to determine which image to use when booting the router.

Flash Memory

Flash memory stores the Cisco IOS software image. On most platforms, it can store boot-images and/or configuration files.

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

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

- Internal Flash memory
  - Internal Flash memory often contains the system image.
  - Some platforms have two or more banks of Flash memory on one in-line memory module (in other words, on one SIMM). If the SIMM has two banks, it is sometimes referred to as *dual-bank Flash memory*. The banks can be partitioned into separate logical devices. See the "Partitioning Flash Memory" section for information about how to partition Flash memory.
- Bootflash
  - Bootflash often contains the boot image.
  - Bootflash sometimes contains the ROM Monitor.
- Flash memory PC cards or PCMCIA cards

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

Note Because some platforms, such as the Cisco 3600 series and Cisco the 7000 family, can boot images and load configuration files from several locations, these systems use special ROM monitor environment variables to specify the location and filename of images and configuration files that the router is to use for various functions.

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

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

Flash memory provides write protection against accidental erasing or reprogramming. Some platforms have a write-protect jumper which can be removed to prevent reprogramming of Flash memory. You must install the jumper when programming is required. Some platforms have write protect switched on Flash memory cards that you can use to protect data. You must set the switch to *unprotected* to write data to the Flash memory card. Refer to your hardware documentation for information on security jumpers and write protect switches.

Note The internal Flash and Flash memory cards of a system cannot be used as a contiguous bank of Flash memory.

Maintaining System Memory Task List

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

- Displaying System Memory Information
- Reallocating DRAM Memory for the Cisco 3600 Series
- Partitioning Flash Memory
- Using Flash Load Helper to Upgrade Software on Run-from-Flash Systems
- Formatting Flash Memory

The tasks in this chapter assume that you have a minimal configuration that you want to modify.

Displaying System Memory Information

Use the following commands in EXEC mode to display information about system memory:

Command	Purpose
`Router# show flash-filesystem: [all | chips | filesys]`	Lists information about Flash memory for Class A file systems.
`Router# show flash-filesystem: [partition number] [all | chips | detailed | err | summary]`	Lists information about Flash memory for Class B file systems.
`Router# show flash-filesystem:`	Lists information about Flash memory for Class C file systems.
`Router# show file systems`	Lists the names of the file systems currently supported on the router.

Partitioning Flash Memory

On most Class B Flash file systems, you can partition banks of Flash memory into separate, logical devices so that the router can hold and maintain two or more different software images. This partitioning allows you to write software into Flash memory while running software in another bank of Flash memory.

Systems that Support Partitioning

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

Note The CiscoFlash MIB variables support partitioned Flash.

Benefits of Partitioning Flash Memory

Partitioning Flash memory provides the following benefits:

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

Flash Load Helper Versus Dual Flash Bank

Flash load helper is a software option that enables you to upgrade system software on run-from-Flash systems that have a single bank of Flash memory. It is a lower-cost software upgrade solution than dual-bank Flash, which requires two banks of Flash memory on one SIMM. Flash load helper is only available on run-from-Flash platforms, such as the Cisco 2500 series, Cisco 3000, and Cisco 5200.

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

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

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

Partitioning Flash Memory

To partition Flash memory, use one of the forms of the following command in global configuration mode:

Command	Purpose
`Router(config)# partition flash partitions [size1 size2]`	Partitions Flash memory.
`Router(config)# partition flash-filesystem: [number-of-partitions] [partition-size]`	Partitions Flash memory on the Cisco 1600 and 3600 series.

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

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

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

Note To remove the partition, use the **no partition** command.

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

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

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

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

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

Flash Load Helper Features

Flash load helper performs the following functions:

- Confirms access to the specified source file on the specified server before erasing Flash memory and reloading to the ROM image for the actual upgrade.
- Warns you if the image being downloaded is not appropriate for the system.
- Prevents reloads to the ROM image for a Flash upgrade if the system is not set up for automatic booting and the user is not on the console terminal. In the event of a catastrophic failure during the upgrade, Flash load helper can bring up the boot ROM image as a last resort rather than forcing the system to wait at the ROM monitor prompt for input from the console terminal.
- Retries Flash downloads automatically up to six times. The retry sequence is as follows:
  - First try
  - Immediate retry
  - Retry after 30 seconds
  - Reload ROM image and retry
  - Immediate retry
  - Retry after 30 seconds
- Allows you to save any configuration changes made before you exit out of the system image.
- Notifies users logged in to the system of the impending switch to the boot ROM image so that they do not lose their connections unexpectedly.
- Logs console output during the Flash load helper operation into a buffer that is preserved through system reloads. You can retrieve the buffer contents from a running image. The output is useful when console access is unavailable or a failure occurs in the download operation.

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

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

Downloading Files Using the Flash Load Helper

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

Command	Purpose
Router# **copy tftp: flash:** Router# **copy rcp: flash:** Router# **copy ftp: flash:**	Loads the specified file to Flash memory.

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

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

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

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

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

```
Router# copy tftp: flash:

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

There are active users logged into the system.

Proceed? [confirm] y
System flash directory:
File Length  Name/status
1    2251320 abc/igs-kf.914
[2251384 bytes used, 1942920 available, 4194304 total]
Address or name of remote host [255.255.255.255]? 172.16.1.111
Source file name? abc/igs-kf.914
Destination file name [default = source name]? <Return>
Accessing file 'abc/igs-kf.914' on 172.16.1.111....
Loading from 172.16.13.111:
Erase flash device before writing? [confirm] n
File 'abc/igs-kf.914' already exists; it will be invalidated!
```

```
Invalidate existing copy of 'abc/igs-kf.914' in flash memory? [confirm] y
Copy 'abc/igs-kf.914' from TFTP server
as 'abc/igs-kf.914' into Flash WITHOUT erase? y

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

The Flash Load Helper operation verifies the request from the running image by trying to copy a single block from the remote server. Then the Flash load helper is executed, causing the system to reload to the ROM-based system image. If the file does not seem to be a valid image for the system, a warning is displayed and a separate confirmation is sought from you.

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

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

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

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

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

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

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

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

Command	Purpose
`Router# more flh:logfile`	View the console output generated during the Flash load helper operation.

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

Formatting Flash Memory

On Class A and Class C Flash file systems, you can format Flash memory. Formatting erases all information in Flash memory.

On the Cisco 7000 family, you must format a new Flash memory card before using it in a PCMCIA slot.

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

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

Flash Memory Formatting Process

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

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

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

Step 2 Format Flash memory.

To format Flash memory, use the following EXEC mode command:

Command	Purpose
Router# **format** [**spare** *spare-number*] *device1*: [[*device2*:][*monlib-filename*]]	Formats Flash memory.

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

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

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

Recovering from Locked Blocks

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

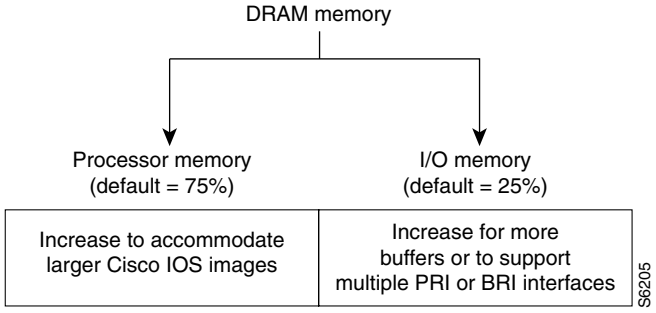
Caution Formatting a Flash memory card to recover from locked blocks will cause existing data to be lost.

Reallocating DRAM Memory for the Cisco 3600 Series

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

Cisco manufacturing configures most Cisco 3600 series routers to have 25 percent of the address space allocated to I/O memory and 75 percent allocated to processor memory. But for customer orders that require two or more ISDN PRI interfaces, DRAM memory is configured to provide 40 percent of the address space for I/O memory and 60 percent for processor memory. (See Figure 11.) Cisco Systems performs these DRAM memory adjustments before it ships each router.

Figure 11 Components and Uses of DRAM Memory for Cisco 3600 Series Routers



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

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

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

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

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

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

	Command	Purpose
Step 1	`Router#` **`show version`**	Displays the total amount of memory loaded on the router.
Step 2	`Router#` **`show memory`**[1]	Displays the amount of free memory.
Step 3	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 4	`Router(config)#` **`memory-size iomem`** *`I/O-memory-percentage`*[2]	Allocates processor memory and I/O memory.
Step 5	`Router(config)#` **`exit`**	Exits global configuration mode.
Step 6	`Router#` **`copy system:running-config nvram:startup-config`**	Saves the configuration to NVRAM.
Step 7	`Router#` **`reload`**	Reloads the router to run the new image.

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

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

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

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

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

Reallocate Processor Memory and I/O Memory Example

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

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

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

```
[OK]

Router# reload

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

Using Memory Scan on the Cisco 7500 Series

On Cisco 7500 series routers (including 7000 series with the RSP7000 card upgrade), a memory scanning feature is available. This feature adds a low-priority background process that searches all installed dynamic random-access memory (DRAM) for possible parity errors. If errors are found in memory areas that are not in use, this feature attempts to scrub (remove) the errors. The time to complete one memory scan and scrub cycle can range from 10 minutes to several hours, depending on the amount of installed memory. The impact of the Memory Scan feature on the central processing unit (CPU) is minimal. The feature can be controlled and monitored with the new **memory scan** and **show memory scan** command-line interface (CLI) commands.

The Memory Scan feature does not discriminate against different information types in DRAM; that is, it perceives text, data, and heap information in the same way. The feature continues to work when a memory cell is busy, although it might respond differently to errors found in different areas. The feature responds to errors in one or more of the following ways:

- A message is logged for all errors found. Each message contains an explanation of the error and suggests corrective action if applicable.
- For errors in heap storage control blocks, attempts are made to scrub errors in the free blocks. If an error is scrubbed, no further action occurs, but there is an entry in the error log. If it is not scrubbed, the block that contains the error is linked to a bad-memory list which will not be allocated to users. If the memory block is large, the block is split and only a small portion containing the error is linked to a bad-memory list.
- For errors in a busy block, or in other areas such as text or data, an error message is produced but no further action is taken, preventing damage to living data.

Configuring and Verifying Memory Scan

Use the **memory scan** command in global configuration mode to enable the feature.

Use the **more system:running-configuration** command in privileged EXEC mode to verify that memory scan appears in the running configuration.

Use the **show memory scan** command to monitor the number and type of parity errors on your system. Use the **show memory scan** command in privileged EXEC mode. In the following example, the feature is enabled and no parity errors are found:

```
Router# show memory scan
Memory scan is on.
No parity error has been detected.
```

If the Memory Scan feature has not been configured, or has been turned off, the **show memory scan** command generates a report. In the following example, Memory Scan is turned off:

```
Router# show memory scan
Memory scan is off
No parity error has been detected.
```

If errors are detected in the system, the **show memory scan** command generates an error report. In the following example, Memory Scan detected a parity error:

```
Router# show memory scan
Memory scan is on.
Total Parity Errors 1.
Address BlockPtr BlckSize Disposit   Region Timestamp
6115ABCD 60D5D090   9517A4  Scrubed    Local 16:57:09 UTC Thu Mar 18
```

For an explanation of the error report fields, see the full details on the **show memory scan** command in the "Router Memory Commands" chapter of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*.

Rebooting

This chapter describes the basic procedure a Cisco device (such as a router) performs when it reboots, how to alter the procedure, and how to use the ROM monitor.

For a complete description of the booting commands mentioned in this chapter, refer to the "Booting Commands" chapter in the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Understanding Rebooting Procedures

The following sections describe what happens when the router reboots:

- Which Configuration File Does the Router Use upon Startup?
- Which Image Does the Router Use upon Startup?

Which Configuration File Does the Router Use upon Startup?

On all platforms except Class A Flash file system platforms:

- If the configuration register is set to ignore NVRAM, the router enters setup mode.
- If the configuration register is not set to ignore NVRAM,
  - The startup software checks for configuration information in NVRAM.
  - If NVRAM holds valid configuration commands, the Cisco IOS software executes the commands automatically at startup.
  - If the software detects a problem with NVRAM or the configuration it contains (a CRC checksum error), it enters **setup** mode and prompts for configuration.

On Class A Flash file system platforms:

- If the configuration register is set to ignore NVRAM, the router enters setup mode.
- If the configuration register is not set to ignore NVRAM,
  - The startup software uses the configuration pointed to by the CONFIG_FILE environment variable.
  - When the CONFIG_FILE environment variable does not exist or is null (such as at first-time startup), the router uses NVRAM as the default startup device.
  - When the router uses NVRAM to start up and the system detects a problem with NVRAM or the configuration it contains, the router enters **setup** mode.

Problems can include a bad checksum for the information in NVRAM or an empty NVRAM with no configuration information. Refer to the "Troubleshooting Hardware and Booting Problems" chapter publication *Internetwork Troubleshooting Guide* for troubleshooting procedures. See the "Using Setup for Configuration Changes" chapter in this publication for details on the **setup** command facility. For more information on environment variables, refer to the "Setting Environment Variables" section.

Which Image Does the Router Use upon Startup?

When a router is powered on or rebooted, the following events happen:

- The ROM monitor initializes.
- The ROM monitor checks the boot field (the lowest four bits) in the configuration register.
  - If the last digit of the boot field is 0 (for example, 0x100), the system does not boot. Instead the system enters ROM monitor mode and waits for user intervention. From ROM monitor mode, you can manually boot the system using the **boot** or **b** command.
  - If the last digit of the boot field is1 (for example, 0x101), the boot helper image is loaded from ROM. (On some platforms, the boot helper image is specified by the BOOTLDR environment variable.)
  - If the last digit of the boot field is 2 through F (for example, 0x102 through 0x10F), the router boots the first valid image specified in the configuration file or specified by the BOOT environment variable.

Note The configuration register boot field value is expressed in hexadecimal. Because the boot field only encompasses the last four bits (represented by the last hexadecimal digit) of the configuration register value, the only digit we are concerned with in this discussion is the last digit. The makes 0x1 (0000 0001) equivalent to 0x101 (1 0000 0001) in discussions of the boot field, as in both cases the last four bits are 0001.

When the boot field is 0x102 through 0x10F, the router goes through each **boot system** command in order until it boots a valid image. If bit 13 in the configuration register is set, each command will be tried once (bit 13 is indicated by the position occupied by *b* in the following hexadecimal notation: 0x*b*000). If bit 13 is not set, the **boot system** commands specifying a network server will be tried up to five more times. The timeouts between each consecutive attempt are 2, 4, 16, 256, and 300 seconds.

If the router cannot find a valid image, the following events happen:

- If all boot commands in the system configuration file specify booting from a network server and all commands fail, the system attempts to boot the first valid file in Flash memory.
- If the "boot-default-ROM-software" option in the configuration register is set, the router will start the boot image (the image contained in boot ROM or specified by the BOORLDR environment variable).
- If the "boot-default-ROM-software" option in the configuration register is not set, the system waits for user intervention at the ROM monitor prompt. You must boot the router manually.
- If a fully functional system image is not found, the router will not function and must be reconfigured through a direct console port connection.

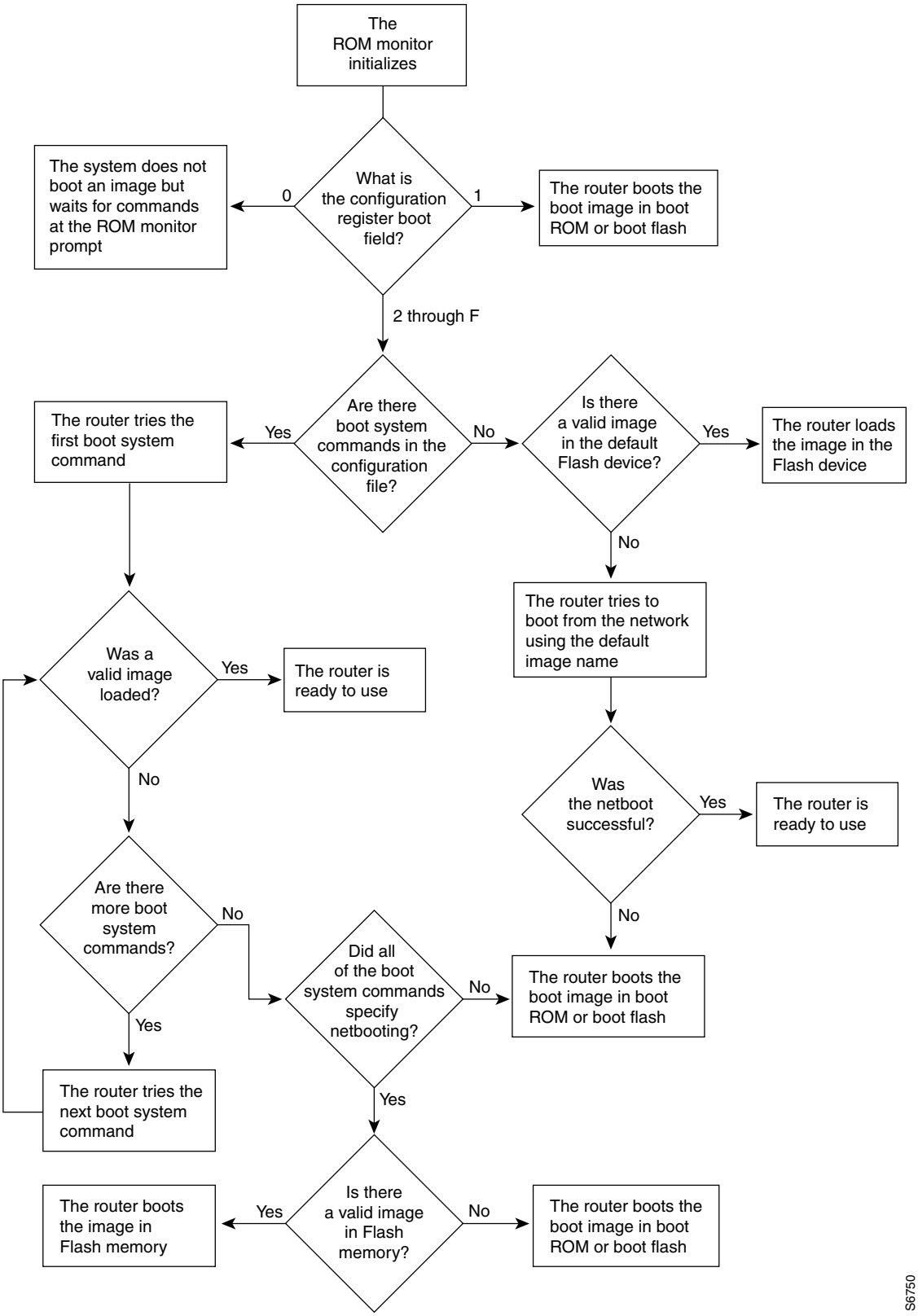
Note Refer to your platform documentation for information on the default location of the boot image.

When looking for a bootable file in Flash memory:

- The system searches for the filename in Flash memory. If a filename is not specified, the software searches through the entire Flash directory for a bootable file instead of picking only the first file.
- The system attempts to recognize the file in Flash memory. If the file is recognized, the software decides whether it is bootable by performing the following checks:
  - For run-from-Flash images, the software determines whether it is loaded at the correct execution address.
  - For run-from-RAM images, the software determines whether the system has enough RAM to execute the image.

Figure 12 illustrates the basic booting decision process.

Figure 12 Booting Process



Rebooting Task List

Tasks related to rebooting are described in the following sections:

- Displaying Boot Information
- Modifying the Configuration Register Boot Field
- Setting Environment Variables
- Scheduling a Reload of the System Image
- Entering ROM Monitor Mode
- Manually Loading a System Image from ROM Monitor

Displaying Boot Information

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

Command	Purpose
`Router# show bootvar`	Lists the contents of the BOOT environment variable, the name of the configuration file pointed to by the CONFIG_FILE environment variable, and the contents of the BOOTLDR environment variable.
`Router# more nvram:startup-config`	Lists the startup configuration information. On all platforms except the Class A Flash file systems, the startup configuration is usually in NVRAM. On Class A Flash file systems, the CONFIG_FILE environment variable points to the startup configuration, defaulting to NVRAM.
`Router# show version`	Lists the system software release version, system image name, configuration register setting, and other information.

Refer to the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference* for examples of these commands.

You can also use the o command (or the **confreg** command for some platforms) in ROM monitor mode to list the configuration register settings on some platforms.

Modifying the Configuration Register Boot Field

The configuration register boot field determines whether the router loads an operating system image, and if so, where it obtains this system image. This section contains the following topics:

- How the Router Uses the Boot Field
- Hardware Versus Software Configuration Register Boot Fields
- Modifying the Software Configuration Register Boot Field

Refer to the documentation for your platform for more information on the configuration register.

How the Router Uses the Boot Field

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

- When the entire boot field equals 0-0-0-0 (0x0), the router does not load a system image. Instead, it enters ROM monitor or "maintenance" mode from which you can enter ROM monitor commands to manually load a system image. Refer to the "Manually Loading a System Image from ROM Monitor" section for details on ROM monitor mode.
- When the entire boot field equals 0-0-0-1 (0x1), the router loads the boot helper or rxboot image.
- When the entire boot field equals a value between 0-0-1-0 (0x2) and 1-1-1-1 (0xF), the router loads the system image specified by **boot system** commands in the startup configuration file. When the startup configuration file does not contain **boot system** commands, the router tries to load a default system image stored on a network server.

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

Hardware Versus Software Configuration Register Boot Fields

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

Most platforms have use a software configuration register. Refer to your hardware documentation for information on the configuration register for your platform.

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

Modifying the Software Configuration Register Boot Field

To modify the software configuration register boot field, use the following commands:

	Command	Purpose
Step 1	`Router# show version`	Obtains the current configuration register setting. The configuration register is listed as a hexadecimal value.
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# config-register` *value*	Modifies the existing configuration register setting to reflect the way in which you want to load a system image. The configuration register value is in hexadecimal form with a leading "0x."
Step 4	`Router(config)# end`	Exits configuration mode.

	Command	Purpose
Step 5	`Router# show version`	(Optional) Verifies that the configuration register setting is correct. Repeat steps 2 through 5 if the setting is not correct.
Step 6	`Router# copy running-config startup-config`	Saves the running configuration to the startup configuration.
Step 7	`Router# reload`	(Optional) Reboots the router to make your changes take effect.

In ROM monitor mode, use the **o** command or the **confreg** command on some platforms to list the value of the configuration register boot field.

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

- 0 to load the system image manually using the **boot** command in ROM monitor mode.
- 1 to load the system image from boot ROMs. On the Cisco 7200 series and Cisco 7500 series, this setting configures the system to automatically load the system image from bootflash.
- 2–F to load the system image from **boot system** commands in the startup configuration file or from a default system image stored on a network server.

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

Modifying the Software Configuration Register Boot Field Example

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

```
Router1# show version

Cisco IOS (tm) Software
4500 Software (C4500-J-M), Version 11.1(10.4), RELEASE SOFTWARE
Copyright (c) 1986-1997 by Cisco Systems, Inc.
Compiled Mon 07-Apr-97 19:51 by lmiller
Image text-base: 0x600088A0, data-base: 0x60718000

ROM: System Bootstrap, Version 5.1(1), RELEASE SOFTWARE (fc1)
FLASH: 4500-XBOOT Bootstrap Software, Version 10.1(1), RELEASE SOFTWARE (fc1)

Router1 uptime is 6 weeks, 5 days, 2 hours, 22 minutes
System restarted by error - a SegV exception, PC 0x6070F7AC
System image file is "c4500-j-mz.111-current", booted via flash

cisco 4500 (R4K) processor (revision 0x00) with 32768K/4096K bytes of memory.
Processor board ID 01242622
R4600 processor, Implementation 32, Revision 1.0
G.703/E1 software, Version 1.0.
Bridging software.
SuperLAT software copyright 1990 by Meridian Technology Corp).
X.25 software, Version 2.0, NET2, BFE and GOSIP compliant.
TN3270 Emulation software (copyright 1994 by TGV Inc).
Basic Rate ISDN software, Version 1.0.
2 Ethernet/IEEE 802.3 interfaces.
2 Token Ring/IEEE 802.5 interfaces.
4 ISDN Basic Rate interfaces.
```

```
128K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (Read/Write)
4096K bytes of processor board Boot flash (Read/Write)

Configuration register is 0x2100

Router1# configure terminal
Router1(config)# config-register 0x210F
Router1(config)# end
Router1# reload
```

Setting Environment Variables

Because many platforms can boot images from several locations, these systems use special ROM monitor environment variables to specify the location and filename of images that the router is to use. In addition, Class A Flash file systems can load configuration files from several locations and use an environment variable to specify startup configurations.

These special environment variables are as follows:

- BOOT Environment Variable
- BOOTLDR Environment Variable
- CONFIG_FILE Environment Variable

BOOT Environment Variable

The BOOT environment variable specifies a list of bootable system images on various file systems. Refer to the "Specify the Startup System Image in the Configuration File" section in the "Loading and Maintaining System Images and Microcode" chapter of the *Configuration Fundamentals Configuration Guide*. After you save the BOOT environment variable to your startup configuration, the router checks the variable upon startup to determine the device and filename of the image to boot.

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

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

BOOTLDR Environment Variable

The BOOTLDR environment specifies the Flash file system and filename containing the boot image that the ROM monitor uses if it cannot find a valid system image. In addition, a boot image is required to boot the router with an image from a network server.

You can change the BOOTLDR environment variable on platforms that use a software boot image rather than boot ROMs. On these platforms, the boot image can be changed without having to replace the boot ROM.

This environment variable allows you to have several boot images. After you save the BOOTLDR environment variable to your startup configuration, the router checks the variable upon startup to determine which boot image to use if the system cannot be loaded.

Note Refer to your platform documentation for information on the default location of the boot image.

CONFIG_FILE Environment Variable

For Class A Flash file systems, the CONFIG_FILE environment variable specifies the file system and filename of the configuration file to use for initialization (startup). Valid file systems can include **nvram:**, **bootflash:**, **slot0:**, and **slot1:**. Refer to the "Location of Configuration Files" section on page 146 in the "Modifying, Downloading, and Maintaining Configuration Files" chapter for more information on devices. After you save the CONFIG_FILE environment variable to your startup configuration, the router checks the variable upon startup to determine the location and filename of the configuration file to use for initialization.

The router uses the NVRAM configuration during initialization when the CONFIG_FILE environment variable does not exist or when it is null (such as at first-time startup). If the router detects a problem with NVRAM or a checksum error, the router enters **setup** mode. Refer to the "Using Setup for Configuration Changes" chapter in this publication for more information on the **setup** command facility.

Controlling Environment Variables

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

Refer to the "Specify the Startup System Image in the Configuration File" section in the "Loading and Maintaining System Images" chapter of this book for details on setting the BOOT environment variable. Refer to the "Specify the Startup Configuration File" section in the "Managing Configuration Files" chapter of this document for details on setting the CONFIG_FILE variable.

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

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

Use the **more nvram:startup-config** command to display the contents of the configuration file pointed to by the CONFIG_FILE environment variable.

Setting the BOOTLDR Environment Variable

To set the BOOTLDR environment variable, use the following commands, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# dir [flash-filesystem:]`	Verifies that internal Flash or bootflash contains the boot helper image.
Step 2	`Router# configure terminal`	Enters the configuration mode from the terminal.
Step 3	`Router(config)# boot bootldr file-url`	Sets the BOOTLDR environment variable to specify the Flash device and filename of the boot helper image. This step modifies the runtime BOOTLDR environment variable.
Step 4	`Router# end`	Exits configuration mode.
Step 5	`Router# copy system:running-config nvram:startup-config`	Saves the configuration you just performed to the system startup configuration.
Step 6	`Router# show bootvar`	(Optional) Verifies the contents of the BOOTLDR environment variable.

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

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

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

Configuration register is 0x0
```

Scheduling a Reload of the System Image

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

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

Configuring a Scheduled Reload

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

Command	Purpose
Router# **reload in** [*hh*:]*mm* [*text*]	Schedules a reload of the software to take effect in *mm* minutes (or *hh* hours and *mm* minutes) from now.
Router# **reload at** *hh*:*mm* [*month day* | *day month*] [*text*]	Schedules a reload of the software to take place at the specified time (using a 24-hour clock). If you specify the month and day, the reload is scheduled to take place at the specified time and date. If you do not specify the month and day, the reload takes place at the specified time on the current day (if the specified time is later than the current time), or on the next day (if the specified time is earlier than the current time). Specifying 00:00 schedules the reload for midnight.

Note The **at** keyword can only be used if the system clock has been set on the router (either through NTP, the hardware calendar, or manually). The time is relative to the configured time zone on the router. To schedule reloads across several routers to occur simultaneously, the time on each router must be synchronized with NTP. For information on configuring NTP, see the "Performing Basic System Management" chapter in the *Cisco IOS Network Management Configuration Guide*, Release 12.4.

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

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

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

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

Display Information about a Scheduled Reload

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

Command	Purpose
Router# **show reload**	Displays reload information, including the time the reload is scheduled to occur, and the reason for the reload if it was specified when the reload was scheduled.

Cancel a Scheduled Reload

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

Command	Purpose
`Router# reload cancel`	Cancels a previously scheduled reload of the software.

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

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

Entering ROM Monitor Mode

During the first 60 seconds of startup, you can force the router to stop booting. The router will enter ROM monitor mode, where you can change the configuration register value or boot the router manually.

To stop booting and enter ROM monitor mode, use the following commands in EXEC mode:

	Command	Purpose

Step 1	`Router# reload`
	
	`Press the Break[1] key during the first 60 seconds while the system is booting.`
		Enter ROM monitor mode from privileged EXEC mode.

Step 2	`?`	List the ROM monitor commands.

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

Timesaver If you are planning to use ROM monitor mode on a regular basis, or wish users to load using ROM monitor commands, you can configure the system to default to ROMMON. To automatically boot your system in ROM monitor mode, reset the configuration register to 0x0 by using the **config-register 0x0** configuration command. The new configuration register value, 0x0, takes effect after the router or access server is rebooted with the **reload** command. If you set the configuration to 0x0, you will have to manually boot the system from the console each time you reload the router or access server.

To exit ROMMON mode, use the continue command. If you have changed the configuration, use the **copy running-config startup-config** command and then issue the **reload** command to save your configuration changes.

Aliasing ROM Monitoring Commands

The ROM monitor supports command aliasing modeled on the aliasing function built into the Korn shell. The **alias** command is used to set and view aliased names. This allows the user to alias command names to a letter or word. Aliasing is often used to shorten command names or automatically invoke command options.

Aliases are stored in NVRAM and remain intact across periods of no power. These are some of the set aliases:

- **b**—boot
- **h**—history
- **i**—intialize/reset
- **r**—repeat
- **k**—stack
- **?**—help

The following example shows a pre-aliased menu-type list for ROMMON commands:

```
> ?
$ state      Toggle cache state (? for help)
B [filename] [TFTP Server IP address | TFTP Server Name]
             Load and execute system image from ROM or from TFTP server
C [address]  Continue execution [optional address]
D /S M L V   Deposit value V of size S into location L with modifier M
E /S M L     Examine location L with size S with modifier M
G [address]  Begin execution
H            Help for commands
I            Initialize
K            Stack trace
L [filename] [TFTP Server IP address | TFTP Server Name]
             Load system image from ROM or from TFTP server, but do not
             begin execution
O            Show configuration register option settings
P            Set the break point
S            Single step next instruction
T function   Test device (? for help)
Deposit and Examine sizes may be B (byte), L (long) or S (short).
Modifiers may be R (register) or S (byte swap).
Register names are: D0-D7, A0-A6, SS, US, SR, and PC
```

If your options appear in the above menu-type format, you can use the listed aliased commands. To initialize the router or access server, enter the **i** command. The **i** command causes the bootstrap program to reinitialize the hardware, clear the contents of memory, and boot the system. To boot the system image file, use the **b** command.

The ROM monitor software characteristics will vary depending on your platform. For further details on ROM monitor mode commands, refer to the appropriate hardware installation guide, or perform a search on Cisco.com.

Manually Loading a System Image from ROM Monitor

If your router does not find a valid system image, or if its configuration file is corrupted at startup, or the configuration register is set to enter ROM monitor mode, the system enters ROM monitor mode. From this mode, you can manually load a system image from the following locations:

- Internal Flash memory or a Flash memory PC card
- A network server file
- ROM
- A local or remote computer, using the Xmodem or Ymodem protocol (Cisco 1600 series and Cisco 3600 series routers only)

You may only boot from a location if the router can store an image there. Therefore, not all platforms can manually load from these locations.

You can also enter ROM monitor mode by restarting the router and then pressing the **Break** key or issuing a "send break" command from a telnet session during the first 60 seconds of startup.

Manually Booting from Flash Memory in ROMMON

To manually boot from Flash memory, use the following command in ROM monitor mode:

Command	Purpose

`ROMMON >` **`boot flash`** `[`*`filename`*`]`
`ROMMON >` **`boot flash`** *`partition-number`***`:`**`[`*`filename`*`]`
`ROMMON >` **`boot flash flash:`**`[` *`partition-number`***`:`**`] [`*`filename`*`]`
`ROMMON >` **`boot`** `[`*`flash-fs`***`:`**`][`*`partition-number`***`:`**`][`*`filename`*`]`
`(Cisco 1600 series and Cisco 3600 series)`
`ROMMON >`
`boot` *`device`***`:`**`[`*`filename`*`] (Cisco 7000 family)`
	Manually boot the router from Flash. Refer to your hardware documentation for the correct form of this command to use.

If the filename is not specified, the first bootable file found in the device and partition is used.

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

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

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

              Restricted Rights Legend

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

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

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

Booting gs7-k from flash memory RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
```

```
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRR [OK - 1903912/13765276 bytes]
F3: 1858676+45204+166896 at 0x1000

              Restricted Rights Legend

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

The following command instructs the ROM monitor to boot the first file in the first partition of internal Flash memory:

```
> boot flash:
```

This command instructs the ROM monitor to boot the first file in the second partition of the Flash memory card in slot 0:

```
> boot slot0:2:
```

In this example, the ROM monitor boots the file named image name from the third partition of the Flash memory card in slot 0:

```
> boot slot0:3:imagename
```

The following command fails to specify a valid device type (**flash:**, **slot0:**, or **slot1:**), so the ROM monitor invokes the boot helper to boot a system image.

```
> boot flash
```

Manually Booting from a Network File in ROMMON

To manually boot from a network file, use the following command in ROM monitor mode:

Command	Purpose
ROMMON > **boot** *filename* [*ip-address*]	Manually boots the router from a network file.

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

```
>boot network1
```

Manually Booting from ROM in ROMMON

To manually boot the router from ROM, use the following command in ROM monitor mode:

Command	Purpose
ROMMON > **boot**	Manually boots the router from ROM.

On the Cisco 7200 series and Cisco 7500 series, the **boot** command loads the first bootable image located in bootflash.

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

```
>boot
```

Manually Booting Using MOP in ROMMON

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

To manually boot the router using MOP, use the following command in ROM monitor mode:

Command	Purpose
ROMMON > **boot system mop** *filename* [*mac-address*] [*interface*]	Manually boots the router using MOP.

The Cisco 7200 series and Cisco 7500 series do not support the **boot mop** command.

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

```
>boot mop network1
```

Exiting from ROMMON

To return to EXEC mode from the ROM monitor, you must continue loading from the default system image. To exit ROMMON mode and resume loading, use the following command in ROM monitor mode:

Command	Purpose
ROMMON > **continue**	Resumes loading the startup configuration file and brings the user to EXEC mode.

Configuring Basic File Transfer Services

This chapter describes how to configure a router as a Trivial File Transfer Protocol (TFTP) or Reverse Address Resolution Protocol (RARP) server, configure the router to forward extended BOOTP requests over asynchronous interfaces, and configure rcp, rsh, and FTP in Cisco IOS Release 12.2.

For a complete description of the file transfer function commands mentioned in this chapter, refer to the "Basic File Transfer Services Commands" chapter in the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Basic File Transfer Services Configuration Task List

To configure basic file transfer services, perform any of the tasks described in the following sections:

- Configuring a Router as a TFTP or RARP Server
- Configuring System BOOTP Parameters
- Configuring a Router to Use rsh and rcp
- Configuring a Router to Use FTP Connections

All tasks in this chapter are optional.

Configuring a Router as a TFTP or RARP Server

It is too costly and inefficient to have a machine that acts only as server on every network segment. However, when you do not have a server on every segment, your network operations can incur substantial time delays across network segments. You can configure a router to serve as a RARP or TFTP server to reduce costs and time delays in your network while allowing you to use your router for its regular functions.

Typically, a router that is configured as a TFTP or RARP server provides other routers with system image or router configuration files from its Flash memory. You can also configure the router to respond to other types of service requests, such as requests.

Configuring a Router as a TFTP Server

As a TFTP server host, the router responds to TFTP Read Request messages by sending a copy of the system image contained in ROM or one of the system images contained in Flash memory to the requesting host. The TFTP Read Request message must use one of the filenames that are specified in the configuration.

Note For the Cisco 7000 family, 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.

Flash memory can be used as a TFTP file server for other routers on the network. This feature allows you to boot a remote router with an image that resides in the Flash server memory.

Some Cisco devices allow you to specify one of the different Flash memory locations (**bootflash:**, **slot0:**, **slot1:**, **slavebootflash:**, **slaveslot0:**, or **slaveslot1:**) as the TFTP server.

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

TFTP Router Configuration Prerequisite Tasks

The server and client router must be able to reach each other before the TFTP function can be implemented. Verify this connection by testing the connection between the server and client router (in either direction) using the **ping** *a.b.c.d* command (where *a.b.c.d* is the address of the client device). After the **ping** command is issued, connectivity is indicated by a series of exclamation points (!), while a series of periods (.) plus `[timed out]` or `[failed]` indicates that the connection attempt failed. If the connection fails, reconfigure the interface, check the physical connection between the Flash server and client router, and ping again.

After you verify the connection, ensure that a TFTP-bootable image is present on the server. This is the system software image the client router will boot. Note the name of this software image so you can verify it after the first client boot.

Caution For full functionality, the software image sent to the client must be the same type as the ROM software installed on the client router. For example, if the server has X.25 software, and the client does not have X.25 software in ROM, the client will not have X.25 capabilities after booting from the server's image in Flash memory.

Enabling the TFTP Server

To enable TFTP server operation, use the following commands, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.

Step 2	`Router(config)# tftp-server flash [partition-number:]filename1 [alias filename2] [access-list-number]`
	or
	`Router(config)# tftp-server flash device:filename` (Cisco 7000 family only)
	or
	`Router(config)# tftp-server flash [device:][partition-number:]filename` (Cisco 1600 series and Cisco 3600 series only)
	or
	`Router(config)# tftp-server rom alias filename1 [access-list-number]`
		Specifies the system image to send in response to Read Requests. You can enter multiple lines to specify multiple images.

Step 3	`Router(config)# end`	Ends the configuration session and returns you to privileged EXEC mode.
Step 4	`Router# copy running-config startup-config`	Saves the running configuration to the startup configuration file.

The TFTP session can sometimes fail. TFTP generates the following special characters to help you determine why a TFTP session fails:

- An “E” character indicates that the TFTP server received an erroneous packet.
- An “O” character indicates that the TFTP server received an out-of-sequence packet.
- A period (.) indicates a timeout.

For diagnosing any undue delay in the transfer, the output is useful. For troubleshooting procedures, refer to the *Internetwork Troubleshooting Guide* publication.

In the following example, the system can use TFTP to send copies of the Flash memory file *version-10.3* in response to a TFTP Read Request for that file. The requesting host is checked against access list 22.

```
tftp-server flash version-10.3 22
```

In the following example, the system can use TFTP to send a copy of the ROM image *gs3-k.101* in response to a TFTP Read Request for the *gs3-k.101* file:

```
tftp-server rom alias gs3-k.101
```

The following example a router to send a copy of the file *gs7-k.9.17* in Flash memory in response to a TFTP Read Request. The client router must reside on a network specified by access list 1. Thus, in the example, the any clients on network 172.16.101.0 are permitted access to the file.

```
Server# configure terminal
Enter configuration commands, one per line. End with CTRL/Z
Server(config)# tftp-server flash gs7-k.9.17 1
Server(config)# access-list 1 permit 172.16.101.0 0.0.0.255
```

```
Server(config)# end
Server# copy running-config startup-config
[ok]
Server#
```

Configuring the Client Router

Configure the client router to first load a system image from the server. As a backup, configure the client router to then load its own ROM image if the load from a server fails. To configure the client router, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# no boot system`	(Optional) Removes all previous **boot system** statements from the configuration file.
Step 3	`Router(config)# boot system [tftp] filename [ip-address]`	Specifies that the client router load a system image from the server.
Step 4	`Router(config)# boot system rom`	Specifies that the client router loads its own ROM image if the load from a server fails.
Step 5	`Router(config)# config-register value`	Sets the configuration register to enable the client router to load a system image from a network server.
Step 6	`Router(config)# end`	Exits global configuration mode.
Step 7	`Router# copy running-config startup-config`	Saves the configuration file to your startup configuration.
Step 8	`Router# reload`	(Optional) Reloads the router to make your changes take effect.

After the system reloads, you should use the **show version** EXEC mode command to verify that the system booted the desired image.

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

In the following example, the router is configured to boot from a specified TFTP server:

```
Client# configure terminal
Enter configuration commands, one per line. End with CTRL/Z
Client(config)# no boot system
Client(config)# boot system c5300-js-mz.121-5.T.bin 172.16.111.111
Client(config)# boot system rom
Client(config)# config-register 0x010F
Client(config)# end
Client# copy running-config startup-config
[ok]
Client# 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 `c5300-js-mz.121-5.T.bin` on the TFTP server with an IP address of 172.16.111.111. Failing this, the client router will boot from its system ROM in response to the **boot system rom** command, which is included as a backup in case of a network problem. The **copy running-config startup-config** command copies the configuration to the startup configuration, and the **reload** command boots the system.

Note The system software to be booted from the server must reside in Flash memory on the server. If it is not in Flash memory, the client router will boot the server's system ROM.

The following example shows sample output of the **show version** command after the router has rebooted:

```
Router> show version

Cisco Internetwork Operating System Software
Cisco IOS (tm) 5300 Software (C5300-JS-M), Version 12.1(5)T,  RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2000 by Cisco Systems, Inc.
Compiled Sat 11-Nov-00 03:03 by joe
Image text-base: 0x60008958, data-base: 0x611C6000

ROM: System Bootstrap, Version 11.2(9)XA, RELEASE SOFTWARE (fc2)
BOOTFLASH: 5300 Software (C5300-BOOT-M), Version 12.0(7)T,  RELEASE SOFTWARE (f)

Router uptime is 8 weeks, 4 days, 22 hours, 36 minutes
System returned to ROM by power-on
System restarted at 00:37:38 UTC Thu Feb 22 2001
System image file is "flash:c5300-js-mz.121-5.T.bin"

.
.
.

Configuration register is 0x010F
```

The important information in this example is contained in the first line "Cisco IOS (tm).." and in the line that begins "System image file...." The "Cisco IOS (tm)..." line shows the version of the operating system in NVRAM. The "System image file...." line show the filename of the system image loaded from the TFTP server.

Configuring a Router as a RARP Server

Reverse Address Resolution Protocol (RARP) is a protocol in the TCP/IP stack that provides a method for finding IP addresses based on MAC (physical) addresses. This functionality is the reverse of broadcasting Address Resolution Protocols (ARPs), through which a host can dynamically discover the MAC-layer address corresponding to a particular IP network-layer address. RARP makes diskless booting of various systems possible (for example, diskless workstations that do not know their IP addresses when they boot, such as Sun workstations or PCs on networks where the client and server are on separate subnets). RARP relies on the presence of a RARP server with cached table entries of MAC-layer-to-IP address mappings.

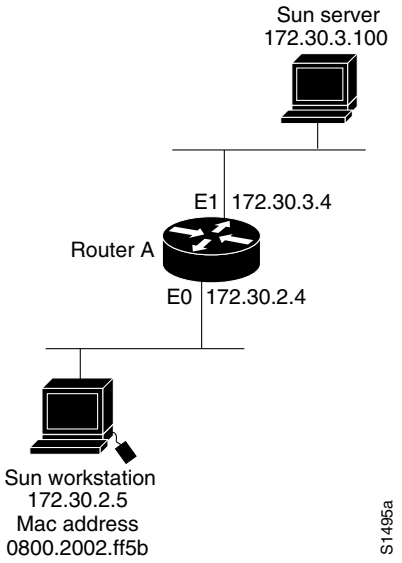
You can configure a Cisco router as a RARP server. This feature enables the Cisco IOS software to answer RARP requests.

To configure the router as a RARP server, use the following commands, beginning in global configuration mode:

Command	Purpose
`Router(config)# interface type [slot/]port`	Specifies the interface that you will be configuring the RARP service on and enters interface configuration mode for the specified interface.
`Router(config-if)# ip rarp-server ip-address`	Enables the RARP service on the router.

Figure 13 illustrates a network configuration in which a router is configured to act as a RARP server for a diskless workstation. In this example, the Sun workstation attempts to resolve its MAC (hardware) address to an IP address by sending a SLARP request, which is forwarded by the router to the Sun server.

Figure 13 Configuring a Router As a RARP Server



Router A has the following configuration:

```
! Allow the router to forward broadcast portmapper requests
ip forward-protocol udp 111
! Provide the router with the IP address of the diskless sun
arp 172.30.2.5 0800.2002.ff5b arpa
interface ethernet 0
! Configure the router to act as a RARP server, using the Sun Server's IP
! address in the RARP response packet.
ip rarp-server 172.30.3.100
! Portmapper broadcasts from this interface are sent to the Sun Server.
ip helper-address 172.30.3.100
```

The Sun client and server's IP addresses must use the same major network number because of a limitation with the current SunOS *rpc.bootparamd* daemon.

In the following example, an access server is configured to act as a RARP server.

```
! Allow the access server to forward broadcast portmapper requests
 ip forward-protocol udp 111
! Provide the access server with the IP address of the diskless sun
```

```
 arp 172.30.2.5 0800.2002.ff5b arpa
 interface ethernet 0
! Configure the access server to act as a RARP server, using the Sun Server's
! IP address in the RARP response packet.
 ip rarp-server 172.30.3.100
! Portmapper broadcasts from this interface are sent to the Sun Server.
 ip helper-address 172.30.3.100
```

Configuring System BOOTP Parameters

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

To configure extended BOOTP parameters for asynchronous interfaces, use the following command in global configuration mode:

Command	Purpose
`Router(config)# **async-bootp** *tag* [:*hostname*] *data*`	Configures extended BOOTP requests for asynchronous interfaces.

You can display the extended data that will be sent in BOOTP responses by using the following command in EXEC mode:

Command	Purpose
`Router# **show async bootp**`	Displays parameters for BOOTP responses.

For example, if the DNS server address is specified as extended data for BOOTP responses, you will see output similar to the following:

```
Router# show async bootp
The following extended data will be sent in BOOTP responses:

dns-server 172.22.53.210
```

For information about configuring your Cisco device as a BOOTP server, see the "Using AutoInstall and Setup" chapter.

Configuring a Router to Use rsh and rcp

Remote shell (rsh) gives users the ability to execute commands remotely. Remote copy (rcp) allows users to copy files to and from a file system residing on a remote host or server on the network. Cisco's implementation of rsh and rcp interoperates with the industry standard implementations. Cisco uses the abbreviation RCMD (Remote Command) to indicate both rsh and rcp.

This section is divided into the following sections:

- Specifying the Source Interface for Outgoing RCMD Communications
- About DNS Reverse Lookup for rcmd
- Enabling and Using rsh

- Enabling and Using rcp

Specifying the Source Interface for Outgoing RCMD Communications

You can specify the source interface for RCMD (rsh and rcp) communications. For example, the router can be configured so that RCMD connections use the loopback interface as the source address of all packets leaving the router. To specify the interface associated with RCMP communications, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip rcmd source-interface` *interface-id*	Specifies the interface address that will be used to label all outgoing rsh and rcp traffic.

Specifying the source-interface is most commonly used to specify a loopback interface. This allows you to associate a permanent IP address with RCMD communications. Having a permanent IP address is useful for session identification (remote device can consistently idendify the origin of packets for the session). A "well-known" IP address can also be used for security purposes, as you can then create access lists on remote devices which include the address.

About DNS Reverse Lookup for rcmd

As a basic security check, the Cisco IOS software does a reverse lookup of the client IP address using DNS for the remote command (rcmd) applications (rsh and rcp). This check is performed using a host authentication process.

When enabled, the system records the address of the requesting client. That address is mapped to a host name using DNS. Then a DNS request is made for the IP address for that host name. The IP address received is then checked against the original requesting address. If the address does not match with any of the addresses received from DNS, the rcmd request will not be serviced.

This reverse lookup is intended to help protect against "spoofing." However, please note that the process only confirms that the IP address is a valid routable address; it is still possible for a hacker to spoof the valid IP address of a known host.

This feature is enabled by default. You can disable the DNS check for RCMD (rsh and rcp) access using the the following command in global configuration mode:

Command	Purpose
`Router(config)# no ip rcmd domain-lookup`	Disables the Domain Name Service (DNS) reverse lookup function for remote command (rcmp) applications (rsh and rcp).

Enabling and Using rsh

You can use rsh (remote shell) to execute commands on remote systems to which you have access. When you issue the **rsh** command, a shell is started on the remote system. The shell allows you to execute commands on the remote system without having to log in to the target host.

You do not need to connect to the system, router, or access server and then disconnect after you execute a command if you use rsh. For example, you can use rsh to remotely look at the status of other devices without connecting to the target device, executing the command, and then disconnecting. This capability is useful for looking at statistics on many different routers. Configuration commands for enabling rsh use the acronym “rcmd”, which is short for “remote command”.

Maintaining rsh Security

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

You can enable rsh support on a router to allow users on remote systems to execute commands. However, our implementation of rsh does not support an *.rhosts* file. Instead, you must configure a local authentication database to control access to the router by users attempting to execute commands remotely using rsh. A local authentication database is similar to a UNIX *.rhosts* file. Each entry that you configure in the authentication database identifies the local user, the remote host, and the remote user.

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

To configure the router as an rsh server, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip rcmd remote-host local-username {ip-address | host} remote-username [enable [level]]`	Creates an entry in the local authentication database for each remote user who is allowed to execute rsh commands.
Step 2	`Router(config)# ip rcmd rsh-enable`	Enables the software to support incoming rsh commands.

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

Note When support of incoming rsh commands is disabled, you can still issue an rsh command to be executed on other routers that support the remote shell protocol and on UNIX hosts on the network.

The following example shows how to add two entries for remote users to the authentication database, and enable a router to support rsh commands from remote users:

```
ip rcmd remote-host Router1 172.16.101.101 rmtnetad1
ip rcmd remote-host Router1 172.16.101.101 netadmin4 enable
ip rcmd rsh-enable
```

The users, named *rmtnetad1* and *netadmin4*, are both on the remote host at IP address 172.16.101.101. Although both users are on the same remote host, you must include a unique entry for each user. Both users are allowed to connect to the router and remotely execute rsh commands on it after the router is enabled for rsh. The user named *netadmin4* is allowed to execute privileged EXEC mode commands on the router. Both authentication database entries give the router's host name *Router1* as the local username. The last command enables the router for to support rsh commands issued by remote users.

.

Executing Commands Remotely Using rsh

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

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

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

To execute a command remotely on a network server using rsh, use the following commands in user EXEC mode:

	Command	Purpose
Step 1	`Router> enable [password]`	Enters privileged EXEC mode.
Step 2	`Router# rsh {ip-address | host} [/user username] remote-command`	Executes a command remotely using rsh.

The following example executes the "ls -a" command in the home directory of the user sharon on mysys.cisco.com using rsh:

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

Enabling and Using rcp

The remote copy (rcp) commands rely on the rsh server (or daemon) on the remote system. To copy files using rcp, you do not need to create a server for file distribution, as you do with TFTP. You need only to have access to a server that supports the remote shell (rsh). (Most UNIX systems support rsh.) Because you are copying a file from one place to another, you must have read permission on the source file and write permission in the destination directory. If the destination file does not exist, rcp creates it for you.

Although Cisco's rcp implementation emulates the functions of the UNIX rcp implementation—copying files among systems on the network—Cisco's command syntax differs from the UNIX rcp command syntax. The Cisco IOS software offers a set of copy commands that use rcp as the transport mechanism. These rcp copy commands are similar in style to the Cisco IOS TFTP copy commands, but they offer an alternative that provides faster performance and reliable delivery of data. These improvements are possible because the rcp transport mechanism is built on and uses the Transmission Control Protocol/Internet Protocol (TCP/IP) stack, which is connection-oriented. You can use rcp commands to copy system images and configuration files from the router to a network server and vice versa.

You can also enable rcp support to allow users on remote systems to copy files to and from the router.

Configuring the Router to Accept rcp Requests from Remote Users

To configure the Cisco IOS software to support incoming rcp requests, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip rcmd remote-host local-username {ip-address | host} remote-username [enable [level]]`	Create an entry in the local authentication database for each remote user who is allowed to execute rcp commands.
Step 2	`Router(config)# ip rcmd rcp-enable`	Enable the software to support incoming rcp requests.

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

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

The following example shows how to add two entries for remote users to the authentication database and then enable the software to support remote copy requests from remote users. The users, named *netadmin1* on the remote host at IP address 172.16.15.55 and *netadmin3* on the remote host at IP address 172.16.101.101, are both allowed to connect to the router and remotely execute rcp commands on it after the router is enabled to support rcp. Both authentication database entries give the host name *Router1* as the local username. The last command enables the router to support for rcp requests from remote users.

```
ip rcmd remote-host Router1 172.16.15.55 netadmin1
ip rcmd remote-host Router1 172.16.101.101 netadmin3
ip rcmd rcp-enable
```

Configuring the Remote to Send rcp Requests

The rcp protocol requires a client to send a remote username on each rcp request to a server. When you copy a configuration file from a server to the router using rcp, the Cisco IOS software sends the first valid username in the following list:

1. The username set by the **ip rcmd remote-username** command, if the command is configured.
2. The remote username associated with the current tty (terminal) process. For example, if the user is connected to the router through Telnet and was authenticated through the **username** command, the router software sends the Telnet username as the remote username.

Note In Cisco products, ttys are commonly used in access servers. The concept of tty originated with UNIX. For UNIX systems, each physical device is represented in the file system. Terminals are called *tty devices*, which stands for *teletype*, the original UNIX terminal.

3. The router host name.

For **boot** commands using rcp, the software sends the router host name; you cannot explicitly configure the remote username.

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

If you are writing to the server, the rcp server must be properly configured to accept the rcp write request from the user on the router. For UNIX systems, you must add an entry to the *.rhosts* file for the remote user on the rcp server. For example, if the router contains the following configuration lines.

```
hostname Rtr1
ip rcmd remote-username User0
```

and the router's IP address translates to Router1.company.com, then the *.rhosts* file for User0 on the rcp server should contain the following line:

```
Router1.company.com Rtr1
```

Refer to the documentation for your rcp server for more details.

If the server has a directory structure, the configuration file or image is written or copied relative to the directory associated with the remote username on the server. Use the **ip rcmd remote-username** command to specify which directory on the server to use. For example, if the system image resides in the home directory of a user on the server, you can specify that user's name as the remote username.

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

To override the default remote username sent on rcp requests, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip rcmd remote-username username`	Specifies the remote username.

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

Configuring a Router to Use FTP Connections

You configure a router to transfer files between systems on the network using the File Transfer Protocol (FTP). With the Cisco IOS implementation of FTP, you can set the following FTP characteristics:

- Passive-mode FTP
- User name
- Password
- IP address

To configure these FTP characteristics, use any of the following commands in global configuration mode:

Command	Purpose
`Router(config)# ip ftp username string`	Specifies the user name to be used for the FTP connection.
`Router(config)# ip ftp password [type] password`	Specifies the password to be used for the FTP connection.

`Router(config)# ip ftp passive`
or
`Router(config)# no ip ftp passive`
	Configures the router to only use passive-mode FTP connections.
	or
	Allows all types of FTP connections (default).

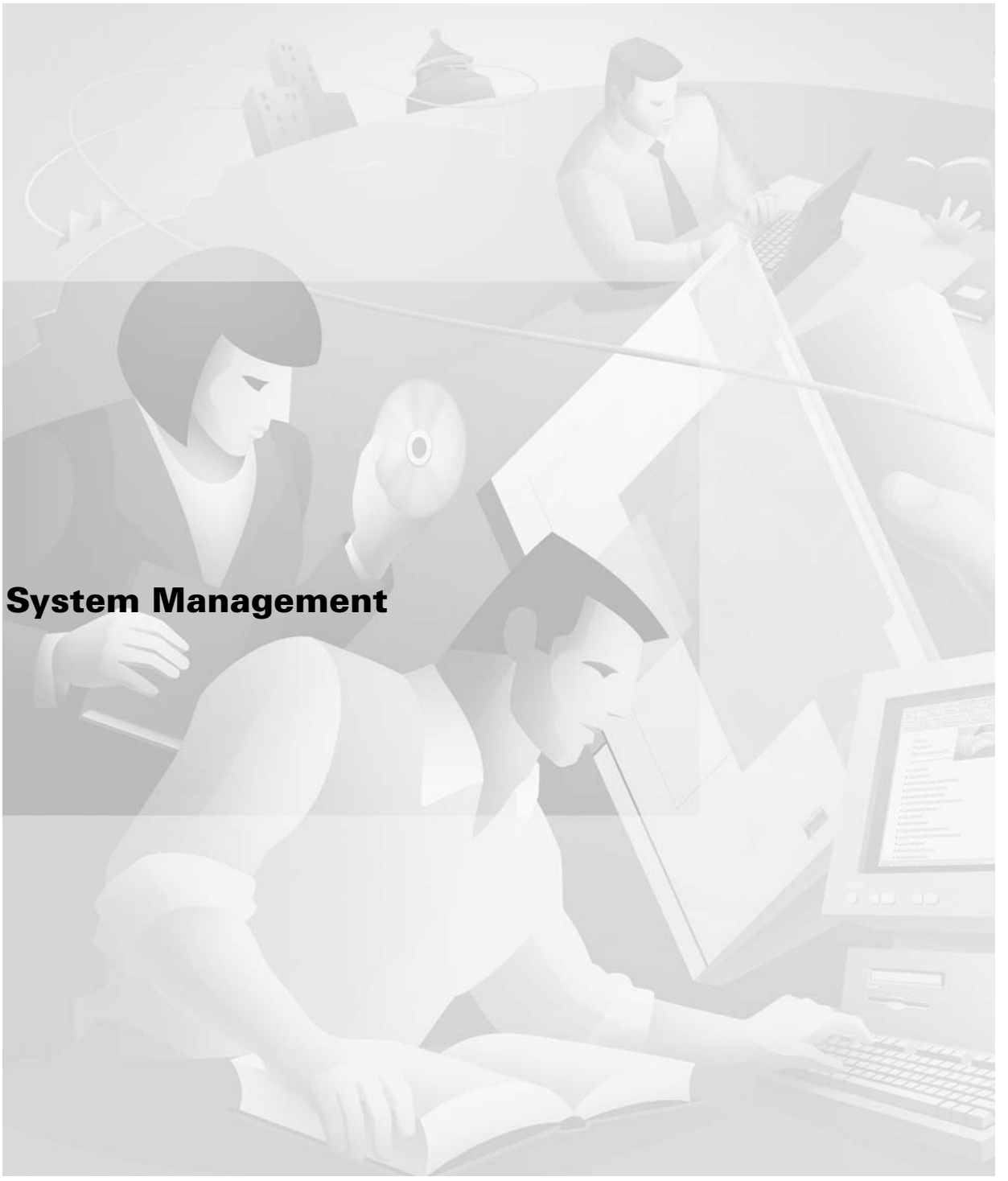
`Router(config)# ip ftp source-interface interface`	Specifies the source IP address for FTP connections.

The following example demonstrates how to capture a core dump using the Cisco IOS FTP feature. The router accesses a server at IP address 192.168.10.3 with login name zorro and password sword. The default passive-mode FTP is used, and the server is accessed using Token Ring interface to1 on the router where the core dump will occur:

```
ip ftp username zorro
ip ftp password sword
ip ftp passive
ip ftp source-interface to1
! The following command allows the core-dump code to use FTP rather than TFTP or RCP
exception protocol ftp
! The following command creates the core dump in the event the system at IP address
! 192.168.10.3 crashes
exception dump 192.168.10.3
```

1

1. To report an error in this document, please send a detailed email to bug-doc@cisco.com.



System Management

Performing Basic System Management

This chapter describes the basic tasks that you can perform to manage the general system features of the Cisco IOS software—those features that are generally not specific to a particular protocol.

This document applies to Cisco IOS Release 12.2.

For a complete description of the basic system management commands in this chapter, refer to the "Basic System Management Commands" chapter in the "Cisco IOS System Management Commands" part of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Basic System Management Task List

To customize the general functionality of your system, perform any of the tasks in the following sections. All tasks in this chapter are optional, though some, such as setting time and calendar services, are highly recommended.

- Configuring the System Name (Recommended)
- Customizing the CLI Prompt
- Creating and Displaying Command Aliases
- Controlling Minor Services (Recommended)
- Hiding Telnet Addresses
- Setting Time and Calendar Services (Recommended)
- Delaying EXEC Startup
- Handling an Idle Telnet Connection
- Setting the Interval for Load Data
- Limiting the Number of TCP Transactions
- Configuring Switching and Scheduling Priorities
- Modifying the System Buffer Size

See the end of this chapter for the "Basic System Management Examples" section.

Configuring the System Name

The most basic system management task is to assign a name to your system (router, access server, switch, and so on). The system name, also called the host name, is used to uniquely identify the system in your network. The system name is displayed at the CLI prompt. If no name is configured, the system default name is `Router`. To configure a name for your device, use the following command in global configuration mode:

Command	Purpose
`Router(config)# hostname name`	Sets the host name.

For an example of configuring a system name, see the section "System Configuration File Example" at the end of this chapter.

Customizing the CLI Prompt

By default, the CLI prompt consists of the system name followed by an angle bracket (>) for EXEC mode or a pound sign (#) for privileged EXEC mode. To customize the CLI prompt for your system, use either of the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)# prompt string`	Customizes the CLI prompt.
`Router(config)# no service prompt config`	Disables the display of the CLI prompt.

Creating and Displaying Command Aliases

Command aliases allow you to configure alternative syntax for commands. You may want to create aliases for commonly used or complex commands. For example, you could assign the alias **save config** to the **copy running-config startup-config** command to reduce the amount of typing you have to perform, or if your users might find a **save config** command easier to remember. Use word substitutions or abbreviations to tailor command syntax for you and your user community.

To create a command alias, use the following command in global configuration mode:

Command	Purpose
`Router(config)# alias mode alias-name alias-command-line`	Configures a command alias.

To display a list of command aliases currently configured on your system, and the original command syntax for those aliases, use the following command in EXEC mode:

Command	Purpose
`Router# show aliases [mode]`	Displays all command aliases and original command syntax, or displays the aliases for only a specified command mode.

Keep in mind that any aliases you configure will only be effective on your system, and that the original command syntax will appear in the configuration file.

Controlling Minor Services

The minor services are "small servers" that run on your routing device and are useful for basic system testing and for providing basic network functions. Minor services are useful for testing connections from another host on the network.

Cisco small servers are conceptually equivalent to daemons.

Small servers provided by Cisco IOS software-based devices include TCP, UDP, HTTP, BOOTP, and Finger. For information about the HTTP server, see the"Using the Cisco Web Browser User Interface" chapter in this book.

The TCP small server provides the following minor services:

- Echo—Echoes back whatever you type. To test this service, issue the **telnet** *a.b.c.d* **echo** command from a remote host.
- Chargen—Generates a stream of ASCII data. To test this service, issue the **telnet** *a.b.c.d* **chargen** command from a remote host.
- Discard—Discards whatever you type. To test this service, issue the **telnet** *a.b.c.d* **discard** command from a remote host.
- Daytime—Returns system date and time if you have configured NTP or have set the date and time manually. To test this service, issue the **telnet** *a.b.c.d* **daytime** command from a remote host.

The User Datagram Protocol (UDP) small server provides the following minor services:

- Echo—Echoes the payload of the datagram you send.
- Chargen—Discards the datagram you send and responds with a 72 character string of ASCII characters terminated with a CR+LF (carriage return and line feed).
- Discard—Silently discards the datagram you send.

To enable TCP or UDP services, use the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)# service tcp-small-servers`	Enables the minor TCP services echo, chargen, discard, and daytime.
`Router(config)# service udp-small-servers`	Enables the minor UDP services echo, chargen, and discard.

Because the minor services can be misused, these commands are disabled by default.

Caution Enabling minor services creates the potential for certain types of denial-of-service attacks, such as the UDP diagnostic port attack. Therefore, any network device that has UDP, TCP, BOOTP, or Finger services should be protected by a firewall or have the services disabled. For information on preventing UDP diagnostic port attacks, see the white paper titled *Defining Strategies to Protect Against UDP Diagnostic Port Denial of Service Attacks*, available on Cisco.com.

Note that the **no** form of the **service tcp-small-servers** and **service udp-small-servers** commands will appear in the configuration file to inform you when these basic services are disabled.

Controlling the BOOTP Server

You can enable or disable an async line Bootstrap Protocol (BOOTP) service on your routing device. This small server is enabled by default. Due to security considerations, this service should be disabled if you are not using it. To disable the BOOTP server on your platform, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no ip bootp server`	Disables the BOOTP server.

Because Dynamic Host Configuration Protocol (DHCP) is based on the Bootstrap Protocol, both of these service share the "well-known" UDP server port of 67 (per the internet standards and RFCs). For more information about DHCP configuration in Cisco IOS software, see the *Cisco IOS IP Configuration Guide*. For more information about BOOTP, see RFC 951. Interoperation between BOOTP and DHCP is defined in RFC 1534. DHCP is defined in RFC 2131.

Controlling the Finger Protocol

The Finger protocol allows users throughout the network to get a list of the users currently using a particular routing device. The information displayed includes the processes running on the system, the line number, connection name, idle time, and terminal location. This information is provided through the Cisco IOS software **show users** EXEC command.

To enable a Cisco device to respond to Finger (port 79) requests, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip finger`	Enables the Finger protocol service, which allows the system to respond to finger requests.

To configure the finger protocol to be compliant with RFC 1288, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip finger rfc-compliant`	Configures the device to wait for "Return" or "/W" input when processing Finger requests.

The **rfc-compliant** form of this command should not be configured for devices with more than 20 simultaneous users (see caveat CSCds92731 on Cisco.com for details). The difference between the two forms of this command is as follows: when the **ip finger** command is configured, the router will respond to a **telnet** *a.b.c.d* **finger** command from a remote host by immediately displaying the output of the **show users** command and then closing the connection. When the **ip finger rfc-compliant** command is configured, the router will wait for input before displaying anything. The remote user can then press the Return key to display the output of the **show users** command, or enter **/W** to display the output of the **show users wide** command. After this information is displayed, the connection is closed.

Hiding Telnet Addresses

You can hide addresses while attempting to establish a Telnet session. To configure the router to suppress Telnet addresses, use the following command in global configuration mode:

Command	Purpose
`Router(config)# service hide-telnet-address`	Hides addresses while establishing a Telnet session.

The hide feature suppresses the display of the address and continues to display all other messages that normally would be displayed during a connection attempt, such as detailed error messages if the connection failed.

Use the **busy-message** line configuration command with the **service hide-telnet-address** command to customize the information displayed during Telnet connection attempts. If the connection attempt fails, the router suppresses the address and displays the message specified with the **busy-message** command.

Setting Time and Calendar Services

All Cisco routers provide an array of time-of-day services. These services allow the products to accurately keep track of the current time and date, to synchronize multiple devices to the same time, and to provide time services to other systems. The following sections describe the concepts and task associated with time and calendar services:

- Understanding Time Sources
- Configuring NTP

- Configuring SNTP
- Configuring VINES Time Service
- Configuring Time and Date Manually
- Using the Hardware Clock
- Monitoring Time and Calendar Services
- Configuring Time Ranges

Understanding Time Sources

Most Cisco routers have two clocks: a battery-powered hardware clock (referenced in CLI commands as the "calendar") and a software clock (referenced in CLI commands as the "clock"). These two clocks are managed separately.

The primary source for time data on your system is the software clock. This clock runs from the moment the system starts up and keeps track of the current date and time. The software clock can be set from a number of sources and in turn can be used to distribute the current time through various mechanisms to other systems. When a router with a hardware clock is initialized or rebooted, the software clock is initially set based on the time in the hardware clock. The software clock can then be updated from the following sources:

- Network Time Protocol (NTP)
- Simple Network Time Protocol (SNTP)
- VINES Time Service
- Manual configuration (using the hardware clock)

Because the software clock can be dynamically updated it has the potential to be more accurate than the hardware clock.

The software clock can provide time to the following services:

- Access lists
- NTP
- VINES time service
- User **show** commands
- Logging and debugging messages
- The hardware clock

Note The software clock cannot provide time to the NTP or VINES Time Service if it was set using SNTP.

The software clock keeps track of time internally based on Coordinated Universal Time (UTC), also known as Greenwich Mean Time (GMT). You can configure information about the local time zone and summer time (daylight savings time) so that the time is displayed correctly relative to the local time zone.

The software clock keeps track of whether the time is "authoritative" (that is, whether it has been set by a time source considered to be authoritative). If it is not authoritative, the time will be available only for display purposes and will not be redistributed.

Network Time Protocol

The Network Time Protocol (NTP) is a protocol designed to time-synchronize a network of machines. NTP runs over UDP, which in turn runs over IP. NTP Version 3 is documented in RFC 1305.

An NTP network usually gets its time from an authoritative time source, such as a radio clock or an atomic clock attached to a time server. NTP then distributes this time across the network. NTP is extremely efficient; no more than one packet per minute is necessary to synchronize two machines to the accuracy of within a millisecond of one another.

NTP uses the concept of a "stratum" to describe how many NTP "hops" away a machine is from an authoritative time source. A "stratum 1" time server typically has an authoritative time source (such as a radio or atomic clock, or a GPS time source) directly attached, a "stratum 2" time server receives its time via NTP from a "stratum 1" time server, and so on.

NTP avoids synchronizing to a machine whose time may not be accurate in two ways. First, NTP will never synchronize to a machine that is not in turn synchronized itself. Second, NTP will compare the time reported by several machines, and will not synchronize to a machine whose time is significantly different than the others, even if its stratum is lower. This strategy effectively builds a self-organizing tree of NTP servers.

The Cisco implementation of NTP does not support stratum 1 service; in other words, it is not possible to connect to a radio or atomic clock (for some specific platforms, however, you can connect a GPS time-source device). We recommend that time service for your network be derived from the public NTP servers available in the IP internet.

If the network is isolated from the internet, the Cisco implementation of NTP allows a machine to be configured so that it acts as though it is synchronized via NTP, when in fact it has determined the time using other means. Other machines can then synchronize to that machine via NTP.

A number of manufacturers include NTP software for their host systems, and a publicly available version for systems running UNIX and its various derivatives is also available. This software also allows UNIX-derivative servers to acquire the time directly from an atomic clock which would subsequently propagate time information along to Cisco routers.

The communications between machines running NTP (known as "associations") are usually statically configured; each machine is given the IP address of all machines with which it should form associations. Accurate timekeeping is made possible by exchanging NTP messages between each pair of machines with an association.

However, in a LAN environment, NTP can be configured to use IP broadcast messages instead. This alternative reduces configuration complexity because each machine can simply be configured to send or receive broadcast messages. However, the accuracy of timekeeping is marginally reduced because the information flow is one-way only.

The time kept on a machine is a critical resource, so we strongly recommend that you use the security features of NTP to avoid the accidental or malicious setting of incorrect time. Two mechanisms are available: an access list-based restriction scheme and an encrypted authentication mechanism.

When multiple sources of time (VINES, hardware clock, manual configuration) are available, NTP is always considered to be more authoritative. NTP time overrides the time set by any other method.

Simple Network Time Protocol

Simple Network Time Protocol (SNTP) is a simplified, client-only version of NTP for use on Cisco 1003, Cisco 1004, Cisco 1005, Cisco 1600, Cisco 1720, and Cisco 1750 routers. SNTP can receive only the time from NTP servers; it cannot be used to provide time services to other systems.

SNTP typically provides time within 100 milliseconds of the accurate time, but it does not provide the complex filtering and statistical mechanisms of NTP. In addition, SNTP does not authenticate traffic, although you can configure extended access lists to provide some protection. An SNTP client is more vulnerable to misbehaving servers than an NTP client and should be used only in situations where strong authentication is not required.

You can configure SNTP to request and accept packets from configured servers or to accept NTP broadcast packets from any source. When multiple sources are sending NTP packets, the server with the best stratum is selected. (See the "Network Time Protocol" section for a description of strata.) If multiple servers are at the same stratum, a configured server is preferred over a broadcast server. If multiple servers pass both tests, the first one to send a time packet is selected. SNTP will choose a new server only if it stops receiving packets from the currently selected server, or if a better server (according to the above criteria) is discovered.

VINES Time Service

Time service is available when Banyan VINES is configured. This protocol is a standard part of VINES. The Cisco implementation allows the VINES time service to be used in two ways. First, if the system has learned the time from some other source, it can act as a VINES time server and provide time to other machines running VINES. Second, it can use the VINES time service to set the software clock if no other form of time service is available.

Note Support for Banyan VINES and XNS is removed from Cisco IOS software in Cisco IOS Release 12.2(13)T and later.

Hardware Clock

Some routers contain a battery-powered hardware clock that tracks the date and time across system restarts and power outages. The hardware clock is always used to initialize the software clock when the system is restarted.

Note Within the CLI command syntax, the hardware clock is referred to as the "system calendar."

If no other source is available, the hardware clock can be considered to be an authoritative source of time and be redistributed via NTP or VINES time service. If NTP is running, the hardware clock can be updated periodically from NTP, compensating for the inherent drift in the hardware clock.

Configuring NTP

NTP services are disabled on all interfaces by default. The following sections contain optional tasks that you can perform on your networking device:

- Configuring Poll-Based NTP Associations
- Configuring Broadcast-Based NTP Associations
- Configuring an NTP Access Group
- Configuring NTP Authentication
- Disabling NTP Services on a Specific Interface

- Configuring the Source IP Address for NTP Packets
- Configuring the System as an Authoritative NTP Server
- Updating the Hardware Clock
- Configuring an External Reference Clock

Configuring Poll-Based NTP Associations

Networking devices running NTP can be configured to operate in variety of association modes when synchronizing time with reference time sources. There are two ways that a networking device can obtain time information on a network: by polling host servers and by listening to NTP broadcasts. In this section, we will focus on the poll-based association modes. Broadcast-based NTP associations will be discussed in the next section.

The following are two most commonly used, poll-based association modes:

- Client mode
- Symmetric active mode

The *client* and the *symmetric active* modes should be used when NTP is required to provide a high level of time accuracy and reliability.

When a networking device is operating in the *client mode*, it polls its assigned time serving hosts for the current time. The networking device will then pick a host from all the polled time servers to synchronize with. Since the relationship that is established in this case is a client-host relationship, the host will not capture or use any time information sent by the local client device. This mode is most suited for file-server and workstation clients that are not required to provide any form of time synchronization to other local clients. Use the **ntp server** command to individually specify the time serving hosts that you want your networking device to consider synchronizing with and to set your networking device to operate in the *client mode*.

When a networking device is operating in the *symmetric active mode*, it polls its assigned time serving hosts for the current time and it responds to polls by its hosts. Since this is a peer-to-peer relationship, the host will also retain time-related information about the local networking device that it is communicating with. This mode should be used when there is a number of mutually redundant servers that are interconnected via diverse network paths. Most Stratum 1 and stratum 2 servers on the Internet today adopt this form of network setup. Use the **ntp peer** command to individually specify the time serving hosts that you want your networking device to consider synchronizing with and to set your networking device to operate in the *symmetric active mode*.

The specific mode that you should set each of your networking devices to depends primarily on the role that you want it to assume as a timekeeping device (server or client) and its proximity to a stratum 1 timekeeping server.

A networking device engages in polling when it is operating as a client or a host in the *client mode* or when it is acting as a peer in the *symmetric active mode*. Although polling does not usually exact a toll on memory and CPU resources such as bandwidth, an exceedingly large number of ongoing and simultaneous polls on a system can seriously impact the performance of a system or slow the performance of a given network. To avoid having an excessive number of ongoing polls on a network, you should limit the number of direct, peer-to-peer or client-to-server associations. Instead, you should consider using NTP broadcasts to propagate time information within a localized network.

Command	Purpose
`Router(config)# ntp peer ip-address [normal-sync] [version number] [key keyid] [source interface] [prefer]`	Forms a peer association with another system.
`Router(config)# ntp server ip-address [version number] [key keyid] [source interface] [prefer]`	Forms a server association with another system.

Note that only one end of an association needs to be configured; the other system will automatically establish the association.

Caution The **ntp clock-period** command is automatically generated to reflect the constantly changing *correction factor* when the **copy running-configuration startup-configuration** command is entered to save the configuration to NVRAM. Do not attempt to manually use the **ntp clock-period** command. Ensure that you remove this command line when copying configuration files to other devices.

For an example of configuring an NTP server-peer relationship, see the "Clock, Calendar, and NTP Configuration Examples" section at the end of this chapter.

Configuring Broadcast-Based NTP Associations

Broadcast-based NTP associations should be used when time accuracy and reliability requirements are modest and if your network is localized and has a large number of clients (more than 20). Broadcast-based NTP associations is also recommended for use on networks that have limited bandwidth, system memory, or CPU resources.

When a networking device is operating in the *broadcastclient mode*, it does not engage in any polling. Instead, it listens for NTP broadcast packets transmitted by broadcast time servers. Consequently, time accuracy can be marginally reduced since time information flows only one way.

Use the **ntp broadcast client** command to set your networking device to listen for NTP broadcast packets propagated through a network. In order for *broadcastclient mode* to work, the broadcast server and its clients must be located on the same subnet. The time server that is transmitting NTP broadcast packets will also have to be enabled on the interface of the given device using the **ntp broadcast** command.

To configure an interface to send NTP broadcasts, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ntp broadcast [version number]`	Configures the specified interface to send NTP broadcast packets.

To configure an interface to receive NTP broadcasts, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ntp broadcast client`	Configures the specified interface to receive NTP broadcast packets.

To manually set the estimated round-trip delay between the device and the NTP broadcast server, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ntp broadcastdelay` *microseconds*	Adjusts the estimated round-trip delay for NTP broadcasts.

Caution The **ntp clock-period** command is automatically generated to reflect the constantly changing *correction factor* when the **copy running-configuration startup-configuration** command is entered to save the configuration to NVRAM. Do not attempt to manually use the **ntp clock-period** command. Ensure that you remove this command line when copying configuration files to other devices.

For an example of configuring broadcast-based NTP associations, see the "Clock, Calendar, and NTP Configuration Examples" section at the end of this chapter.

Configuring an NTP Access Group

The access list-based restriction scheme allows you to grant or deny certain access privileges to an entire network, a subnet within a network, or a host within a subnet. To define an NTP access group, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ntp access-group {query-only | serve-only | serve | peer}` *access-list-number*	Creates an access group and applies a basic IP access list to it.

The access group options are scanned in the following order, from least restrictive to most restrictive:

1. **peer**—Allows time requests and NTP control queries and allows the system to synchronize itself to a system whose address passes the access list criteria.
2. **serve**—Allows time requests and NTP control queries, but does not allow the system to synchronize itself to a system whose address passes the access list criteria.
3. **serve-only**—Allows only time requests from a system whose address passes the access list criteria.
4. **query-only**—Allows only NTP control queries from a system whose address passes the access list criteria.

If the source IP address matches the access lists for more than one access type, the first type is granted. If no access groups are specified, all access types are granted to all systems. If any access groups are specified, only the specified access types will be granted.

For details on NTP control queries, see RFC 1305 (NTP version 3).

Configuring NTP Authentication

The encrypted NTP authentication scheme should be used when a reliable form of access control is required. Unlike the access list-based restriction scheme which is based on IP addresses, the encrypted authentication scheme uses authentication keys and an authentication process to determine if NTP synchronization packets sent by designated peers or servers on a local network are deemed as trusted before the time information that it carries along with it, is accepted.

The authentication process begins from the moment an NTP packet is created. Cryptographic checksum keys are generated using the MD5 Message Digest Algorithm and are embedded into the NTP synchronization packet that is sent to a receiving client. Once a packet is received by a client, its cryptographic checksum key is decrypted and checked against a list of trusted keys. If the packet contains a matching authenticator key, the timestamp information that is contained within it is accepted by the receiving client. NTP synchronization packets that do not contain a matching authenticator key will be ignored.

It is important to note that the encryption and decryption processes used in NTP authentication can be very CPU-intensive and can seriously degrade the accuracy of the time that is propagated within a network. If your network setup permits a more comprehensive model of access control, you should consider the use of the access list-based form of control instead.

After NTP authentication is properly configured, your networking device will only synchronize with and provide synchronization to trusted time sources. To enable your networking device to send and receive encrypted synchronization packets, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ntp authenticate`	Enables the NTP authentication feature.
Step 2	`Router(config)# ntp authentication-key` *number* `md5` *value*	Defines the authentication keys. Each key has a key number, a type, and a value. Currently the only key type supported is **md5**.
Step 3	`Router(config)# ntp trusted-key` *key-number*	Defines trusted authentication keys. If a key is trusted, this system will be ready to synchronize to a system that uses this key in its NTP packets.

Note In Cisco IOS software versions previous to release 12.0, the cryptotype value is displayed along with the ntp authentication key md5 value when the **show running-configuration** command is entered. Avoid copying and pasting the string cryptotype value that is displayed with the authentication-key as it will result in authentication failure.

Disabling NTP Services on a Specific Interface

NTP services are disabled on all interfaces by default.

NTP is enabled globally when any NTP commands are entered. you can selectively prevent NTP packets from being received through a specific interface by using the following command in interface configuration mode to turn off NTP on a given interface:

Command	Purpose
`Router(config-if)# ntp disable`	Disables NTP services on a specific interface.

Configuring the Source IP Address for NTP Packets

When the system sends an NTP packet, the source IP address is normally set to the address of the interface through which the NTP packet is sent. Use the following command in global configuration mode if you want to configure a specific interface from which the IP source address will be taken:

Command	Purpose
`Router(config)# ntp source` *interface*	Configures an interface from which the IP source address will be taken.

This interface will be used for the source address for all packets sent to all destinations. If a source address is to be used for a specific association, use the **source** parameter on the **ntp peer** or **ntp server** command shown earlier in this chapter.

Configuring the System as an Authoritative NTP Server

Use the following command in global configuration mode if you want the system to be an authoritative NTP server, even if the system is not synchronized to an outside time source:

Command	Purpose
`Router(config)# ntp master` [*stratum*]	Makes the system an authoritative NTP server.

Note Use the **ntp master** command with caution. It is very easy to override valid time sources using this command, especially if a low stratum number is configured. Configuring multiple machines in the same network with the **ntp master** command can cause instability in timekeeping if the machines do not agree on the time.

For an example of configuring an authoritative NTP server, see the "Clock, Calendar, and NTP Configuration Examples" section at the end of this chapter.

Updating the Hardware Clock

On devices that have hardware clocks (system calendars), you can configure the hardware clock to be periodically updated from the software clock. This is advisable for any device using NTP, because the time and date on the software clock (set using NTP) will be more accurate than the hardware clock, because the time setting on the hardware clock has the potential to drift slightly over time.

Use the following command in global configuration mode if a routing device is synchronized to an outside time source via NTP and you want the hardware clock to be synchronized to NTP time:

Command	Purpose
`Router(config)# ntp update-calendar`	Configures the system to update its hardware clock from the software clock at periodic intervals.

For an example of configuring NTP to update the calendar, see the section "Clock, Calendar, and NTP Configuration Examples" at the end of this chapter.

Configuring an External Reference Clock

Because Cisco's implementation of NTP does not support stratum 1 service, it is not possible to connect to a radio or atomic clock (for some specific platforms however, you can connect a GPS timesource device). However, certain Cisco devices allow you to connect a external GPS-based time-source device for the purposes of distributing a time signal to your network using NTP.

For example, the Trimble Palisade NTP Synchronization Kit can be connected to the auxiliary port of a Cisco 7200 Series router. Also, selected platforms support the use of GPS clocks from Symmetricom (formerly Telecom-Solutions). The refclock (reference clock) drivers provided on these platforms provides the ability to receive an RTS time-stamp signal on the auxiliary port of your routing device.

To configure a Trimble Palisade GPS product connected to the auxiliary port of a Cisco 7200 series router as the NTP reference clock, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# line aux 0`	Enters line configuration mode for the auxiliary port 0.
Step 2	`Router(config-line)# ntp refclock trimble pps none stratum 1`	Enables the driver that allows the Trimble Palisade NTP Synchronization Kit to be used as the NTP reference clock source (Cisco 7200 series routers only).

To configure a Symmetricom GPS product connected to the auxiliary port of a supported router or switch as the NTP reference clock, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# line aux 0`	Enters line configuration mode for the auxiliary port zero.
Step 2	`Router(config-line)# ntp refclock telecom-solutions pps cts stratum 1`	Enables the driver that allows the Symmetricom GPS product to be used as the NTP reference clock source.

To configure a PPS signal as the source for NTP synchronization, use the following form of the **ntp refclock** command in line configuration mode:

Command	Purpose
`Router(config-line)# ntp refclock pps {cts | ri} [inverted] [pps-offset number] [stratum number] [timestamp-offset number]`	Configures a PPS signal as the source for NTP synchronization.

Verifying the Status of the External Reference Clock

To verify the status of NTP components, use the following commands in privileged EXEC mode:

Command	Purpose
`Router# show ntp associations`	Displays the status of NTP associations, including the status of the GPS reference clock.
`Router# show ntp status`	Displays the status of NTP.
`Router# debug ntp refclock`	Allows advanced monitoring of reference clock activities for the purposes of debugging.

Configuring SNTP

SNTP generally is supported on those platforms that do not provide support for NTP, such as the Cisco 1000 series, 1600 series, and 1700 series platforms. SNTP is disabled by default. In order to enable SNTP, use one or both of the following commands in global configuration mode:

Command	Purpose
`Router(config)# sntp server {address | hostname} [version number]`	Configures SNTP to request NTP packets from an NTP server.
`Router(config)# sntp broadcast client`	Configures SNTP to accept NTP packets from any NTP broadcast server.

Enter the **sntp server** command once for each NTP server. The NTP servers must be configured to respond to the SNTP messages from the router.

If you enter both the **sntp server** command and the **sntp broadcast client** command, the router will accept time from a broadcast server but prefer time from a configured server, assuming that the strata are equal. To display information about SNTP, use the **show sntp** EXEC command.

Configuring VINES Time Service

Note Support for Banyan VINES and XNS has been removed from Cisco IOS software, beginning in Cisco IOS Release 12.2(13)T. The following VINES commands are not available in releases derived from 12.2(13)T, such as the 12.3 mainline release.

To distribute the system time and date to other devices on the network using VINES time services, use the following command in global configuration mode:

Command	Purpose
`Router(config)# vines time use-system`	Distributes the system software clock time to other VINES systems.

To set the system time and date from received VINES time services, use the following command in global configuration mode:

Command	Purpose
`Router(config)# vines time set-system`	Sets the software clock system time from received VINES time services.

Configuring Time and Date Manually

If no other source of time is available, you can manually configure the current time and date after the system is restarted. The time will remain accurate until the next system restart. We recommend that you use manual configuration only as a last resort.

To set up time services, complete the tasks in the following sections as needed. If you have an outside source to which the router can synchronize, you do not need to manually set the software clock.

- Configuring the Time Zone
- Configuring Summer Time (Daylight Savings Time)
- Manually Setting the Software Clock
- Using the Hardware Clock

Configuring the Time Zone

To manually configure the time zone used by the Cisco IOS software, use the following command in global configuration mode :

Command	Purpose
`Router(config)# clock timezone` *zone hours-offset* [*minutes-offset*]	Sets the time zone. The *zone* argument is the name of the time zone (typically a standard acronym). The *hours-offset* argument is the number of hours the time zone is different from UTC. The *minutes-offset* argument is the number of minutes the time zone is different from UTC.

Tips The *minutes-offset* argument of the **clock timezone** command is available for those cases where a local time zone is a percentage of an hour different from UTC/GMT. For example, the time zone for some sections of Atlantic Canada (AST) is UTC -3.5. In this case, the necessary command would be **clock timezone AST -3 30**.

For an example of configuring the time zone, see the section "Clock, Calendar, and NTP Configuration Examples" at the end of this chapter.

Configuring Summer Time (Daylight Savings Time)

To configure summer time (daylight savings time) in areas where it starts and ends on a particular day of the week each year, use the following command in global configuration mode:

Command	Purpose
`Router(config)# clock summer-time zone recurring [week day month hh:mm week day month hh:mm [offset]]`	Configures a recurring summer time start and end date. The *offset* argument is used to indicate the number of minutes to add to the clock during summer time.

If summer time in your area does not follow this pattern, you can configure the exact date and time of the next summer time event by using one of the following commands in global configuration mode:

Command	Purpose

`Router(config)# clock summer-time zone date month date year hh:mm month date year hh:mm [offset]`
or
`Router(config)# clock summer-time zone date date month year hh:mm date month year hh:mm [offset]`
	Configures a specific summer time start and end date. The *offset* argument is used to indicate the number of minutes to add to the clock during summer time.

For an example of configuring summer time, see the section "Clock, Calendar, and NTP Configuration Examples" at the end of this chapter.

Manually Setting the Software Clock

Generally, if the system is synchronized by a valid outside timing mechanism, such as an NTP or VINES clock source, or if you have a router with a hardware clock, you need not set the software clock. Use this command if no other time sources are available. The time specified in this command is relative to the configured time zone. To set the software clock manually, use the following command in privileged EXEC mode:

Command	Purpose

`Router# clock set hh:mm:ss date month year`
or
`Router# clock set hh:mm:ss month date year`
	Sets the software clock.

Using the Hardware Clock

Most Cisco devices have a separate hardware-based clock in addition to the software-based clock. The hardware clock is a chip with a rechargeable backup battery that can retain the time and date information across reboots of the device.

To maintain the most accurate time update from an authoritative time source on the network, the software clock should receive time updates from an authoritative time on the network. The hardware clock should in turn be updated at regular intervals from the software clock while the system is running.

To customize the use of the hardware clock on your system, perform any of the following optional tasks:

- Setting the Hardware Clock
- Configuring the Router as a Network Time Source

- Setting the Software Clock from the Hardware Clock
- Setting the Hardware Clock from the Software Clock

Setting the Hardware Clock

The hardware clock (system calendar) maintains time separately from the software clock. The hardware clock continues to run when the system is restarted or when the power is turned off. Typically, the hardware clock needs to be manually set only once, when the system is first installed.

You should avoid setting the hardware clock manually if you have access to a reliable external time source. Time synchronization should instead be established using NTP.

If you do not have access to an external time source, use one of the forms of the following command in EXEC mode to set the hardware clock:

Command	Purpose

`Router> calendar set hh:mm:ss day month year`
or
`Router> calendar set hh:mm:ss month day year`
	Sets the hardware clock manually.

Configuring the Router as a Network Time Source

By default, the time maintained on the software clock is not considered to be authoritative and will not be redistributed with NTP or VINES Time Service. To classify the hardware clock as authoritative, use the following command in global configuration mode:

Command	Purpose
`Router(config)# clock calendar-valid`	Enables the router to act as a valid time source to which network peers can synchronize.

For an example of making the hardware clock authoritative, see the "Clock, Calendar, and NTP Configuration Examples" section at the end of this chapter.

Setting the Software Clock from the Hardware Clock

To set the software clock to the new hardware clock setting, use the following command in EXEC mode:

Command	Purpose
`Router# clock read-calendar`	Sets the software clock from the hardware clock.

Setting the Hardware Clock from the Software Clock

To update the hardware clock with a new software clock setting, use the following command in EXEC mode:

Command	Purpose
`Router# clock update-calendar`	Sets the hardware clock from the software clock.

Monitoring Time and Calendar Services

To monitor clock, calendar, and NTP EXEC services, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show calendar`	Displays the current hardware clock time.
`Router# show clock [detail]`	Displays the current software clock time.
`Router# show ntp associations [detail]`	Displays the status of NTP associations.
`Router# show ntp status`	Displays the status of NTP.
`Router# show sntp`	Displays information about SNTP (Cisco 1003, Cisco 1004, Cisco 1005, Cisco 1600, Cisco 1720, or Cisco 1750 routers only).

Configuring Time Ranges

Cisco IOS allows implementation of features based on the time of day. The **time-range** global configuration command defines specific times of the day and week, which then can be referenced by a function, so that those time restrictions are imposed on the function itself.

In Cisco IOS Release 12.2, IP and IPX extended access lists are the only functions that can use time ranges. The time range allows the network administrator to define when the **permit** or **deny** statements in the access list are in effect. Prior to the introduction of this feature, access list statements were always in effect once they were applied. Both named or numbered access lists can reference a time range.

Benefits of time ranges include the following:

- The network administrator has more control over permitting or denying a user access to resources. These resources could be an application (identified by an IP address/mask pair and a port number), policy routing, or an on-demand link (identified as interesting traffic to the dialer).
- Network administrators can set a time-based security policy, including the following:
  - Perimeter security using the Cisco IOS Firewall feature set or access lists
  - Data confidentiality with Cisco Encryption Technology or IPSec
- Policy-based routing and queueing functions are enhanced.
- When provider access rates vary by time of day, it is possible to automatically reroute traffic cost effectively.
- Service providers can dynamically change a committed access rate (CAR) configuration to support the quality of service (QoS) service level agreements (SLAs) that are negotiated for certain times of day.
- Network administrators can control logging messages. Access list entries can log traffic at certain times of the day, but not constantly. Therefore, administrators can simply deny access without needing to analyze many logs generated during peak hours.

Defining a Time Range

Note The time range relies on the system's software clock. For the time range feature to work the way you intend, you need a reliable clock source. We recommend that you use NTP to synchronize the system's software clock.

To define a time range, use the following commands beginning in global configuration mode.

	Command	Purpose
Step 1	`Router(config)# time-range time-range-name`	Assigns a name to the time range to be configured and enters time-range configuration mode.

Step 2	`Router(config-time-range)# absolute [start time date] [end time date]`
	or
	`Router(config-time-range)# periodic days-of-the-week hh:mm to [days-of-the-week] hh:mm`
		Specifies when the time range will be in effect. Use some combination of these commands; multiple **periodic** statements are allowed; only one **absolute** statement is allowed.

Repeat these tasks if you have multiple items you want in effect at different times. For example, repeat the steps to include multiple **permit** or **deny** statements in an access list in effect at different times. For more information about these commands, refer to the "Basic System Management Commands" chapter in the "Cisco IOS System Management Commands" part of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*.

Referencing the Time Range

In order for a time range to be applied, you must reference it by name in a feature that can implement time ranges. You can reference the time range in the following Cisco IOS software features:

- IP Extended Access Lists
  - Refer to the "Configuring IP Services" chapter of the Release 12.2 *Cisco IOS IP Configuration Guide* for instructions on creating an IP Extended Access List and referencing a time range.
- IPX Extended Access Lists
  - Refer to the "Configuring Novell IPX" chapter of the Release 12.2 *Cisco IOS AppleTalk and Novell IPX Configuration Guide* for instructions on creating an IPX Extended Access List and referencing a time range.

Delaying EXEC Startup

To delay the startup of the EXEC process on noisy lines until the line has been idle for 3 seconds, use the following command in global configuration mode:

Command	Purpose
`Router(config)# service exec-wait`	Delays startup of the EXEC.

This command is useful on noisy modem lines or when a modem attached to the line is configured to ignore MNP or V.42 negotiations, and MNP or V.42 modems may be dialing in. In these cases, noise or MNP/V.42 packets might be interpreted as usernames and passwords, causing authentication failure before the user can type a username or password. The command is not useful on nonmodem lines or lines without some kind of login configured.

Handling an Idle Telnet Connection

To configure the Cisco IOS software to set the TCP window to zero (0) when the Telnet connection is idle, use the following command in global configuration mode:

Command	Purpose
`Router(config)# service telnet-zero-idle`	Sets the TCP window to zero when the Telnet connection is idle.

Normally, data sent to noncurrent Telnet connections is accepted and discarded. When **service telnet-zero-idle** is enabled, if a session is suspended (that is, some other connection is made active), the TCP window is set to zero. This action prevents the remote host from sending any more data until the connection is resumed. Use this command when it is important that all messages sent by the host be seen by the users and the users are likely to use multiple sessions. Do not use this command if your host will eventually time out and log out a TCP user whose window is zero.

Setting the Interval for Load Data

You can change the period of time over which a set of data is used for computing load statistics. Decisions, such as for dial backup, depend on these statistics. If you decrease the load interval, the average statistics are computed over a shorter period of time and are more responsive to bursts of traffic.

To change the length of time for which a set of data is used to compute load statistics, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# load-interval` *seconds*	Sets the length of time for which data is used for load calculations.

Limiting the Number of TCP Transactions

When using a standard TCP implementation to send keystrokes between machines, TCP tends to send one packet for each keystroke typed, which can use up bandwidth and contribute to congestion on larger networks.

John Nagle's algorithm (RFC 896) helps alleviate the small-packet problem in TCP. The first character typed after connection establishment is sent in a single packet, but TCP holds any additional characters typed until the receiver acknowledges the previous packet. Then the second, larger packet is sent, and

additional typed characters are saved until the acknowledgment comes back. The effect is to accumulate characters into larger chunks, and pace their transmission to the network at a rate matching the round-trip time of the given connection. This method is usually preferable for all TCP-based traffic.

By default, the Nagle algorithm is not enabled. To enable the Nagle algorithm and thereby reduce the number of TCP transactions, use the following command in global configuration mode:

Command	Purpose
`Router(config)# service nagle`	Enables the Nagle slow packet avoidance algorithm.

Configuring Switching and Scheduling Priorities

The normal operation of the network server allows the switching operations to use as much of the central processor as is required. If the network is running unusually heavy loads that do not allow the processor the time to handle the routing protocols, you may need to give priority to the system process scheduler. To do so, use the following command in global configuration mode:

Command	Purpose
`Router(config)# scheduler interval` *milliseconds*	Defines the maximum amount of time that can elapse without running the lowest-priority system processes.

To change the amount of time that the CPU spends on fast-switching and process-level operations on the Cisco 7200 series and Cisco 7500 series routers, use the following command in global configuration mode:

Command	Purpose
`Router(config)# scheduler allocate` *network-microseconds process-microseconds*	For the Cisco 7200 series and Cisco 7500 series routers, changes the default time the CPU spends on process tasks and fast switching.

Caution We recommend that you do not change the default values of the **scheduler allocate** command.

To configure the characteristics for a looping process, use the following command in global configuration mode:

Command	Purpose
`Router(config)# scheduler process-watchdog {hang | normal | reload | terminate}`	Configures an action for a looping process.

Modifying the System Buffer Size

You can adjust initial buffer pool settings and the limits at which temporary buffers are created and destroyed. To do so, use the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)# buffers {small | middle | big | verybig | large | huge | type number} {permanent | max-free | min-free | initial} number`	Adjusts the system buffer sizes.
`Router(config)# buffers huge size number`	Dynamically resizes all huge buffers to the value that you supply.

Caution Normally you need not adjust these parameters; do so only after consulting with technical support personnel. Improper settings can adversely impact system performance.

During normal system operation, there are two sets of buffer pools: public and interface. They behave as follows:

- The buffers in the public pools grow and shrink based upon demand. Some public pools are temporary and are created and destroyed as needed. Other public pools are permanently allocated and cannot be destroyed. Public buffer pools are labeled as small, middle, big, large, very big, and huge.
- Interface pools are static—that is, they are all permanent. One interface pool exists for each interface. For example, a Cisco 4000 1E 4T configuration has one Ethernet buffer pool and four serial buffer pools. In the **buffers** EXEC command, the *type* and *number* arguments allow the user to tune the interface pools.

See the section "Buffer Modification Examples" at the end of this chapter for more information.

The server has one pool of queueing elements and six public pools of packet buffers of different sizes. For each pool, the server keeps count of the number of buffers outstanding, the number of buffers in the free list, and the maximum number of buffers allowed in the free list. To display statistics about the buffer pool on the system, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router> show buffers`	Displays all public pool information.
`Router> show buffers address hex-addr`	Displays buffer information for an address.
`Router> show buffers all [dump | header | packet]`	Displays all public and interface pool information.
`Router> show buffers assigned [dump | header | packet]`	Displays a listing of all buffers in use.
`Router> show buffers failures [dump | header | packet]`	Displays buffer allocation failures.
`Router> show buffers free [dump | header | packet]`	Displays buffers available for use.
`Router> show buffers old [dump | header | packet]`	Displays buffers older than one minute.
`Router> show buffers input-interface interface-type identifier`	Displays buffer information for an input interface.
`Router> show buffers pool pool name`	Displays all interface pool information.

Basic System Management Examples

This section provides the following system management examples:

- System Configuration File Example
- Clock, Calendar, and NTP Configuration Examples
- Buffer Modification Examples

System Configuration File Example

The following is an example of a typical system configuration file:

```
! Define line password
line 0 4
 password secret
 login
!
! Define privileged-level password
enable-password Secret Word
!
! Define a system hostname
hostname TIP
! Specify a configuration file to load at system startup
boot host host1-confg 192.168.1.111
boot host host2-confg 192.168.1.111
! Specify the system image to boot at startup
boot system sys1-system 192.168.13.111
boot system sys2-system 192.168.1.111
boot system rom
!
! Enable SNMP
snmp-server community red
snmp-server enable traps snmp authentication
snmp-server host 192.168.1.27 public
snmp-server host 192.168.1.111 public
snmp-server host 192.168.2.63 public
!
! Define TACACS server hosts
tacacs-server host 192.168.1.27
tacacs-server host 192.168.13.33
tacacs-server host 192.168.1.33
!
! Define a message-of-the-day banner
banner motd ^C
The Information Place welcomes you

Please call 1-800-555-2222 for a login account, or enter
your password at the prompt.
^C
```

Clock, Calendar, and NTP Configuration Examples

In the following example, a router with a hardware clock has server associations with two other systems, sends broadcast NTP packets, periodically updates the hardware clock, and redistributes time into VINES:

```
clock timezone PST -8
```

```
clock summer-time PDT recurring
ntp update-calendar
ntp server 192.168.13.57
ntp server 192.168.11.58
interface Ethernet 0/0
 ntp broadcast
vines time use-system
```

In the following example, a router with a hardware clock has no outside time source, so it uses the hardware clock as an authoritative time source and distributes the time via NTP broadcast packets:

```
clock timezone MET 2
clock calendar-valid
ntp master
interface fddi 0/0
 ntp broadcast
```

Buffer Modification Examples

The following example instructs the system to keep at least 50 small buffers free:

```
Router> buffers small min-free 50
```

The following example instructs the system to keep no more than 200 middle buffers free:

```
Router> buffers middle max-free 200
```

The following example instructs the system to create one large temporary extra buffer, just after a reload:

```
Router> buffers large initial 1
```

The following example instructs the system to create one permanent huge buffer:

```
Router> buffers huge permanent 1
```

Troubleshooting and Fault Management

This chapter describes basic tasks that you can perform to troubleshoot your system and the network. For detailed troubleshooting procedures and scenarios, refer to the *Internetwork Troubleshooting Guide*. For complete details on all **debug** commands, refer to the *Cisco IOS Debug Command Reference*.

For a complete description of the troubleshooting commands in this chapter, refer to the "Troubleshooting and Fault Management Commands" chapter in "Cisco IOS System Management Commands" part of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Troubleshooting and Fault Management Task List

To manage network faults, you need to discover, isolate, and correct problems. You can discover problems with the system monitoring commands, isolate problems with the system test commands, and resolve problems with other commands, including **debug** commands.

To perform general fault management, perform the tasks described in the following sections:

- Displaying System Information Using show Commands
- Testing Network Connectivity
- Testing Memory and Interfaces
- Logging System Messages
- Using Field Diagnostics on Line Cards
- Troubleshooting Specific Line Cards
- Storing Line Card Crash Information
- Creating Core Dumps for System Exceptions
- Enabling Debug Operations
- Enabling Conditionally Triggered Debugging
- Using the Environmental Monitor

In addition to the material presented in this chapter, many chapters in the Cisco IOS software configuration guides include fault management tasks specific to certain technologies and features. You can find these tasks in the "Monitoring and Maintaining" sections.

Displaying System Information Using show Commands

To provide information about system processes, the Cisco IOS software includes an extensive list of EXEC commands that begin with the word **show**, which, when executed, display detailed tables of system information. Following is a partial list of system management **show** commands. To display the information described, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# **show c2600**	Displays information about the Cisco 2600 platform, including interrupts, IOS Priority Masks, and IDMA status, for troubleshooting.
Router# **show c7200**	Displays information about the CPU and midplane for the Cisco 7200 series routers.
Router# **show context**	Displays information stored in NVRAM when the router crashes. This command is only useful to your technical support representative. This command is supported on the Cisco 2600 and 7000 series routers.
Router# **show controllers**	Displays information specific to the hardware on a line card.
Router# **show controllers logging**	Displays logging information about a line card.
Router# **show controllers tech-support**	Displays general information about a line for use when reporting a problem.
Router# **show controllers vip** *slot-number* **tech-support**	Displays information about the Versatile Interface Processor (VIP) card for use when reporting a problem
Router# **show diag**	Displays hardware information (including DRAM and static RAM details) for line cards.
Router# **show environment** [**all** | **last** | **table**]	Displays a message indicating whether an environmental warning condition currently exists, the temperature and voltage information, the last measured value from each of the six test points stored in nonvolatile memory, or environmental specifications. Examples of systems that support this command include the Cisco 7000 and the Cisco 12000 series routers.
Router# **show gsr**	Displays hardware information on the Cisco 12000 series Gigabit Switch Router (GSR).
Router# **show gt64010**	Displays all GT64010 internal registers and interrupt status on the Cisco 7200 series routers.
Router# **show memory** [*memory-type*] [**free**] [**summary**]	Displays memory pool statistics including summary information about the activities of the system memory allocator and a block-by-block listing of memory use.

Command	Purpose
Router# **show pci** {**hardware** | **bridge** [*register*]}	Displays information about the peripheral component interconnect (PCI) hardware registers or bridge registers for the Cisco 2600 and 7000 series routers.
Router# **show processes** [**cpu**]	Displays information about all active processes.
Router# **show processes memory**	Displays information about memory usage.
Router# **show protocols**	Displays the configured protocols.
Router# **show stacks**	Displays stack usage of processes and interrupt routines, including the reason for the last system reboot. This command is only useful to your technical support representative.
Router# **show subsys** [**class** *class* | **name** *name*]	Displays subsystem information.
Router# **show tcp** [*line-number*]	Displays the status of TCP connections.
Router# **show tcp brief** [**all**]	Displays a concise description of TCP connection endpoints.
Router# **show tdm connections** [**motherboard** | **slot** *number*]	Displays a snapshot of the time-division multiplexing (TDM) bus connection or data memory in a Cisco AS5200 access server.
Router# **show tech-support** [**page**] [**password**]	Displays information about the system for use when reporting a problem.

Refer to specific **show** commands in the tables of configuration commands found throughout the chapters in Cisco IOS software configuration guides. Refer to the Cisco IOS software command reference publications for detailed descriptions of the commands.

Testing Network Connectivity

To test basic network connectivity, perform the tasks described in the following sections:

- Configuring the TCP Keepalive Packet Service
- Testing Connections with the ping Command
- Tracing Packet Routes

Configuring the TCP Keepalive Packet Service

The TCP keepalive capability allows a router to detect when the host with which it is communicating experiences a system failure, even if data stops being sent (in either direction). This capability is most useful on incoming connections. For example, if a host failure occurs while the router is communicating with a printer, the router might never notice, because the printer does not generate any traffic in the opposite direction. If keepalives are enabled, they are sent once every minute on otherwise idle connections. If 5 minutes pass and no keepalives are detected, the connection is closed. The connection is also closed if the host replies to a keepalive packet with a reset packet. This will happen if the host crashes and comes back up again.

To generate the TCP keepalive packet service, use the following command in global configuration mode:

Command	Purposes
`Router(config)# service {tcp-keepalives-in | tcp-keepalives-out}`	Generates TCP keepalive packets on idle network connections, either incoming connections initiated by a remote host, or outgoing connections initiated by a user.

Testing Connections with the ping Command

As an aid to diagnosing basic network connectivity, many network protocols support an echo protocol. The protocol involves sending a special datagram to the destination host, then waiting for a reply datagram from that host. Results from this echo protocol can help in evaluating the path-to-host reliability, delays over the path, and whether the host can be reached or is functioning.

To invoke the echo protocol, use the following command in either user or privileged EXEC mode:

Command	Purposes
`Router# ping [protocol] {host | address}`	Invokes a diagnostic tool for testing connectivity.

Refer to specific **ping** commands in the tables of configuration commands found throughout the chapters in Cisco IOS software configuration guides. Refer to the Cisco IOS software command reference publications for detailed descriptions of the command.

Tracing Packet Routes

To trace the routes that packets will actually take when traveling to their destinations, use the following command in either user or privileged EXEC mode:

Command	Purposes
`Router# trace [protocol] [destination]`	Traces packet routes through the network (privileged level).

Logging System Messages

By default, routers send logging messages (including debug command output) a logging process. The logging process controls the distribution of logging messages to various destinations, such as the logging buffer, terminal lines, or a UNIX syslog server, depending on your configuration. The process also sends messages to the console. When the logging process is on, the messages are displayed on the console after the process that generated them has finished.

When the logging process is disabled, messages are sent only to the console. The messages are sent as they are generated, so error and debug output will be interspersed with prompts or output from the command.

You can set the severity level of the messages to control the type of messages displayed for the console and each destination. You can time-stamp log messages or set the syslog source address to enhance real-time debugging and management.

System logging messages are traditionally referred to as System Error Messages. Refer to the *Cisco IOS Software System Error Messages* publication for detailed information on specific system logging messages.

Enabling System Message Logging

System message logging is enabled by default. It must be enabled in order to send messages to any destination other than the console.

To disable message logging, use the **no logging on** command. Note that disabling the logging process can slow down the router because a process cannot continue until the messages are written to the console.

To reenable message logging after it has been disabled, use the following command in global configuration mode:

Command	Purposes
`Router(config)# logging on`	Enables message logging.

Enabling Message Logging for a Slave Card

To enable slave VIP cards to log status messages to the console (print the messages to the screen), use the following command in global configuration mode:

Command	Purposes
`Router(config)# service slave-log`	Enables slave message logging.

Setting the Syslog Destination

If message logging is enabled, you can send messages to specified locations, in addition to the console.

To set the locations that receive messages, use the following commands in global configuration mode, as needed:

Command	Purposes
`Router(config)# logging buffered [size]`	Logs messages to an internal buffer.
`Router(config)# terminal monitor`	Logs messages to a nonconsole terminal.
`Router(config)# logging host`	Logs messages to a syslog server host.

The **logging buffered** command copies logging messages to an internal buffer. The buffer is circular, so newer messages overwrite older messages after the buffer is full. To display the messages that are logged in the buffer, use the **show logging** EXEC command. The first message displayed is the oldest message in the buffer. To clear the current contents of the buffer, use the **clear logging** privileged EXEC command.

The **terminal monitor** EXEC command locally accomplishes the task of displaying the system logging messages to a terminal.

The **logging** command identifies a syslog server host to receive logging messages. The *host* argument is the name or IP 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.

Configuring Synchronization of Logging Messages

You can configure the system to synchronize unsolicited messages and **debug** command output with solicited device output and prompts for a specific line. You can identify the types of messages to be output asynchronously based on the level of severity. You can also determine the maximum number of buffers for storing asynchronous messages for the terminal after which messages are dropped.

When synchronous logging of unsolicited messages and **debug** command output is turned on, unsolicited device output is displayed on the console or printed after solicited device output is displayed or printed. Unsolicited messages and **debug** command output is displayed on the console after the prompt for user input is returned. Therefore, unsolicited messages and **debug** command output are not interspersed with solicited device output and prompts. After the unsolicited messages are displayed, the console displays the user prompt again.

To configure for synchronous logging of unsolicited messages and **debug** command output with solicited device output and prompts, use the following commands beginning in global configuration mode:

	Command	Purposes
Step 1	`Router(config)# line [aux | console | vty] beginning-line-number [ending-line-number]`	Specifies the line to be configured for synchronous logging of messages.
Step 2	`Router(config-line)# logging synchronous [level severity-level | all] [limit number-of-buffers]`	Enables synchronous logging of messages.

Enabling Time-Stamps on Log Messages

By default, log messages are not time-stamped. To enable time-stamping of log messages, use either of the following commands in global configuration mode:

Command	Purposes

`Router(config)# service timestamps log uptime`
 	
or
 	
`Router(config)# service timestamps log datetime [msec] [localtime] [show-timezone]`
	Enables log time stamps.

Limiting the Error Message Severity Level and Facilities

You can limit the number of messages displayed to the selected device by specifying the severity level of the error message (see Table 18 for level descriptions). To do so, use the following commands in global configuration mode, as needed:

Command	Purposes
`Router(config)# logging console level`	Limits the number of messages logged to the console.
`Router(config)# logging monitor level`	Limits the number of messages logged to the terminal lines.
`Router(config)# logging trap level`	Limits the number of messages logged to the syslog servers.

If you have enabled syslog messages traps to be sent to a Simple Network Management Protocol (SNMP) network management station with the **snmp-server enable trap** command, you can change the level of messages sent and stored in a history table on the router. You can also change the number of messages that get stored in the history table.

Messages are stored in the history table because SNMP traps are not guaranteed to reach their destination. By default, one message of the level warning and above (see Table 18) is stored in the history table even if syslog traps are not enabled.

To change level and table size defaults, use the following commands in global configuration mode:

	Command	Purposes
Step 1	`Router(config)# logging history level`	Changes the default level of syslog messages stored in the history file and sent to the SNMP server.
Step 2	`Router(config)# logging history size number`	Changes the number of syslog messages that can be stored in the history table.

Note Table 18 lists the level keywords and severity level. For SNMP usage, the severity level values use +1. For example, **emergency** equals 1 not 0 and **critical** equals 3 not 2.

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. Table 18 lists the error message *level* keywords and corresponding UNIX syslog definitions in order from the most severe level to the least severe level.

Table 18 System Logging Message Severity Levels

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

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

The default is to log messages to the console at the **debugging** level and those level numbers that are lower, which means all levels. The **logging monitor** command defaults to **debugging** also. The **logging trap** command defaults to the **informational** level.

To display logging messages on a terminal, use the **terminal monitor** EXEC command.

Current software generates the following four categories of error messages:

- Error messages about software or hardware malfunctions, displayed at levels **warnings** through **emergencies**
- 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

Defining the UNIX System Logging Facility

You can log messages produced by UNIX system utilities. To do this, enable this type logging and define the UNIX system facility from which you want to log messages. Table 19 lists the UNIX system facilities supported by the Cisco IOS software. Consult the operator manual for your UNIX operating system for more information about these UNIX system facilities. The syslog format is compatible with Berkeley Standard Distribution (BSD) UNIX version 4.3.

To define UNIX system facility message logging, use the following command in global configuration mode:

Command	Purposes
`Router(config)# logging facility` *facility-type*	Configures system log facilities.

Table 19 Logging Facility Type Keywords

Facility Type Keyword	Description
auth	Indicates the authorization system.
cron	Indicates the cron facility.
daemon	Indicates the system daemon.
kern	Indicates the Kernel.
local0–7	Reserved for locally defined messages.
lpr	Indicates line printer system.
mail	Indicates mail system.
news	Indicates USENET news.
sys9	Indicates system use.
sys10	Indicates system use.
sys11	Indicates system use.
sys12	Indicates system use.
sys13	Indicates system use.
sys14	Indicates system use.

Table 19 Logging Facility Type Keywords (continued)

Facility Type Keyword	Description
syslog	Indicates the system log.
user	Indicates user process.
uucp	Indicates UNIX-to-UNIX copy system.

Displaying Logging Information

To display logging information, use the following commands in EXEC mode, as needed:

Command	Purposes
`Router# show logging`	Displays the state of syslog error and event logging, including host addresses, whether console logging is enabled, and other logging statistics.
`Router# show controllers vip slot-number logging`	Displays the state of syslog error and event logging of a VIP card, including host addresses, whether console logging is enabled, and other logging statistics.
`Router# show logging history`	Displays information in the syslog history table such as the table size, the status of messages, and the text of the messages stored in the table.

Logging Errors to a UNIX Syslog Daemon

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

```
local7.debugging /usr/adm/logs/cisco.log
```

The **debugging** keyword specifies the syslog level; see Table 18 for a general description of other keywords. The **local7** keyword specifies the logging facility to be used; see Table 19 for a general description of other keywords.

The syslog daemon sends messages at this level or at a more severe 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.

Setting the Syslog Source Address

By default, a syslog message contains the IP address of the interface it uses to leave the router. To set all syslog messages to contain the same IP address, regardless of which interface they use, use the following command in global configuration mode:

Command	Purposes
`Router(config)# logging source-interface type number`	Sets the syslog source address.

Using Field Diagnostics on Line Cards

Each line card on the Cisco 12000 series routers can perform field diagnostic testing to isolate faulty hardware without disrupting normal operation of the system. However, performing field diagnostic testing on a line card does halt all activity on the line card for the duration of the testing. After successful completion of the field diagnostic testing, the Cisco IOS software is automatically reloaded on the line card.

Note The field diagnostic **diag** command must be executed from the Gigabit Route Processor (GRP) main console port.

To perform field diagnostic testing on a line card, use the following command in privileged EXEC mode:

Command	Purposes
`Router# diag slot-number [previous | post | verbose | wait]`	Specifies the line card on which you want to perform diagnostic testing. Optionally, specifies that previous test results are displayed, that only extended power-on self-tests (POST) be performed, that the maximum messages are displayed, or that the Cisco IOS software not be reloaded on the line card after successful completion of the tests. The following prompt is displayed: `Running Diags will halt ALL activity on the requested slot. [confirm]` At the prompt, press **Return** to confirm that you want to perform field diagnostic testing on the specified line card, or type **no** to stop the testing.

To stop field diagnostic testing on a line card, use either of the following commands in privileged EXEC mode:

Command	Purpose

`Router# diag slot-number halt`
 	
or
 	
`Router# no diag slot-number`
	Specifies the line card on which you want to stop diagnostic testing.

Note When you stop the field diagnostic test, the line card remains down (that is, in an unbooted state). In most cases, you stopped the testing because you need to remove the line card or replace the line card. If that is not the case and you want to bring the line card back up (that is, online), you must use the **microcode reload** global configuration command or power cycle the line card.

Troubleshooting Specific Line Cards

Cisco IOS provides the **execute-on** command to allow you to issue Cisco IOS commands (such as **show** commands) to a specific line card for monitoring and maintenance. For example, you could show which Cisco IOS image is loaded on the card in slot 3 of a Cisco 12012 Gigabit Switch Router (GSR) by issuing the **execute-on slot 3 show version** command. You can also use this command for troubleshooting cards in the dial shelf of Cisco access servers. For complete documentation of this command, refer to the "Troubleshooting" chapter of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*.

Storing Line Card Crash Information

This section explains how to enable storing of crash information for a line card and optionally specify the type and amount of information stored. Technical support representatives need to be able to look at the crash information from the line card to troubleshoot serious problems on the line card. The crash information contains all the line card memory information, including the main memory and transmit and receive buffer information.

Caution Use the **exception linecard** global configuration command only when directed by a technical support representative, and only enable options that the technical support representative requests you to enable.

To enable and configure the crash information options for a line card, use the following command in global configuration mode:

Command	Purpose
Router(config)# **exception linecard** {**all** | **slot** *slot-number*} [**corefile** *filename* | **main-memory** *size* [**k** | **m**] | **queue-ram** *size* [**k** | **m**] | **rx-buffer** *size* [**k** | **m**] | **sqe-register-rx** | **sqe-register-tx** | **tx-buffer** *size* [**k** | **m**]]	Specifies the line card for which you want crash information when a line card resets. Optionally, specify the type and amount of memory to be stored.

Creating Core Dumps for System Exceptions

"System exceptions" are any unexpected system shutdowns or reboots (most frequently caused by a system failure, commonly referred to as a "system crash"). When an exception occurs, it is sometimes useful to obtain a full copy of the memory image (called a core dump) to identify the cause of the unexpected shutdown. Not all exception types will produce a core dump.

Core dumps are generally useful only to your technical support representative. The core dump file, which is a very large binary file, can be transferred to a Trivial File Transfer Protocol (TFTP), File Transfer Protocol (FTP), or Remote Copy Protocol (RCP) server, or (on limited platforms) saved to the flash disk, and subsequently interpreted by technical personnel who have access to source code and detailed memory maps.

Caution Use the **exception** commands only under the direction of a technical support representative. Creating a core dump while the router is functioning in a network can disrupt network operation.

Specifying the Destination for the Core Dump File

To configure the router to generate a core dump, you must enable exception dumps and configure a destination for the core dump file, as described in the following sections:

- Using TFTP for Core Dumps
- Using FTP for Core Dumps
- Using rcp for Core Dumps
- Using a Flash Disk for Core Dumps

Using TFTP for Core Dumps

Due to a limitation of most TFTP applications, the router will dump only the first 16 MB of the core file. Therefore, if your router's main memory is larger than 16 MB, do not use TFTP.

To configure a router for a core dump using TFTP, use the following commands in global configuration mode:

	Command or Action	Purpose
Step 1	**exception protocol tftp**	(Optional) Explicitly specifies TFTP as the protocol to be used for router exceptions (core dumps for unexpected system shutdowns). **Note** Because TFTP is the default exception protocol, the **exception protocol tftp** command does not need to be used unless the protocol has been previously changed to ftp or rcp in your system's configuration. To determine if the exception protocol has been changed, use the **show running-config** command in EXEC mode.
Step 2	**exception dump** *ip-address*	Configures the router to dump a core file to the specified server if the router crashes.
Step 3	**exception core-file** [*filepath*/]*filename*	(Optional) Specifies the name to be used for the core dump file. The file usually must pre-exist on the TFTP server, and be writable.

For example, the following command configures a router to send a core file to the server at the IP address 172.17.92.2. As the exception protocol is not specified, the default protocol of TFTP will be used.

```
Router(config)# exception dump 172.17.92.2
```

The core dump is written to a file named "*hostname*-core" on the TFTP server, where *hostname* is the name of the route (in the example above, the file would be named Router-core). You can change the name of the core file by adding the **exception core-file** *filename* configuration command.

Depending on the TFTP server application used, it may be necessary to create, on the TFTP server, the empty target file to which the router can write the core. Also, make sure there is enough memory on your TFTP server to hold the complete core dump.

Using FTP for Core Dumps

To configure the router for a core dump using FTP, use the following commands in global configuration mode:

	Command	Purposes
Step 1	`Router(config)# ip ftp username username`	(Optional) Configures the user name for FTP connections.
Step 2	`Router(config)# ip ftp password [type] password`	(Optional) Specifies the password to be used for FTP connections.
Step 3	`Router(config)# exception protocol ftp`	Specifies that FTP should be used for core dump file transfers.
Step 4	`Router(config)# exception dump ip-address`	Configures the router to dump a core file to a particular server if the router crashes.
Step 5	`Router(config)# exception core-file filename`	(Optional) Specifies the name to be used for the core dump file.

The following example configures a router to use FTP to dump a core file named "dumpfile" to the FTP server at 172.17.92.2 when it crashes.

```
ip ftp username red
ip ftp password blue
exception protocol ftp
exception dump 172.17.92.2
exception core-file dumpfile
```

Using rcp for Core Dumps

The remote copy protocol can also be used to send a core dump file. To configure the router to send core dump files using rcp, use the following commands:

	Command or Action	Purpose
Step 1	`ip rcmd remote-username username`	(Optional) Specifies the username sent by the router to the remote server with an rcp copy/write request. The remote rcp server must configured to grant write access to the specified username (in other words, an account must be defined on the network server for the username).
Step 2	`exception protocol rcp`	Configures the rcp as the protocol to use for sending core dump files.
Step 3	`exception dump ip-address`	Configures the router to dump a core file to the specified server if the router crashes.
Step 4	`exception core-file filename`	(Optional) Specifies the name to be used for the core dump file.

When an rcp username is not configured through the **ip rcmd remote-username** command, the rcp username defaults to the username associated with the current terminal (tty) connection. For example, if the user is connected to the router through Telnet and was authenticated through the username command, the router software sends the Telnet username as the rcp username. If the terminal username is not available, the router hostname will be used as the rcp username.

Using a Flash Disk for Core Dumps

Some router platforms support the Flash disk as an alternative to the linear Flash memory or PCMCIA Flash card. The large storage capacity of these Flash disks makes them good candidates for another means of capturing a core dump. To configure a router for a core dump using a Flash disk, use the following command in global configuration mode:

Command	Purpose
`Router(config)# exception flash [procmem |iomem | all] device-name[:partition-number] [erase | no_erase]`	Configures the router for a core dump using a flash disk.
`Router(config)# exception core-file filename`	(Optional) Specifies the name to be used for the core dump file.

The **show flash all** EXEC command will list the devices you can use for the **exception flash** command.

Creating an Exception Memory Core Dump

To cause the router to create a core dump and reboot when certain memory size parameters are violated during the debugging process, use the following commands in global configuration mode:

As a debugging procedure, you can cause the router to create a core dump and reboot when certain memory size parameters are violated. The following **exception memory** commands are used to trigger a core dump:

Command	Purpose
`Router(config)# exception memory minimum bytes`	Triggers a core dump and system reload when the amount of free memory falls below the specified number of bytes. • Do not specify too low a memory value, as the router needs some amout of free memory to provide the core dump. • If you enter a size that is greater than the free memory (and the **exception dump** command has been configured), a core dump and router reload is generated after 60 seconds.
`Router(config)# memory check-interval seconds`	(Optional) Increases the interval at which memory will be checked. The default is 60 seconds, but much can happen in 60 seconds to mask the cause of corruption. Reducing the interval will increase CPU utilization (by around 12 %) which will be acceptable in most cases, but will also increase the chance of getting a usable core. To make sure CPU utilization doesn't hit 100%, you should gradually decrease the interval on busy routers. The ideal interval is as low as possible without causing other system problems.
`Router(config)# exception memory fragment bytes`	Triggers a core dump and system reload when the amount of contiguous (non-fragmented) free memory falls below the specified number of bytes.
`Router(config)# exception core-file filename`	(Optional) Specifies the name to be used for the core dump file. The file usually must exist on the TFTP server, and be writable. Note that the file will be the same size as the amount of processor memory on the router.

Note that the **exception memory minimum** command is primarily useful if you anticipate running out of memory before a core dump can be triggered or other debugging can be performed (rapid memory leak); if the memory leak is gradual (slow drift), you have generally have time to perform debugging before the system runs out of memory and must be reloaded.

By default, the number of free memory bytes is checked every 60 seconds when these commands are configured. The frequency of this checking can be increased using the **memory check-interval** *seconds* command.

The **exception dump** *ip-address* command must be configured with these commands. If the **exception dump** command is not configured, the router reloads without triggering a core dump.

The following example configures the router to monitor the free memory. If the memory falls below 250000 bytes, the core dump is created and the router reloads.

```
exception dump 172.18.92.2
exception core-file memory.overrun
exception memory minimum 250000
```

Setting a Spurious Interrupt Core Dump

During the debugging process, you can configure the router to create a spurious interrupt core dump and reboot when a specified number of interrupts have occurred.

Caution Use the **exception spurious-interrupt** global configuration command only when directed by a technical support representative and only enable options requested by the technical support representative.

To enable and configure the crash information for spurious interrupts, use the following commands in global configuration mode:

Command	Purpose
`Router(config)# exception spurious-interrupt` *number*	Sets the maximum number of spurious interrupts to include in the core dump before reloading.
`Router(config)# exception dump` *ip-address* or `Router(config)# exception flash`	Specifies the destination for the core dump file.

The following example configures a router to create a core dump with a limit of two spurious interrupts:

```
exception spurious-interrupt 2
exception dump 209.165.200.225
```

Enabling Debug Operations

Your router includes hardware and software to aid in troubleshooting internal problems and problems with other hosts on the network. The **debug** privileged EXEC mode commands start the console display of several classes of network events. The following commands describe in general the system debug message feature. Refer to the *Cisco IOS Debug Command Reference* for all information regarding **debug** commands. Also refer to the *Internetwork Troubleshooting Guide* publication for additional information.

To enable debugging operations, use the following commands:

Command	Purposes
`Router# show debugging`	Displays the state of each debugging option.
`Router# debug ?`	Displays a list and brief description of all the **debug** command options.
`Router# debug` *command*	Begins message logging for the specified **debug** command.
`Router# no debug` *command*	Turns message logging off for the specified **debug** command.

Caution The system gives high priority to debugging output. For this reason, debugging commands should be turned on only for troubleshooting specific problems or during troubleshooting sessions with technical support personnel. Excessive debugging output can render the system inoperable.

You can configure time-stamping of system **debug** messages. Time-stamping enhances real-time debugging by providing the relative timing of logged events. This information is especially useful when customers send debugging output to your technical support personnel for assistance. To enable time-stamping of system **debug** messages, use either of the following commands in global configuration mode:

Command	Purposes

`Router(config)# service timestamps debug uptime`
or
`Router(config)# service timestamps debug datetime [msec] [localtime] [show-timezone]`
	Enables time-stamping of system **debug** messages.

Normally, the messages are displayed only on the console terminal. Refer to the section "Setting the Syslog Destination" earlier in this chapter to change the output device.

Enabling Conditionally Triggered Debugging

When the Conditionally Triggered Debugging feature is enabled, the router generates debugging messages for packets entering or leaving the router on a specified interface; the router will not generate debugging output for packets entering or leaving through a different interface. You can specify the interfaces explicitly. For example, you may only want to see debugging messages for one interface or subinterface. You can also turn on debugging for all interfaces that meet specified condition. This feature is useful on dial access servers, which have a large number of ports.

Normally, the router will generate debugging messages for every interface, resulting in a large number of messages. The large number of messages consumes system resources, and can affect your ability to find the specific information you need. By limiting the number of debugging messages, you can receive messages related to only the ports you wish to troubleshoot.

Conditionally Triggered Debugging controls the output from the following protocol-specific **debug** commands:

- **debug aaa** {**accounting** | **authorization** | **authentication**}
- **debug dialer** {**events** | **packets**}
- **debug isdn** {**q921** | **q931**}
- **debug modem** {**oob** | **trace**}
- **debug ppp** {**all** | **authentication** | **chap** | **error** | **negotiation** | **multilink events** | **packet**}

Although this feature limits the output of the commands listed, it does not automatically enable the generation of debugging output from these commands. Debugging messages are generated only when the protocol-specific **debug** command is enabled. The **debug** command output is controlled through two processes:

- The protocol-specific **debug** commands specify which protocols are being debugged. For example, the **debug dialer events** command generates debugging output related to dialer events.
- The **debug condition** commands limit these debugging messages to those related to a particular interface. For example, the **debug condition username bob** command generates debugging output only for interfaces with packets that specify a username of bob.

To configure Conditionally Triggered Debugging, perform the tasks described in the following sections:

- Enabling Protocol-Specific debug Commands
- Enabling Conditional Debugging Commands
- Specifying Multiple Debugging Conditions

Enabling Protocol-Specific debug Commands

In order to generate any debugging output, the protocol-specific **debug** command for the desired output must be enabled. Use the **show debugging** command to determine which types of debugging are enabled. To display the current debug conditions, use the **show debug condition** command. To enable the desired protocol-specific **debug** commands, use the following commands in privileged EXEC mode :

Command	Purpose
`Router# show debugging`	Determines which types of debugging are enabled.
`Router# show debug condition [condition-id]`	Displays the current **debug** conditions.
`Router# debug protocol`	Enables the desired debugging commands.
`Router# no debug protocol`	Disables the debugging commands that are not desired.

If you do not want output, disable all the protocol-specific **debug** commands.

Enabling Conditional Debugging Commands

If no **debug condition** commands are enabled, all debugging output, regardless of the interface, will be displayed for the enabled protocol-specific **debug** commands.

The first **debug condition** command you enter enables conditional debugging. The router will display only messages for interfaces that meet one of the specified conditions. If multiple conditions are specified, the interface must meet at least one of the conditions in order for messages to be displayed.

To enable messages for interfaces specified explicitly or for interfaces that meet certain conditions, perform the tasks described in the following sections:

- Displaying Messages for One Interface
- Displaying Messages for Multiple Interfaces
- Limiting the Number of Messages Based on Conditions

Displaying Messages for One Interface

To disable debugging messages for all interfaces except one, use the following command in privileged EXEC mode:

Command	Purpose
`Router# debug condition interface interface`	Enables debugging output for only the specified interface.

To reenable debugging output for all interfaces, use the **no debug interface** command.

Displaying Messages for Multiple Interfaces

To enable debugging messages for multiple interfaces, use the following commands in privileged EXEC mode:

	Command	Purposes
Step 1	`Router# debug condition interface interface`	Enables debugging output for only the specified interface
Step 2	`Router# debug condition interface interface`	Enable debugging messages for additional interfaces. Repeat this task until debugging messages are enabled for all desired interfaces.

If you specify more than one interface by entering this command multiple times, debugging output will be displayed for all of the specified interfaces. To turn off debugging on a particular interface, use the **no debug interface** command. If you use the **no debug interface all** command or remove the last **debug interface** command, debugging output will be reenabled for all interfaces.

Limiting the Number of Messages Based on Conditions

The router can monitor interfaces to learn if any packets contain the specified value for one of the following conditions:

- username

- calling party number
- called party number

If you enter a condition, such as calling number, debug output will be stopped for all interfaces. The router will then monitor every interface to learn if a packet with the specified calling party number is sent or received on any interfaces. If the condition is met on an interface or subinterface, **debug** command output will be displayed for that interface. The debugging output for an interface is "triggered" when the condition has been met. The debugging output continues to be disabled for the other interfaces. If, at some later time, the condition is met for another interface, the debug output also will become enabled for that interface.

Once debugging output has been triggered on an interface, the output will continue until the interface goes down. However, the session for that interface might change, resulting in a new username, called party number, or calling party number. Use the **no debug interface** command to reset the debug trigger mechanism for a particular interface. The debugging output for that interface will be disabled until the interface meets one of the specified conditions.

To limit the number of debugging messages based on a specified condition, use the following command in privileged EXEC mode:

Command	Purpose
`Router# debug condition {username username | called dial-string | caller dial-string}`	Enables conditional debugging. The router will display only messages for interfaces that meet this condition.

To reenable the debugging output for all interfaces, enter the **no debug condition all** command.

Specifying Multiple Debugging Conditions

To limit the number of debugging messages based on more than one condition, use the following commands in privileged EXEC mode:

	Command	Purposes
Step 1	`Router# debug condition {username username | called dial-string | caller dial-string}`	Enables conditional debugging, and specifies the first condition.
Step 2	`Router# debug condition {username username | called dial-string | caller dial-string}`	Specifies the second condition. Repeat this task until all conditions are specified.

If you enter multiple **debug condition** commands, debugging output will be generated if an interface meets at least one of the conditions. If you remove one of the conditions using the **no debug condition** command, interfaces that meet only that condition no longer will produce debugging output. However, interfaces that meet a condition other than the removed condition will continue to generate output. Only if no active conditions are met for an interface will the output for that interface be disabled.

Conditionally Triggered Debugging Configuration Examples

In this example, four conditions have been set by the following commands:

- **debug condition interface serial 0**

- **debug condition interface serial 1**
- **debug condition interface virtual-template 1**
- **debug condition username fred**

The first three conditions have been met by one interface. The fourth condition has not yet been met:

```
Router# show debug condition

Condition 1: interface Se0 (1 flags triggered)
        Flags: Se0
Condition 2: interface Se1 (1 flags triggered)
        Flags: Se1
Condition 3: interface Vt1 (1 flags triggered)
        Flags: Vt1
Condition 4: username fred (0 flags triggered)
```

When any **debug condition** command is entered, debugging messages for conditional debugging are enabled. The following debugging messages show conditions being met on different interfaces as the serial 0 and serial 1 interfaces come up. For example, the second line of output indicates that serial interface 0 meets the username fred condition.

```
*Mar  1 00:04:41.647: %LINK-3-UPDOWN: Interface Serial0, changed state to up
*Mar  1 00:04:41.715: Se0 Debug: Condition 4, username fred triggered, count 2
*Mar  1 00:04:42.963: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed
state to up
*Mar  1 00:04:43.271: Vi1 Debug: Condition 3, interface Vt1 triggered, count 1
*Mar  1 00:04:43.271: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
*Mar  1 00:04:43.279: Vi1 Debug: Condition 4, username fred triggered, count 2
*Mar  1 00:04:43.283: Vi1 Debug: Condition 1, interface Se0 triggered, count 3
*Mar  1 00:04:44.039: %IP-4-DUPADDR: Duplicate address 172.27.32.114 on Ethernet 0,
sourced by 00e0.1e3e.2d41
*Mar  1 00:04:44.283: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to up
*Mar  1 00:04:54.667: %LINK-3-UPDOWN: Interface Serial1, changed state to up
*Mar  1 00:04:54.731: Se1 Debug: Condition 4, username fred triggered, count 2
*Mar  1 00:04:54.735: Vi1 Debug: Condition 2, interface Se1 triggered, count 4
*Mar  1 00:04:55.735: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed
state to up
```

After a period of time, the **show debug condition** command displays the revised list of conditions:

```
Router# show debug condition

Condition 1: interface Se0 (2 flags triggered)
        Flags: Se0 Vi1
Condition 2: interface Se1 (2 flags triggered)
        Flags: Se1 Vi1
Condition 3: interface Vt1 (2 flags triggered)
        Flags: Vt1 Vi1
Condition 4: username fred (3 flags triggered)
        Flags: Se0 Vi1 Se1
```

Next, the serial 1 and serial 0 interfaces go down. When an interface goes down, conditions for that interface are cleared.

```
*Mar  1 00:05:51.443: %LINK-3-UPDOWN: Interface Serial1, changed state to down
*Mar  1 00:05:51.471: Se1 Debug: Condition 4, username fred cleared, count 1
*Mar  1 00:05:51.479: Vi1 Debug: Condition 2, interface Se1 cleared, count 3
*Mar  1 00:05:52.443: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed
state to down
*Mar  1 00:05:56.859: %LINK-3-UPDOWN: Interface Serial0, changed state to down
*Mar  1 00:05:56.887: Se0 Debug: Condition 4, username fred cleared, count 1
*Mar  1 00:05:56.895: Vi1 Debug: Condition 1, interface Se0 cleared, count 2
```

```
*Mar  1 00:05:56.899: Vi1 Debug: Condition 3, interface Vt1 cleared, count 1
*Mar  1 00:05:56.899: Vi1 Debug: Condition 4, username fred cleared, count 0
*Mar  1 00:05:56.903: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to down
*Mar  1 00:05:57.907: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed
state to down
*Mar  1 00:05:57.907: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to down
```

The final **show debug condition** output is the same as the output before the interfaces came up:

```
Router# show debug condition

Condition 1: interface Se0 (1 flags triggered)
        Flags: Se0
Condition 2: interface Se1 (1 flags triggered)
        Flags: Se1
Condition 3: interface Vt1 (1 flags triggered)
        Flags: Vt1
Condition 4: username fred (0 flags triggered)
```

Using the Environmental Monitor

Some routers and access servers have an environmental monitor that monitors the physical condition of the router. If a measurement exceeds acceptable margins, a warning message is printed to the system console. The system software collects measurements once every 60 seconds, but warnings for a given test point are printed at most once every 4 hours. If the temperature measurements are out of specification more than the shutdown, the software shuts the router down (the fan will remain on). The router must be manually turned off and on after such a shutdown. You can query the environmental monitor using the **show environment** command at any time to determine whether a measurement is out of tolerance. Refer to the *Cisco IOS System Error Messages* publication for a description of environmental monitor warning messages.

On routers with an environmental monitor, if the software detects that any of its temperature test points have exceeded maximum margins, it performs the following steps:

1. Saves the last measured values from each of the six test points to internal nonvolatile memory.
2. Interrupts the system software and causes a shutdown message to be printed on the system console.
3. Shuts off the power supplies after a few milliseconds of delay.

The system displays the following message if temperatures exceed maximum margins, along with a message indicating the reason for the shutdown:

```
Router#
%ENVM-1-SHUTDOWN: Environmental Monitor initiated shutdown
%ENVM-2-TEMP: Inlet temperature has reached SHUTDOWN level at 64(C)
```

Refer to the hardware installation and maintenance publication for your router for more information about environmental specifications.

This document first published April 2001. Last updated April 2004 (revision 13).

To report errors in this document, send a detailed email to bug-doc@cisco.com.

Configuring SNMP Support

This chapter describes the Simple Network Management Protocol (SNMP), SNMP MIBs, and how to configure SNMP on Cisco devices.

For a complete description of the router monitoring commands mentioned in this chapter, see the "SNMP Commands" chapter in the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online. For further information about using SNMP, see the SNMP Technical Tips area on Cisco.com at http://www.cisco.com/warp/public/477/SNMP/snmp-indx.html.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

This chapter contains the following sections:

- Understanding SNMP
- SNMP Configuration Task List
- SNMP Configuration Examples
- New MIB Features in Cisco IOS Release 12.2

Understanding SNMP

SNMP is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language used for the monitoring and management of devices in a network.

The SNMP framework has three parts:

- An SNMP manager
- An SNMP agent
- A MIB

The SNMP manager is the system used to control and monitor the activities of network hosts using SNMP. The most common managing system is called a Network Management System (NMS). The term NMS can be applied to either a dedicated device used for network management, or the applications used on such a device. A variety of network management applications are available for use with SNMP. These features range from simple command-line applications to feature-rich graphical user interfaces (such as the CiscoWorks2000 line of products).

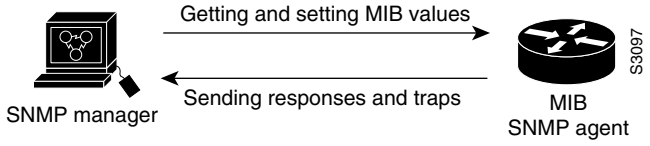
The SNMP agent is the software component within the managed device that maintains the data for the device and reports these data, as needed, to managing systems. The agent and MIB reside on the routing device (router, access server, or switch). To enable the SNMP agent on a Cisco routing device, you must define the relationship between the manager and the agent.

The Management Information Base (MIB) is a virtual information storage area for network management information, which consists of collections of managed objects. Within the MIB there are collections of related objects, defined in MIB modules. MIB modules are written in the SNMP MIB module language, as defined in STD 58, RFC 2578, RFC 2579, and RFC 2580 (see the "MIBs and RFCs" section for an explanation of RFC and STD documents). Note that individual MIB modules are also referred to as MIBs; for example, the Interfaces Group MIB (IF-MIB) is a MIB module within *the* MIB on your system.

The SNMP agent contains MIB variables whose values the SNMP manager can request or change through Get or Set operations. A manager can get a value from an agent or store a value into that agent. The agent gathers data from the MIB, the repository for information about device parameters and network data. The agent can also respond to manager requests to Get or Set data.

Figure 14 illustrates the communications relationship between the SNMP manager and agent. A manager can send the agent requests to get and set MIB values. The agent can respond to these requests. Independent of this interaction, the agent can send unsolicited notifications (traps or informs) to the manager to notify the manager of network conditions.

Figure 14 Communication Between an SNMP Agent and Manager



Note This chapter discusses how to enable the SNMP agent on your Cisco device, and how to control the sending of SNMP notifications from the agent. For information on using SNMP management systems, see the appropriate documentation for your NMS application.

SNMP Notifications

A key feature of SNMP is the ability to generate notifications from an SNMP agent. These notifications do not require that requests be sent from the SNMP manager. Unsolicited (asynchronous) notifications can be generated as *traps* or *inform requests*. Traps are messages alerting the SNMP manager to a condition on the network. Inform requests (informs) are traps that include a request for confirmation of receipt from the SNMP manager. Notifications can indicate improper user authentication, restarts, the closing of a connection, loss of connection to a neighbor router, or other significant events.

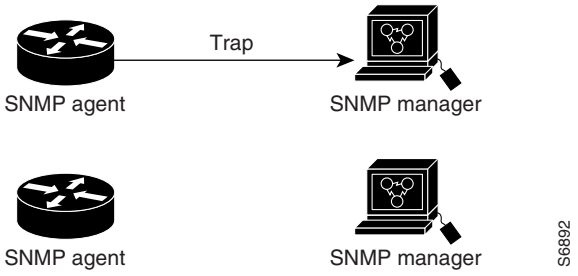
Traps are less reliable than informs because the receiver does not send any acknowledgment when it receives a trap. The sender cannot determine if the trap was received. An SNMP manager that receives an inform request acknowledges the message with an SNMP response protocol data unit (PDU). If the manager does not receive an inform request, it does not send a response. If the sender never receives a response, the inform request can be sent again. Thus, informs are more likely to reach their intended destination.

However, traps are often preferred because informs consume more resources in the router and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request must be held in memory until a response is received or the request times out. Also, traps are sent only once, while an inform may be retried several times. The retries increase traffic and contribute to a higher overhead on the network. Thus, traps and inform requests provide a trade-off between reliability and resources. If it is important that the SNMP manager receives every notification, use inform requests. However, if you are concerned about traffic on your network or memory in the router and you need not receive every notification, use traps.

Figure 15 through Figure 18 illustrate the differences between traps and inform requests.

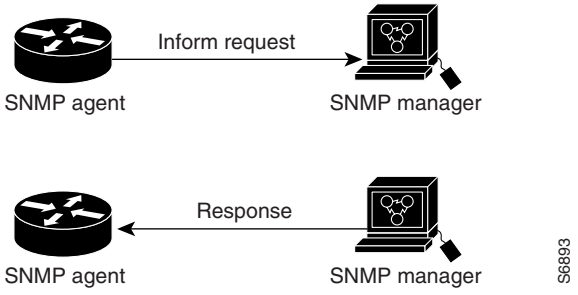
In Figure 15, the agent router successfully sends a trap to the SNMP manager. Although the manager receives the trap, it does not send any acknowledgment to the agent. The agent has no way of knowing that the trap reached its destination.

Figure 15 Trap Successfully Sent to SNMP Manager



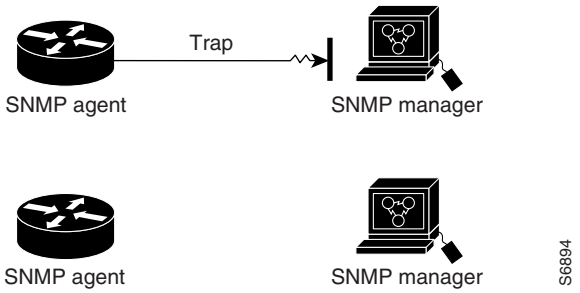
In Figure 16, the agent router successfully sends an inform request to the manager. When the manager receives the inform request, it sends a response to the agent. Thus, the agent knows that the inform request reached its destination. Notice that, in this example, twice as much traffic is generated as in Figure 15; however, the agent knows that the manager received the notification.

Figure 16 Inform Request Successfully Sent to SNMP Manager



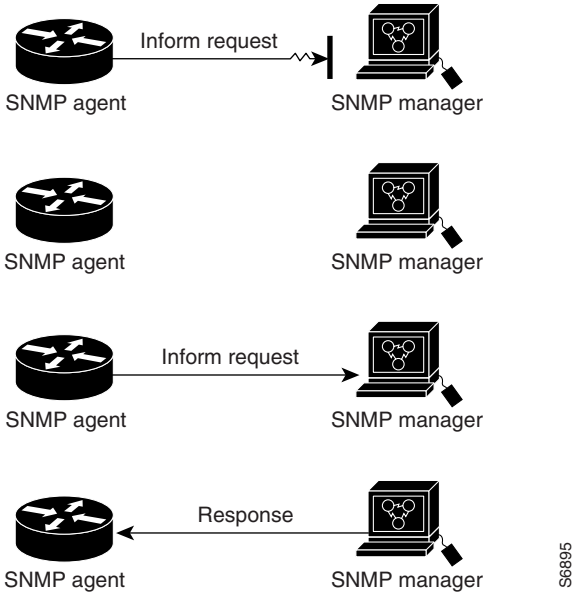
In Figure 17, the agent sends a trap to the manager, but the trap does not reach the manager. Because the agent has no way of knowing that the trap did not reach its destination, the trap is not sent again. The manager never receives the trap.

Figure 17 Trap Unsuccessfully Sent to SNMP Manager



In Figure 18, the agent sends an inform request to the manager, but the inform request does not reach the manager. Because the manager did not receive the inform request, it does not send a response. After a period of time, the agent will resend the inform request. The second time, the manager receives the inform request and replies with a response. In this example, there is more traffic than in Figure 17; however, the notification reaches the SNMP manager.

Figure 18 Inform Request Unsuccessfully Sent to SNMP Manager



MIBs and RFCs

MIB modules typically are defined in RFC documents submitted to the Internet Engineering Task Force (IETF), an international standards body. RFCs are written by individuals or groups for consideration by the Internet Society and the Internet community as a whole, usually with the intention of establishing a recommended Internet standard. Before being given RFC status, recommendations are published as Internet Draft (I-D) documents. RFCs that have become recommended standards are also labeled as standards (STD) documents. You can learn about the standards process and the activities of the IETF at the Internet Society website at http://www.isoc.org. You can read the full text of all RFCs, I-Ds, and STDs referenced in Cisco documentation at the IETF website at http://www.ietf.org.

The Cisco implementation of SNMP uses the definitions of MIB II variables described in RFC 1213 and definitions of SNMP traps described in RFC 1215.

Cisco provides its own private MIB extensions with every system. Cisco enterprise MIBs comply with the guidelines described in the relevant RFCs unless otherwise noted in the documentation. You can find the MIB module definition files and list of which MIBs are supported on each Cisco platform on the Cisco MIB website on Cisco.com.

For a list of new MIB-related functionality, see the "New MIB Features in Cisco IOS Release 12.2" section.

SNMP Versions

Cisco IOS software supports the following versions of SNMP:

- SNMPv1—The Simple Network Management Protocol: A Full Internet Standard, defined in RFC 1157. (RFC 1157 replaces the earlier versions that were published as RFC 1067 and RFC 1098.) Security is based on community strings.
- SNMPv2c—The community-string based Administrative Framework for SNMPv2. SNMPv2c (the "c" stands for "community") is an Experimental Internet Protocol defined in RFC 1901, RFC 1905, and RFC 1906. SNMPv2c is an update of the protocol operations and data types of SNMPv2p (SNMPv2 Classic), and uses the community-based security model of SNMPv1.
- SNMPv3—Version 3 of SNMP. SNMPv3 is an interoperable standards-based protocol defined in RFCs 2273 to 2275. SNMPv3 provides secure access to devices by a combination of authenticating and encrypting packets over the network.

  The security features provided in SNMPv3 are as follows:

  - Message integrity—Ensuring that a packet has not been tampered with in transit.
  - Authentication—Determining that the message is from a valid source.
  - Encryption—Scrambling the contents of a packet prevent it from being learned by an unauthorized source.

Both SNMPv1 and SNMPv2c use a community-based form of security. The community of managers able to access the agent MIB is defined by an IP address Access Control List and password.

SNMPv2c support includes a bulk retrieval mechanism and more detailed error message reporting to management stations. The bulk retrieval mechanism supports the retrieval of tables and large quantities of information, minimizing the number of round-trips required. The SNMPv2C improved error handling support includes expanded error codes that distinguish different kinds of error conditions; these conditions are reported through a single error code in SNMPv1. Error return codes now report the error type. Three kinds of exceptions are also reported: no such object exceptions, no such instance exceptions, and end of MIB view exceptions.

SNMPv3 is a security model.A security model is an authentication strategy that is set up for a user and the group in which the user resides. A security level is the permitted level of security within a security model. A combination of a security model and a security level will determine which security mechanism is employed when handling an SNMP packet. See Table 20 for a list of security levels available in SNMPv3.

Three security models are available: SNMPv1, SNMPv2c, and SNMPv3. Table 20 identifies what the combinations of security models and levels mean.

Table 20 SNMP Security Models and Levels

Model	Level	Authentication	Encryption	What Happens
v1	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v2c	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v3	noAuthNoPriv	Username	No	Uses a username match for authentication.
v3	authNoPriv	MD5 or SHA	No	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms.
v3	authPriv	MD5 or SHA	DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides DES 56-bit encryption in addition to authentication based on the CBC-DES (DES-56) standard.

Note SNMPv2p (SNMPv2 Classic) is not supported in any Cisco IOS releases after 11.2. SNMPv2c replaces the Party-based Administrative and Security Framework of SNMPv2p with a Community-based Administrative Framework. SNMPv2c retained the bulk retrieval and error handling capabilities of SNMPv2p.

You must configure the SNMP agent to use the version of SNMP supported by the management station. An agent can communicate with multiple managers; for this reason, you can configure the Cisco IOS software to support communications with one management station using the SNMPv1 protocol, one using the SNMPv2c protocol, and another using SMNPv3.

The SNMPv3 feature supports RFCs 1901 to 1908, 2104, 2206, 2213, 2214, and 2271 to 2275. For additional information on SNMPv3, refer to RFC 2570, *Introduction to Version 3 of the Internet-standard Network Management Framework* (note that this is not a standards document).

SNMP Configuration Task List

There is no specific command that you use to enable SNMP. The first **snmp-server** command that you enter enables the supported versions of SNMP.

To configure SNMP support, perform the tasks described in the following sections. Each task is labeled as required or optional.

- Creating or Modifying an SNMP View Record (Optional)
- Creating or Modifying Access Control for an SNMP Community (Required)
- Specifying an SNMP-Server Engine Name (ID) (Optional)

- Specifying SNMP-Server Group Names (Optional)
- Configuring SNMP-Server Hosts (Required)
- Configuring SNMP-Server Users (Optional)
- Enabling the SNMP Agent Shutdown Mechanism (Optional)
- Setting the Contact, Location, and Serial Number of the SNMP Agent (Optional)
- Defining the Maximum SNMP Agent Packet Size (Optional)
- Limiting the Number of TFTP Servers Used via SNMP (Optional)
- Monitoring and Troubleshooting SNMP Status (Optional)
- Disabling the SNMP Agent (Optional)
- Configuring SNMP Notifications (Required)
- Configuring the Router as an SNMP Manager (Optional)

Creating or Modifying an SNMP View Record

You can assign views to community strings to limit which MIB objects an SNMP manager can access. You can use a predefined view, or create your own view. If you are using a predefined view or no view at all, skip this task.

To create or modify an SNMP view record, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server view` *view-name oid-tree* {**included** | **excluded**}	Creates or modifies a view record.

To remove a view record, use the **no snmp-server view** command.

You can enter this command multiple times for the same view record. Later lines take precedence when an object identifier is included in two or more lines.

Creating or Modifying Access Control for an SNMP Community

Use an SNMP community string to define the relationship between the SNMP manager and the agent. The community string acts like a password to regulate access to the agent on the router. Optionally, you can specify one or more of the following characteristics associated with the string:

- An access list of IP addresses of the SNMP managers that are permitted to use the community string to gain access to the agent.
- A MIB view, which defines the subset of all MIB objects accessible to the given community.
- Read and write or read-only permission for the MIB objects accessible to the community.

To configure a community string, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server community` *string* [**view** *view-name*] [**ro** | **rw**] [*number*]	Defines the community access string.

You can configure one or more community strings. To remove a specific community string, use the **no snmp-server community** command.

For an example of configuring a community string, see the "SNMP Configuration Examples" section.

Specifying an SNMP-Server Engine Name (ID)

To specify an identification name (ID) for a local SNMP engine, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server engineID local engineid-string`	Specifies the name of the local SNMP engine (or copy of SNMP).

To specify an ID for a remote SNMP engine, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server engineID remote ip-address [udp-port port-number] engineid-string`	Specifies the name of the remote SNMP engine (or copy of SNMP).

Specifying SNMP-Server Group Names

To specify a new SNMP group, or a table that maps SNMP users to SNMP views, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server group [groupname {v1 | v2c | v3 [auth | noauth | priv]}][read readview] [write writeview] [notify notifyview] [access access-list]`	Configures a new SNMP group, or a table that maps SNMP users to SNMP views.

Configuring SNMP-Server Hosts

To configure the recipient of an SNMP trap operation, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server host host-id [traps | informs][version {1 | 2c | 3 [auth | noauth | priv]} ] community-string [udp-port port-number] [notification-type]`	Specifies whether you want the SNMP notifications sent as traps or informs, the version of SNMP to use, the security level of the notifications (for SNMPv3), and the recipient (host) of the notifications.

Configuring SNMP-Server Users

To configure a new user to an SNMP group, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server user username groupname [remote ip-address [udp-port port]] {v1 | v2c | v3 [encrypted] [auth {md5 | sha} auth-password ]} [access access-list]`	Configures a new user to an SNMP group.

Enabling the SNMP Agent Shutdown Mechanism

Using SNMP packets, a network management tool can send messages to users on virtual terminals and the console. This facility operates in a similar fashion to the **send** EXEC 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. After a system is shut down, typically it is reloaded. Because the ability to cause a reload from the network is a powerful feature, it is protected by the **snmp-server system-shutdown** global configuration command. If you do not issue this command, the shutdown mechanism is not enabled. To enable the SNMP agent shutdown mechanism, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server system-shutdown`	Enables system shutdown using the SNMP message reload feature.

Setting the Contact, Location, and Serial Number of the SNMP Agent

You can set the system contact, location, and serial number of the SNMP agent so that these descriptions can be accessed through the configuration file. To do so, use the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)# snmp-server contact text`	Sets the system contact string.
`Router(config)# snmp-server location text`	Sets the system location string.
`Router(config)# snmp-server chassis-id number`	Sets the system serial number.

Defining the Maximum SNMP Agent Packet Size

You can define the maximum packet size permitted when the SNMP agent is receiving a request or generating a reply. To do so, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server packetsize byte-count`	Establishes the maximum packet size.

Limiting the Number of TFTP Servers Used via SNMP

You can limit the number of TFTP servers used for saving and loading configuration files via SNMP to the servers specified in an access list. To do so, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server tftp-server-list number`	Limits the number of TFTP servers used for configuration file copies via SNMP to the servers in an access list.

Monitoring and Troubleshooting SNMP Status

To monitor and troubleshoot SNMP status and information, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router> show snmp`	Monitors SNMP status.
`Router> show snmp engineID [local | remote]`	Displays information about the local SNMP engine and all remote engines that have been configured on the device.
`Router> show snmp groups`	Displays information about each SNMP group on the network.
`Router> show snmp user`	Displays information about each SNMP username in the SNMP users table.

To monitor SNMP trap activity in real time for the purposes of troubleshooting, use the SNMP **debug** commands, including the **debug snmp packet** EXEC command. For documentation of SNMP **debug** commands, see the *Cisco IOS Debug Command Reference.*

Disabling the SNMP Agent

To disable any version of the SNMP agent, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no snmp-server`	Disables SNMP agent operation.

Configuring SNMP Notifications

To configure the router to send SNMP traps or informs, perform the tasks described in the following sections:

- Configuring the Router to Send SNMP Notifications (Required)
- Changing Notification Operation Values (Optional)
- Controlling Individual RFC 1157 SNMP Traps (Optional)

Note Most Cisco IOS commands use the word "traps" in their command syntax. Unless there is an option within the command to specify either traps or informs, the keyword **traps** should be taken to mean either traps or informs, or both. Use the **snmp-server host** command to specify whether you want SNMP notifications to be sent as traps or informs.
The SNMP Proxy manager must be available and enabled on the device for informs to be used. The SNMP Proxy manager is shipped with PLUS software images only.

Configuring the Router to Send SNMP Notifications

To configure the router to send traps or informs to a host, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# snmp-server engineID remote remote-ip-addr remote-engineID`	Specifies the engine ID for the remote host.
Step 2	`Router(config)# snmp-server user username groupname [remote host [udp-port port] {v1 | v2c | v3 [encrypted] [auth {md5 | sha} auth-password]} [access access-list]`	Configures an SNMP user to be associated with the host created in Step 1. **Note** You cannot configure a remote user for an address without first configuring the engine ID for that remote host . This is a restriction imposed in the design of these commands; if you try to configure the user before the host, you will receive a warning message and the command will not be executed
Step 3	`Router(config)# snmp group groupname {v1 | v2 | v3 {auth | noauth | priv}} [read readview] [write writeview] [notify notifyview] [access access-list]`	Configures an SNMP group.
Step 4	`Router(config)# snmp-server host host [traps | informs] [version {1 | 2c | 3 [auth | noauth | priv]}] community-string [notification-type]`	Specifies whether you want the SNMP notifications sent as traps or informs, the version of SNMP to use, the security level of the notifications (for SNMPv3), and the recipient (host) of the notifications.
Step 5	`Router(config)# snmp-server enable traps [notification-type [notification-options]]`	Enables sending of traps or informs, and specifies the type of notifications to be sent. If a *notification-type* is not specified, all supported notification will be enabled on the router. To discover which notifications are available on your router, enter the **snmp-server enable traps ?** command.

The **snmp-server host** command specifies which hosts will receive SNMP notifications, and whether you want the notifications sent as traps or inform requests. The **snmp-server enable traps** command globally enables the production mechanism for the specified notification types (such as Border Gateway Protocol [BGP] traps, config traps, entity traps, Hot Standby Router Protocol [HSRP] traps, and so on).

Changing Notification Operation Values

You can specify a value other than the default for the source interface, message (packet) queue length for each host, or retransmission interval.

To change notification operation values, use the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)# snmp-server trap-source interface`	Specifies a source interface for trap or inform notifications.
`Router(config)# snmp-server queue-length length`	Establishes the message queue length for each notification.
`Router(config)# snmp-server trap-timeout seconds`	Defines how often to resend notifications on the retransmission queue.

For inform requests, you can configure inform-specific operation values in addition to the operation values mentioned. To change inform operation values, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server informs [retries retries] [timeout seconds] [pending pending]`	Sets the maximum number of times to resend an inform request, the number of seconds to wait for an acknowledgment before resending, and the maximum number of informs waiting for acknowledgments at any one time.

Controlling Individual RFC 1157 SNMP Traps

Starting with Cisco IOS Release 12.1(3)T, you can globally enable or disable authenticationFailure, linkUp, linkDown, warmStart, and coldStart notifications (traps or informs) individually. (These traps constitute the "generic traps" defined in RFC 1157.) To enable any of these notification types, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server enable traps snmp [authentication] [linkup] [linkdown] [warmstart] [coldstart]`	Enables RFC 1157 generic traps. When used without any of the optional keywords, enables authenticationFailure, linkUp, linkDown, warmStart, and coldStart traps. When used with keywords, enables only the trap types specified.

For example, to globally enable only linkUp and linkDown SNMP traps or informs for all interfaces, use the **snmp-server enable traps snmp linkup linkdown** form of this command.

Note that linkUp and linkDown notifications are enabled by default on specific interfaces, but will not be sent unless they are enabled globally. To control (disable or reenable) the sending of linkUp/linkDown notifications for specific interfaces, use the **no snmp trap link-status** command in interface configuration mode.

Configuring the Router as an SNMP Manager

The SNMP manager feature allows a router to act as a network management station. In other words, configuring a router as an SNMP manager allows it to act as an SNMP client. As an SNMP manager, the router can send SNMP requests to agents and receive SNMP responses and notifications from agents. When the SNMP manager process is enabled, the router can query other SNMP agents and process incoming SNMP traps.

Security Considerations

Most network security policies assume that routers will accept SNMP requests, send SNMP responses, and send SNMP notifications.

With the SNMP manager functionality enabled, the router may also send SNMP requests, receive SNMP responses, and receive SNMP notifications. Your security policy implementation may need to be updated prior to enabling this feature.

SNMP requests typically are sent to User Datagram Protocol (UDP) port 161. SNMP responses are typically sent from UDP port 161. SNMP notifications are typically sent to UDP port 162.

SNMP Sessions

Sessions are created when the SNMP manager in the router sends SNMP requests, such as inform requests, to a host, or receives SNMP notifications from a host. One session is created for each destination host. If there is no further communication between the router and host within the session timeout period, the session will be deleted.

The router tracks statistics, such as the average round-trip time required to reach the host, for each session. Using the statistics for a session, the SNMP manager in the router can set reasonable timeout periods for future requests, such as informs, for that host. If the session is deleted, all statistics are lost. If another session with the same host is later created, the request timeout value for replies will return to the default value.

Sessions consume memory. A reasonable session timeout value should be large enough that regularly used sessions are not prematurely deleted, yet small enough such that irregularly used, or one-time sessions, are purged expeditiously.

Enabling the SNMP Manager

To enable the SNMP manager process and set the session timeout value, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# snmp-server manager`	Enables the SNMP manager.
Step 2	`Router(config)# snmp-server manager session-timeout` *seconds*	(Optional) Changes the session timeout value.

Monitoring the SNMP Manager

To monitor the SNMP manager process, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router>` **`show snmp`**	Displays global SNMP information.
`Router>` **`show snmp sessions`** [**`brief`**]	Displays information about current sessions.
`Router>` **`show snmp pending`**	Displays information about current pending requests.

SNMP Configuration Examples

The following example enables SNMPv1, SNMPv2c, and SNMPv3. The configuration permits any SNMP manager to access all objects with read-only permissions using the community string named public. This configuration does not cause the router to send any traps.

```
snmp-server community public
```

The following example permits any SNMP to access all objects with read-only permission using the community string named public. The router also will send ISDN traps to the hosts 172.16.1.111 and 172.16.1.33 using SNMPv1 and to the host 172.16.1.27 using SNMPv2c. The community string named public is sent with the traps.

```
snmp-server community public
snmp-server enable traps isdn
snmp-server host 172.16.1.27 version 2c public
snmp-server host 172.16.1.111 version 1 public
snmp-server host 172.16.1.33 public
```

The following example allows read-only access for all objects to members of access list 4 that specify the comaccess community string. No other SNMP managers have access to any objects. SNMP Authentication Failure traps are sent by SNMPv2c to the host cisco.com using the community string named public.

```
snmp-server community comaccess ro 4
snmp-server enable traps snmp authentication
snmp-server host cisco.com version 2c public
```

The following example sends Entity MIB inform notifications to the host cisco.com. The community string is restricted. The first line enables the router to send Entity MIB notifications in addition to any traps or informs previously enabled. The second line specifies that the notifications should be sent as inform requests, specifies the destination of these informs, and overwrites any previous **snmp-server host** commands for the host cisco.com.

```
snmp-server enable traps entity
snmp-server host informs cisco.com restricted entity
```

The following example sends the SNMP and Cisco environmental monitor enterprise-specific traps to address 172.30.2.160:

```
snmp-server enable traps
snmp-server host 172.30.2.160 public snmp envmon
```

The following example enables the router to send all traps to the host myhost.cisco.com using the community string public:

```
snmp-server enable traps
```

```
snmp-server host myhost.cisco.com public
```

The following example will not send traps to any host. The BGP traps are enabled for all hosts, but only the ISDN traps are enabled to be sent to a host.

```
snmp-server enable traps bgp
snmp-server host bob public isdn
```

The following example enables the router to send all inform requests to the host myhost.cisco.com using the community string named public:

```
snmp-server enable traps
snmp-server host myhost.cisco.com informs version 2c public
```

In the following example, the SNMP manager is enabled and the session timeout is set to a larger value than the default:

```
snmp-server manager
snmp-server manager session-timeout 1000
```

New MIB Features in Cisco IOS Release 12.2

This section outlines the new MIBs and MIB enhancements for the current Cisco IOS software release.

Circuit Interface Identification MIB

The Circuit Interface Identification MIB (also known as the Circuit Interface MIB) is a Cisco enterprise MIB used to assist in SNMP monitoring of circuit-based interfaces. The Circuit Interface MIB (CISCO-CIRCUIT-INTERFACE-MIB) provides a MIB object that can be used to identify individual circuit-based interfaces (for example, interfaces using ATM or Frame Relay). This user-specified identification will then be returned when linkup and linkdown SNMP traps are generated for the interface.

No Cisco IOS software configuration commands are associated with this MIB.

For more information, refer to the CISCO-CIRCUIT-INTERFACE-MIB.my file, available from the Cisco.com MIB website.

Ethernet-like Interfaces MIB

The Ethernet-like Interfaces MIB (ETHERLIKE-MIB) was introduced in Cisco IOS Release 12.1(2)T. The Cisco implementation of the Ethernet-like Interfaces MIB (defined in the ETHERLIKE-MIB.my and CISCO-ETHERLIKE-CAPABILITY.my files on the Cisco MIB website) complies with RFC 2665 (*Definitions of Managed Objects for the Ethernet-like Interface Types*), and Data Over Cable Service Interface Specification (DOCSIS) 1.0 requirements for Cable Modem Termination Systems (CMTSs) and cable modems (CMs). Support for RFC 2665 in the ETHERLIKE-MIB was achieved through the addition of two new objects in the *dot3StatsTable*: *dot3StatsSymbolErrors* and *dot3StatsDuplexStatus*.

No Cisco IOS software configuration commands are associated with this MIB.

Event MIB

The Event MIB was introduced in Cisco IOS Release 12.0(11)S and 12.1(3)T. No Cisco IOS software configuration commands are associated with this MIB. Instead, Event MIB configuration is done with applications external to Cisco IOS software. The Event MIB allows specialized monitoring capabilities that can be configured through a network management system (NMS) application using SNMP Get and Set operations. The Event MIB provides an asynchronous notification mechanism supported by SNMP that can be set to monitor any SNMP MIB object on a Cisco device and perform notification (trap or inform) operations or Set operations when specific conditions occur. Conditions are defined in event values. Event values that have been configured on your system can be displayed using the **show management event** command in privileged EXEC mode. By allowing SNMP notifications to take place only when a specified condition is met, Event MIB support reduces the load on affected devices, substantially improving the scalability of network management solutions.

For further information, see the Event MIB Support feature module document at http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t3/dtevent.htm

Expression MIB Support for Delta, Wildcarding, and Aggregation

Expression MIB adds support of the Delta, Wildcarding, Delta Wildcarding, and Aggregation features in the Distributed Management Expression MIB (EXPRESSION-MIB) to Cisco IOS software for use by SNMP. No Cisco IOS software configuration commands are associated with this MIB. The functionality provided by this MIB is especially useful when used with the Event MIB (described previously).

The Delta function enables the Expression MIB to use Delta values of an object instead of absolute values when evaluating an expression. Delta is obtained by taking the difference in the current value of an object with its previous value. Wildcarding empowers the Expression MIB to evaluate multiple instances of an object. This feature is useful in cases when the expression must be applied to all instances of an object. The user need not individually specify all instances of an object in the Expression but only needs to set the expWildcardedObject in expObjectTable to TRUE for the respective object. Aggregation is done by using the sum() function in the Expression MIB. The operand to the sum function must be a wildcarded object. The result of the sum() function is the sum of values of all instances of the wildcarded object.

For more information, see the *EXPRESSION-MIB.my* document available from the Cisco.com MIB website.

Interfaces Group MIB Enhancements

The Cisco implementation of the Interfaces Group MIB (IF-MIB) has been enhanced to allow you to enable linkUp and linkDown SNMP traps that are compliant with RFC 2233. The default implementation of linkUp and linkDown traps is defined in CISCO-IF-CAPABILITY.my and OLD-CISCO-INTERFACES- MIB.my. To enable linkUp and linkDown traps that will function for both interfaces and subinterfaces, use the **snmp-server trap link ietf** command in global configuration mode.

The IF-MIB implementation also has been enhanced to allow the consistent identification of interfaces using the Interface Index (ifIndex) value of the IF-MIB.

Interfaces Group MIB Support for ATM Subinterfaces

Introduced in Cisco IOS Release 12.1, the Interfaces Group MIB support for ATM subinterfaces feature provides the implementation of RFC 2233 (MIB-II) for ATM subinterfaces. ATM subinterfaces are visible in the ifTable and accessible to NMS applications. There are two entities in the ifTable corresponding to each subinterface—an atmSubif entity and an aal5 entity. The atmSubif entity corresponds to the ATM layer and the aal5 entity corresponds to the AAL5 layer. The MIB variables are defined in RFC 1695.

MIB Enhancements for Universal Gateways and Access Servers

The following MIB enhancements were designed to monitor modem and line status for network access servers (NASs).

CISCO-AAA-SERVER-MIB

The CISCO-AAA-SERVER-MIB provides statistics reflecting the state of authentication, authorization, and accounting (AAA) server operation within a device and AAA communications with external servers for the Cisco AS5300 and AS5800 series platforms. The Cisco AAA Server MIB provides the following information:

- A table for configuring AAA servers
- Identities of external AAA servers
- Statistics for each AAA function (**show radius statistics** command)
- Status of servers providing AAA functions

ServerStateChange notifications are controlled (enabled or disabled) through use of the **snmp-server enable traps aaa_server** command in global configuration mode. ServerStateChange notifications, when enabled, will be sent when the server moves from an “up” to “dead” state or when a server moves from a “dead” to “up” state.

Statistics for AAA functions can be displayed through use of the **show radius statistics** command in EXEC mode.

The implementation of this MIB is defined in the CISCO-AAA-SERVER-MIB.my and CISCO-AAA-SERVER-CAPABILITY.my files available from the Cisco.com MIB website.

CISCO-AAA-SESSION-MIB

The CISCO-AAA-SESSION-MIB provides the ability to both monitor and terminate authenticated client connections using SNMP for the Cisco AS5300 and AS5800 series platforms. Real-time information can be provided on data such as idle time, allowing configurations that can terminate calls when there are periods of inactivity on a line. Data provided by this MIB is directly related to the accounting information reported by AAA to RADIUS or TACACS servers. You can verify SNMP queried values through use of the **show accounting** and **show caller timeouts** commands in EXEC mode.

To enable the ability to terminate connections, you must configure the device through use of the **aaa session-mib** {**disconnect**} command in global configuration mode. When this command is found in a system configuration, SNMP managers have the ability to disconnect all lines that have AAA accounting

records associated to them using the Disconnect object. (AAA must already be configured with accounting enabled for this feature to function.) For more information, see the Release 12.2 *Cisco IOS Security Configuration Guide*.

CISCO-CALL-TRACKER-MIB, CISCO-CALL-TRACKER-MODEM-MIB, and CISCO-CALL-TRACKER-TCP-MIB

The CISCO-CALL-TRACKER-MIB, the CISCO-CALL-TRACKER-MODEM-MIB, and the CISCO-CALL-TRACKER-TCP-MIB provide the ability to capture detailed data on the progress and status of calls, from the time the NAS receives a setup request or allocates a channel, to the time a call is rejected or terminated. This data is maintained within the Call Tracker database tables, which are accessible through SNMP, command-line interface (CLI), or SYSLOG.

Call Tracker SNMP notifications are controlled through use of the **snmp-server enable traps snmp calltracker** command in global configuration mode. CallSetup notifications are generated at the start of each call, when an entry is created in the active table (cctActiveTable), and CallTerminate notifications are generated at the end of each call, when an entry is created in the history table (cctHistoryTable).

The Call Tracker feature is supported on the Cisco AS5300 and the Cisco AS5800 series platforms. For more information on this feature, see the *Call Tracker plus ISDN and AAA Enhancements for the Cisco AS5300 and Cisco AS5800* document available from Cisco.com.

CISCO-ISDN-MIB

The CISCO-ISDN-MIB supplies ISDN PRI channel-not-available traps that can be generated when a requested DS 0 channel is not available, or when no modem is available to take the incoming call. ISDN PRI channel-not-available notifications are controlled (enabled or disabled) through use of the **no snmp-server enable traps isdn** [**chan-not-avail**] command in global configuration mode. These notifications are disabled by default and are available only for ISDN PRI interfaces on the Cisco AS5300, Cisco AS5400, and Cisco AS5800 universal access servers.

CISCO-MODEM-MGMT-MIB

The CISCO-MODEM-MGMT-MIB supplies modem health traps that can be generated when a modem port is bad, disabled, reflashed, or shut down, or when there is a request to busyout the modem. Modem health notifications are controlled (enabled or disabled) through use of the **no snmp-server enable traps modem-health** command in global configuration mode. Modem health traps are disabled by default and are supported on the Cisco AS5300, Cisco AS5400, and Cisco AS5800 universal access servers.

CISCO-POP-MGMT-MIB

The CISCO-POP-MGMT-MIB supplies the DS 0 busyout notification. DS 0 busyout traps or informs can be generated when there is a request to busyout a DS 0, when there is a request to take a DS 0 out of busyout mode, or when busyout completes and the DS 0 is out of service. DS 0 busyout traps are controlled (enabled or disabled) through use of the **no snmp-server enable traps pop** command in global configuration mode. Busyout is enabled on a device using the **isdn snmp busyout b-channel** command. DS 0 busyout notifications are disabled by default and are supported on Cisco AS5300, Cisco AS5400, and Cisco AS5800 universal access servers.

DS 1 loopback traps can be generated when a DS 1 line goes into loopback mode. DS 1 loopback traps are controlled (enabled or disabled) through use of the **no snmp-server enable traps ds1-loopback** command in global configuration mode. DS 1 loopback traps are disabled by default and are supported only on the Cisco AS5300 and Cisco AS5400 universal access servers.

RFC1406-MIB

The RCF1406-MIB supplies dsx1LineStatus and dsx1LineIndex objects.

MSDP MIB

The Multicast Source Discovery Protocol (MSDP) MIB feature adds support in Cisco IOS software for the MSDP MIB. This MIB describes objects used for managing MSDP operations using SNMP. MSDP MIB notifications are controlled (enabled or disabled) through use of the **no snmp-server enable traps msdp** command in global configuration mode. There are two MSDP MIB notification-types: msdpEstablished (1) and msdpBackwardTransition (2). The msdpEstablish notifications are sent when the MSDP finite state machine (FSM) enters the ESTABLISHED state. The msdpBackwardTransition notifications are sent generated when the MSDP FSM moves from a higher numbered state to a lower numbered state. For more information on the Cisco implementation of the MSDP MIB, refer to the *MSDP-MIB.my* document available from Cisco.com. The Cisco implementation of the MSDP MIB has the following restrictions in Cisco IOS Release 12.2:

- All MSDP MIB objects are implemented as read-only.
- The Requests table is not supported in the Cisco implementation of the MSDP MIB.
- The msdpEstablished notification is not supported in the Cisco implementation of the MSDP MIB.

NTP MIB

The Network Time Protocol (NTP) is used to synchronize timekeeping among a set of distributed time servers and clients. The Cisco NTP MIB enables users to remotely monitor an NTP server using SNMP, provided the MIB itself is implemented on that server. Use of the NTP MIB to monitor the NTP status of routing devices is accomplished using software on an NMS. No new or modified Cisco IOS software commands are associated with this feature.

The Cisco implementation of the NTP MIB is based on NTP version 3 (RFC-1305). The MIB objects are all read-only. SNMP requests are processed by reading the corresponding variables from the NTP subsystem and returning them in the response. The NTP MIB defines a set of NTP server system objects, including an NTP server peers table and an NTP server filter register table. For more information on the Cisco implementation of the NTP MIB, refer to the MIB document itself (*CISCO-NTP-MIB.my*, available from Cisco.com).

Response Time Monitor MIB

The CISCO-RTTMON-MIB is used for network monitoring and management using the Cisco Service Assurance Agent (SA Agent). For information about the enhancements to this MIB, see the "Network Monitoring Using Cisco Service Assurance Agent" chapter in this document.

Configuring Cisco Discovery Protocol

This chapter describes how the Cisco Discovery Protocol (CDP) works with Simple Network Management Protocol (SNMP) to identify other devices in your network in Cisco IOS Release 12.2.

For further details on the commands mentioned in this chapter, refer to the *Cisco IOS Configuration Fundamentals Command Reference*.

Configuring the Cisco Discovery Protocol

CDP is primarily used to obtain protocol addresses of neighboring devices and discover the platform of those devices. CDP can also be used to display information about the interfaces your router uses. CDP is media- and protocol-independent, and runs on all Cisco-manufactured equipment including routers, bridges, access servers, and switches.

Use of SNMP with the CDP MIB allows network management applications to learn the device type and the SNMP agent address of neighboring devices, and to send SNMP queries to those devices. CDP uses the CISCO-CDP-MIB.

CDP runs on all media that support Subnetwork Access Protocol (SNAP), including LAN, Frame Relay, and ATM physical media. CDP runs over the data link layer only. Therefore, two systems that support different network-layer protocols can learn about each other.

Each device configured for CDP sends periodic messages, known as advertisements, to a multicast address. Each device advertises at least one address at which it can receive SNMP messages. The advertisements also contain time-to-live, or holdtime, information, which indicates the length of time a receiving device should hold CDP information before discarding it. Each device also listens to the periodic CDP messages sent by others in order to learn about neighboring devices and determine when their interfaces to the media go up or down.

CDP Version-2 (CDPv2) is the most recent release of the protocol and provides more intelligent device tracking features. These features include a reporting mechanism that allows for more rapid error tracking, thereby reducing costly downtime. Reported error messages can be sent to the console or to a logging server, and cover instances of unmatching native VLAN IDs (IEEE 802.1Q) on connecting ports, and unmatching port duplex states between connecting devices. Refer tothe *Cisco IOS Software System Error Messages* document for detailed examples of CDP error messages.

CDPv2 **show** commands can provide detailed output on VLAN Trunking Protocol (VTP) management domain and duplex modes of neighbor devices, CDP-related counters, and VLAN IDs of connecting ports.

VTP is a discovery technique deployed by switches where each switch advertises its management domain on its trunk ports, its configuration revision number, and its known VLANs and their specific parameters. A VTP domain is made up of one or more interconnected devices that share the same VTP domain name. A switch can be configured to be in only one VTP domain.

Type-Length-Value fields (TLVs) are blocks of information embedded in CDP advertisements. Table 19 summarizes the TLV definitions for CDP advertisements.

Table 21 Type-Length-Value Definitions for CDPv2

TLV	Definition
Device-ID TLV	Identifies the device name in the form of a character string.
Address TLV	Contains a list of network addresses of both receiving and sending devices.
Port-ID TLV	Identifies the port on which the CDP packet is sent.
Capabilities TLV	Describes the functional capability for the device in the form of a device type, for example, a switch.
Version TLV	Contains information about the software release version on which the device is running.
Platform TLV	Describes the hardware platform name of the device, for example, Cisco 4500.
IP Network Prefix TLV	Contains a list of network prefixes to which the sending device can forward IP packets. This information is in the form of the interface protocol and port number, for example, Eth 1/0.
VTP Management Domain TLV	Advertises the system's configured VTP management domain name-string. Used by network operators to verify VTP domain configuration in adjacent network nodes.
Native VLAN TLV	Indicates, per interface, the assumed VLAN for untagged packets on the interface. CDP learns the native VLAN for an interface. This feature is implemented only for interfaces that support the IEEE 802.1Q protocol.
Full/Half Duplex TLV	Indicates status (duplex configuration) of CDP broadcast interface. Used by network operators to diagnose connectivity problems between adjacent network elements.

CDP Configuration Task List

To configure CDP, perform any of the optional tasks in the following sections:

- Setting the CDP Transmission Timer and Hold Time
- Reenabling CDP on a Local Router
- Reenabling CDP Version-2 Advertisements
- Reenabling CDP on an Interface
- Monitoring and Maintaining CDP

The the end of this chapter for "CDP Configuration Examples."

Note The **cdp enable**, **cdp timer**, and **cdp run** global configuration commands affect the operation of the IP on-demand routing feature (that is, the **router odr** global configuration command). For more information on the **router odr** command, see the "On-Demand Routing Commands" chapter in the Release 12.2 *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols* document.

Setting the CDP Transmission Timer and Hold Time

To set the frequency of CDP transmissions and the hold time for CDP packets, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# cdp timer` *seconds*	Specifies frequency of transmission of CDP updates.
Step 2	`Router(config)# cdp holdtime` *seconds*	Specifies the amount of time a receiving device should hold the information sent by your device before discarding it.

Reenabling CDP on a Local Router

CDP is enabled on Cisco devices by default. If you prefer not to use the CDP device discovery capability, you can disable it with the **no cdp run** command.

To reenable CDP after disabling it, use the following command in global configuration mode:

Command	Purpose
`Router(config)# cdp run`	Enables CDP on the system.

Reenabling CDP Version-2 Advertisements

The broadcasting of CDPv2 advertisements is enabled on Cisco routers by default. You can disable CDPv2 advertisements with the **no cdp advertise-v2** command.

To reenable CDPv2 advertisements, use the following command in global configuration mode:

Command	Purpose
`Router(config)# cdp advertise-v2`	Enables CDPv2 advertising functionality on the system.

Reenabling CDP on an Interface

CDP is enabled by default on all supported interfaces (except for Frame Relay multipoint subinterfaces) to send and receive CDP information. However, some interfaces, such as ATM interfaces, do not support CDP.

You can disable CDP on an interface that supports CDP by using the **no cdp enable** command.

To reenable CDP on an interface after disabling it, use any of the following command in interface configuration mode, as needed:

Command	Purpose
`Router(config-if)# cdp enable`	Enables CDP on an interface.

Monitoring and Maintaining CDP

To monitor and maintain CDP on your device, use one or more of the following commands in privileged EXEC mode:

Command	Purpose
`Router# clear cdp counters`	Resets the traffic counters to zero.
`Router# clear cdp table`	Deletes the CDP table of information about neighbors.
`Router# show cdp`	Displays the interval between transmissions of CDP advertisements, the number of seconds the CDP advertisement is valid for a given port, and the version of the advertisement.
`Router# show cdp entry` *device-name* [**protocol** | **version**]	Displays information about a specific neighbor. Display can be limited to protocol or version information.
`Router# show cdp interface` [**type number**]	Displays information about interfaces on which CDP is enabled.
`Router# show cdp neighbors` [*type number*] [**detail**]	Displays the type of device that has been discovered, the name of the device, the number and type of the local interface (port), the number of seconds the CDP advertisement is valid for the port, the device type, the device product number, and the port ID. Issuing the **detail** keyword displays information on the native VLAN ID, the duplex mode, and the VTP domain name associated with neighbor devices.
`Router# show cdp traffic`	Displays CDP counters, including the number of packets sent and received and checksum errors.
`Router# show debugging`	Displays information about the types of debugging that are enabled for your router. Refer to the *Cisco IOS Debug Command Reference* for more information about CDP **debug** commands.

CDP Configuration Examples

The following sections provide CDP configuration examples:

- Example: Setting the CDP Transmission Timer and Hold Time
-

Example: Setting the CDP Transmission Timer and Hold Time

In the following example, the user sets the cdp timer to send updates every 30 seconds to neighboring routers and sets the router to show that the updates are working correctly:

```
Router(config)# cdp timer 30
Router(config)# exit
Router# show cdp interface
Serial0 is up, line protocol is up
Encapsulation is HDLC
Sending CDP packets every 30 seconds
Holdtime is 180 seconds
```

In the following example, the user sets the holdtime to be 90 seconds and sets the router to show that the updates are working correctly:

```
Router(config)# cdp holdtime 90
Router(config)# exit
Router# show cdp interface
Serial0 is up, line protocol is up
Encapsulation is HDLC
Sending CDP packets every 30 seconds
Holdtime is 90 seconds
```

Example: Monitoring and Maintaining CDP

The following example shows a typical series of steps for viewing information about CDP neighbors.

Table 20 describes the significant fields shown in the output of the **show cdp neighbors** command.

```
C3660-2> show cdp
Global CDP information:
        Sending CDP packets every 60 seconds
        Sending a holdtime value of 180 seconds
        Sending CDPv2 advertisements is  enabled

C3660-2> show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
C2950-1          Fas 0/0            148         S I       WS-C2950T-Fas 0/15
RX-SWV.cisco.com Fas 0/1            167         T S       WS-C3524-XFas 0/13

C3660-2> show cdp neighbors detail
-------------------------
Device ID: C2950-1
Entry address(es):
Platform: Cisco WS-C2950T-24,  Capabilities: Switch IGMP
Interface: FastEthernet0/0,  Port ID (outgoing port): FastEthernet0/15
Holdtime : 139 sec

Version :
Cisco IOS C2950 Software (C2950-I6Q4L2-M), Version 12.1(9)EA1, RELEASE SOFTWARE
 .
 .
 .
C3660-2> show cdp traffic
CDP counters :
        Total packets output: 81684, Input: 81790
        Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
```

```
        No memory: 0, Invalid packet: 0, Fragmented: 0
        CDP version 1 advertisements output: 0, Input: 0
        CDP version 2 advertisements output: 81684, Input: 81790
C3660-2>
```

Table 20 describes the significant fields shown in the output of the show cdp neighbors command.

Table 22 show cdp neighbors Field Descriptions

Field	Definition
Device ID	The name of the neighbor device and either the MAC address or the serial number of this device.
Local Intrfce	The protocol being used by the connectivity media.
Holdtme	The remaining amount of time (in seconds) the current device will hold the CDP advertisement from a sending router before discarding it.
Capability (Capability Codes)	Capability (type of routing device) of the listed neighboring device. The capability types that can be discovered are: R—Router T—Transparent bridge B—Source-routing bridge S—Switch H—Host I— device is using IGMP r—Repeater
Platform	The product number of the device.
Port ID	The protocol and port number of the device.

Configuring RMON Support

This chapter describes the Remote Monitoring (RMON) MIB agent specification, and how it can be used in conjunction with Simple Network Management Protocol (SNMP) to monitor traffic using alarms and events.

For a complete description of the RMON commands mentioned in this chapter, refer to the "RMON Commands" chapter in the "System Management" part of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Configuring RMON Support

The RMON option identifies activity on individual nodes and allows you to monitor all nodes and their interaction on a LAN segment. Used in conjunction with the SNMP agent in a router, RMON allows you to view both traffic that flows through the router and segment traffic not necessarily destined for the router. Combining RMON alarms and events (classes of messages that indicate traffic violations and various unusual occurrences over a network) with existing MIBs allows you to choose where proactive monitoring will occur.

Note Full RMON packet analysis (as described in RFC 1757) is supported only on an Ethernet interface of Cisco 2500 series routers and Cisco AS5200 series universal access servers. RMON requires that SNMP be configured (you must be running a version of SNMP on the server that contains the RMON MIB). A generic RMON console application is recommended in order to take advantage of the RMON network management capabilities. This feature supports RFCs 1757 and 2021.

RMON can be very data- and processor-intensive. Users should measure usage effects to ensure that router performance is not degraded by RMON and to minimize excessive management traffic overhead. Native mode in RMON is less intensive than promiscuous mode.

All Cisco IOS software images ordered without the explicit RMON option include limited RMON support (RMON alarms and event groups only). Images ordered with the RMON option include support for all nine management groups (statistics, history, alarms, hosts, hostTopN, matrix, filter, capture, and event). As a security precaution, support for the capture group allows capture of packet header information only; data payloads are not captured.

In Cisco IOS 12.1, the RMON agent was rewritten to improve performance and add some new features. Table 23 highlights some of the improvements implemented.

Table 23 RMON MIB Updates

Prior to the RMON MIB Update in Cisco IOS Release 12.1	New functionality in Cisco IOS Release 12.1
RMON configurations do not persist across reboots. Information is lost after a new session on the RMON server.	RMON configurations persist across reboots. Information is preserved after a new session on the RMON server.
Packet analysis applies only on the MAC header of the packet.	Complete packet capture is performed with analysis applied to all frames in packet.
Only RMON I MIB objects are used for network monitoring.	RMON I and selected RMON II objects are used for network monitoring.

RMON MIB features include the following:

- usrHistory group. This MIB group is similar to the RMON etherHistory group except that the group enables the user to specify the MIB objects that are collected at each interval.
- partial probeConfig group. This MIB group is a subset of the probeConfig group implemented in read-only mode. These objects implement the simple scalars from this group. Table 24 details new partial probeConfig group objects.

Table 24 partial probeConfig Group Objects

Object	Description
probeCapabilities	The RMON software groups implemented.
probeSoftwareRev	The current version of Cisco IOS running on the device.
probeHardwareRev	The current version of the Cisco device.
probeDateTime	The current date and time.
probeResetControl	Initiates a reset.
probeDownloadFile	The source of the image running on the device.
probeDownloadTFTPServer	The address of the server that contains the Trivial File Transfer Protocol (TFTP) file that is used by the device to download new versions of Cisco IOS software.
probeDownloadAction	Specifies the action of the commands that cause the device to reboot.
probeDownloadStatus	The state of a reboot.
netDefaultGateway	The router mapped to the device as the default gateway.
hcRMONCapabilities	Specifies the features mapped to this version of RMON.

Configuring RMON Alarm and Event Notifications

To enable RMON on an Ethernet interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# rmon {native | promiscuous}`	Enables RMON.

In native mode, RMON monitors only the packets normally received by the interface. In promiscuous mode, RMON monitors all packets on the LAN segment.

The default size of the queue that holds packets for analysis by the RMON process is 64 packets. To change the size of the queue, use the following command in global configuration mode:

Command	Purpose
`Router(config)# rmon queuesize size`	Changes the size of the RMON queue.

To set an RMON alarm or event, us the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)# rmon alarm number variable interval {delta | absolute} rising-threshold value [event-number] falling-threshold value [event-number] [owner string]`	Sets an alarm on a MIB object.
`Router(config)# rmon event number [log] [trap community] [description string] [owner string]`	Adds or removes an event in the RMON event table.

You can set an alarm on any MIB object in the access server. To disable an alarm, you must enable the **no** form of this command on each alarm you configure. You cannot disable all the alarms you configure at once. Refer to RFC 1757 to learn more about alarms and events and how they interact with each other.

The RMON MIB defines two traps, the risingAlarm and fallingAlarm traps generated when an RMON alarmEntry risingThreshold or fallingThreshold event occurs. Thresholds allow you to minimize the number of notifications sent on the network. Alarms are triggered when a problem exceeds a set rising threshold value. No more alarm notifications are sent until the agent recovers, as defined by the falling threshold value. This means that notifications are not sent each time a minor failure or recovery occurs.

Configuring RMON Groups

RMON tables can be created for buffer capture, filter, hosts, and matrix information. The buffer capture table details a list of packets captured off a channel or a logical data or events stream. The filter table details a list of packet filter entries that screen packets for specified conditions as they travel between interfaces. The hosts table details a list of host entries. The matrix table details a list of traffic matrix entries indexed by source and destination MAC addresses.

To gather RMON statistics for these data types, use the following commands in interface configuration mode, as needed:

Command	Purpose
`Router(config-if)# rmon collection history {controlEntry integer} [owner ownername] [buckets bucket-number] [interval seconds]`	Enables an RMON history group of statistics on an interface.
`Router(config-if)# rmon collection host {controlEntry integer} [owner ownername]`	Enables an RMON host collection group of statistics on an interface.
`Router(config-if)# rmon collection matrix {controlEntry integer} [owner ownername]`	Enables an RMON matrix group of statistics on an interface.
`Router(config-if)# rmon collection rmon1 {controlEntry integer} [owner ownername]`	Enables all possible autoconfigurable RMON statistic collections on an interface.

To specifically monitor these commands, use the **show rmon capture**, **show rmon filter**, **show rmon hosts**, and **show rmon matrix** EXEC commands listed in the following table.

Monitoring and Verifying RMON Configuration

To display the current RMON status, use one or more of the following commands in EXEC mode:

Command	Function

`Router> show rmon`
or
`Router> show rmon task`
	Displays general RMON statistics.

`Router> show rmon alarms`	Displays the RMON alarm table.
`Router> show rmon capture`	Displays the RMON buffer capture table and current configuration. Available only on Cisco 2500 series routers and Cisco AS5200 access servers.
`Router> show rmon events`	Displays the RMON event table.
`Router> show rmon filter`	Displays the RMON filter table. Available only on Cisco 2500 series routers and Cisco AS5200 access servers.
`Router> show rmon history`	Displays the RMON history table. Available only on Cisco 2500 series routers and Cisco AS5200 access servers.
`Router> show rmon hosts`	Displays the RMON hosts table. Available only on Cisco 2500 series routers and Cisco AS5200 access serverss.
`Router> show rmon matrix`	Display the RMON matrix table and values associated with RMON variables. Available only on Cisco 2500 series routers and Cisco AS5200 access servers.
`Router> show rmon statistics`	Display the RMON statistics table. Available only on Cisco 2500 series routers and Cisco AS5200 access servers.
`Router> show rmon topn`	Display the RMON top-n hosts table. Available only on Cisco 2500 series routers and Cisco AS5200 access servers.

RMON Configuration Examples

This section provides the following examples:

- Alarm and Event Example
- show rmon Command Example

Alarm and Event Example

The following example enables the **rmon event** global configuration command:

```
Router(config)# rmon event 1 log trap eventtrap description "High ifOutErrors" owner
owner_a
```

This example creates RMON event number 1, which is defined as "High ifOutErrors", and generates a log entry when the event is triggered by an alarm. The user "owner_a" owns the row that is created in the event table by this command. This example also generates an SNMP trap when the event is triggered.

The following example configures an RMON alarm using the **rmon alarm** global configuration command:

```
Router(config)# rmon alarm 10 ifEntry.20.1 20 delta rising-threshold 15 1
falling-threshold 0 owner owner_a
```

This example configures RMON alarm number 10. The alarm monitors the MIB variable ifEntry.20.1 once every 20 seconds until the alarm is disabled, and checks the change in the rise or fall of the variable. If the ifEntry.20.1 value shows a MIB counter increase of 15 or more, such as from 100000 to 100015, the alarm is triggered. The alarm in turn triggers event number 1, which is configured with the **rmon event** command. Possible events include a log entry or an SNMP trap. If the ifEntry.20.1 value changes by 0, the alarm is reset and can be triggered again.

show rmon Command Example

To display the RMON buffer capture table and current configuration, enter the **show rmon capture** EXEC command (Cisco 2500 series routers and Cisco AS5200 access servers only). A sample configuration follows:

```
Router# show rmon capture

Buffer 1 is active, owned by John Smith
Captured data is from channel 1
Slice size is 128, download size is 128
Download offset is 0
Full Status is full, full action is wrapWhenFull
Granted -1 octets out of -1 requested
Buffer has been on since 18:59:48, and has captured 522 packets
Current capture buffer entries:
Packet 3271 was captured 2018256 ms since buffer was turned on
Its length is 184 octets and has a status type of 0
Packet ID is 3721, and contains the following data:
03 00 00 00 00 0100 A0CC 3C9D DF00 A6F0 03
Packet 3722 was captured 2018452 ms since buffer was turned on
Its length is 64 octets and has a status type of 0
Packet ID is 3722, and contains the following data:
01 80C2 0000 0000 6009 FDFE D300 2642 03
```

To view values associated with RMON variables, enter the **show rmon matrix** EXEC command (Cisco 2500 series routers and Cisco AS5200 access servers only). The following is a sample output:

```
Router# show rmon matrix

Matrix 1 is active and owned by
Monitors ifEntry.1.1
Table size is 42, last time an entry was deleted was at 11:18:09
Source addr is 0000.0c47.007b, dest addr is ffff.ffff.ffff
Transmitted 2 pkts, 128 octets, 0 errors
Source addr is 0000.92a8.319e, dest addr is 0060.5c86.5b82
Transmitted 2 pkts, 384 octets, 1 error
```

For an explanation of the fields in the examples, refer to the respective command descriptions in the "RMON Commands" chapter of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*.

Network Monitoring Using Cisco Service Assurance Agent

This chapter describes how to configure the Cisco Service Assurance Agent (SAA) to provide advanced network service monitoring information using Cisco IOS Software Release 12.2. This chapter contains the following sections:

- Understanding the Cisco SAA
- Cisco SAA Configuration Task List
- SAA Configuration Using the CLI Examples
- SAA Configuration Using SNMP Examples

For a complete description of the Cisco SAA configuration commands mentioned in this chapter, see the "Cisco Service Assurance Agent Commands" chapter of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Understanding the Cisco SAA

The Cisco SAA is an application-aware synthetic operation agent that monitors network performance by measuring response time, network resource availability, application performance, jitter (interpacket delay variance), connect time, throughput, and packet loss. Performance can be measured between any Cisco device that supports this feature and any remote IP host (server), Cisco routing device, or mainframe host. Performance measurement statistics provided by this feature can be used for troubleshooting, for problem analysis, and for designing network topologies.

The Cisco SAA can be especially useful for enterprise and service provider networks, because it provides expanded measurement and management capabilities. In particular, the SAA is a reliable mechanism for accurately monitoring the metrics in service level agreements (SLAs).

Because SAA is accessible using Simple Network Management Protocol (SNMP), it also can be used in performance monitoring applications for Network Management Systems (NMSs) such as CiscoWorks2000 (CiscoWorks Blue) and the Internetwork Performance Monitor (IPM). SAA notifications also can be enabled via Systems Network Architecture (SNA) network management vector transport (NMVT) for applications such as NetView.

SNMP notifications based on the data gathered by the SAA allow the router to receive alerts when performance drops below a specified level and when problems are corrected. The SAA utilizes the Cisco Round Trip Time Monitor (RTTMON) MIB for interaction between external NMS applications and the SAA running on the Cisco devices. For a complete description of the object variables referenced by the SAA feature, refer to the text of the CISCO-RTTMON-MIB.my file, available from the Cisco MIB website.

New Features in Cisco IOS Release 12.2

Cisco IOS Release 12.2 includes new SAA capabilities that allow you to perform the following tasks:

- Measure file transfer protocol (FTP) file download time using the FTP operation
- Monitor one-way latency reporting through enhancements to the Jitter operation
- Configure a new option for the Dynamic Host Configuration Protocol (DHCP) operation
- Manually enable a responder port (using the **rtr responder** global configuration command)
- Verify data for the udpEcho operation (using the **verify-data** RTR configuration command)
- Configure new options for the **rtr schedule** global configuration ommand
- Restart an operation

Cisco IOS Release 12.2 adds the following enhancements to the RTTMON MIB:

- Addition of rttMonAuthTable (allows the user to configure authentication strings)
- Extensions of the rttMonJitterStatsTable and the rttMonLatestJitterOperTable
- Addition of rttMonEchoAdminMode for FTP operation
- Extension of the rttMonAppl table (allows the user to enable the SAA responder using the MIB)

Cisco SAA Configuration Task List

To configure Cisco SAA, perform the tasks described in the following sections:.

- Configuring SAA Operations (Required)
- Configuring the Operation Type (Required)
- Configuring SAA Operation Characteristics (Optional)
- Scheduling the Operation (Required)
- Enabling the SAA Responder on Operational Targets (Required for certain operations)
- Configuring SAA Control Message Authentication (Optional)
- Resetting the SAA (Optional)
- Restarting a Stopped Operation (Optional)
- Displaying SAA Status and SAA Operational Results (Required)
- Changing the Memory Threshold for the SAA (Optional)
- Configuring Specific Operations (Optional)
- Configuring SAA Operations Using SNMP (Optional)
- Accessing SAA Data Using SNMP (Optional)

- Enabling SAA SNMP Notifications (Optional)

Examples of the Cisco IOS CLI configuration tasks are provided in the "SAA Configuration Using the CLI Examples" section. Examples of SNMP configuration tasks are provided in the "SAA Configuration Using SNMP Examples" section.

Note SAA is an expansion of the Response Time Reporter (RTR) feature introduced in Cisco IOS Release 11.2. SAA retains the use of the RTR acronym in many of the configuration commands, and for the configuration mode used to configure SAA operations. RTR is also used throughout the command line interface (CLI) in the output of **help** and **show** commands.

Configuring SAA Operations

Response time and availability information is collected by *operations* that you configure on the router. Operations use synthetic packets specifically placed in a network to collect data about the network. These packets simulate other forms of network traffic, as determined by the type of operation you configure. Operations usually consist of multiple probe packets sent into the network; operations in general can be thought of as collections of probes.

SAA operations are given specific identification numbers so you can track the various operations you configure and execute. SAA operations are configured in RTR configuration mode. To configure an SAA operation, use the **rtr** global configuration command. When using this command, you specify the identification number for the operation you are about to configure. The router prompt will change to `(config-rtr)` to indicate that you are in RTR configuration mode.

To configure a new SAA operation, perform the following steps beginning in global configuration mode:

Step 1 Enter RTR configuration mode using the **rtr** *operation-number* global configuration command. The *operation-number* argument specifies an identification number for the operation you will be configuring.

Step 2 Use one of the **type** commands listed in the "Configuring the Operation Type" section to specify which type of operation you are configuring.

Step 3 (Optional) Configure characteristics for the operation, one characteristic per line, using the commands found in "Configuring SAA Operation Characteristics" section.

Step 4 Type **exit** to return to global configuration mode.

Step 5 (Optional) Set reaction conditions for the operation, as described in the "Setting Reaction Thresholds."

Step 6 Schedule the operation start time, as described in the "Scheduling the Operation" section.

For an example of this process, see the "IP/ICMP Path Echo Example" found in the "SAA Configuration Using the CLI Examples" section.

Configuring the Operation Type

You must configure the operation type before you can configure any of the other characteristics. Cisco SAA provides the types of operations:

Operation Type	Function	RTR Configuration Command[1]
IP/ICMP Echo	The IP/Internet Control Message Protocol (ICMP) Echo operation measures end-to-end response time between a Cisco router and devices using IP. ICMP is a network-layer Internet protocol that reports errors and provides other information relevant to IP packet processing. Response time is computed by measuring the time taken between sending an ICMP echo request message to the destination and receiving an ICMP echo reply. Cisco SAA provides an option to compute response time on a specific path by using the Loose Source Routing option in IP packets. SAA also allows a user to measure quality of service (QoS) between endpoints by setting type of service (ToS) bits on an IP packet. The Loose Source Routing path that an IP/ICMP Echo operation should take can be set using the **lsr-path** RTR configuration command.	**type echo protocol ipIcmpEcho**
SNA Echo	The Systems Network Architecture (SNA) Echo operation measures end-to-end response time between a Cisco router and devices using SNA. You can use the SNA system services control points Native Echo (SSCP-RU), or you can target SNA LU type 0 connections or SNA logical unit (LU) type 2 connections that use the Cisco NSPECHO host application.	**type echo protocol snaRUEcho** or **type echo protocol snaLU0EchoAppl** or **type echo protocol snaLU2EchoAppl**
IP / ICMP Path Echo	The Path Echo operations record statistics for each hop along the path that the operation takes to reach it destination. The IP/ICMP Path Echo probe computes this hop-by-hop response time between a Cisco router and any IP device on the network by discovering the path using traceroute. Typical usage of this type of operation is to isolate bottlenecks in a path. **Note** Loose Source Routing (lsr) option is not available for this operation.	**type pathEcho protocol IpIcmpEcho**
TCP Connection	The Transmission Control Protocol (TCP) Connection operation is used to discover the time taken to connect to the target device. This operation can be used to test virtual circuit availability or application availability. If the target is a Cisco router, then SAA makes a TCP connection to any port number specified by the user. If the destination is a non-Cisco IP host, then the user must specify a known target port number (for example, 21 for FTP, 23 for Telnet, or 80 for HTTP server). This operation is useful in testing Telnet or HTTP connection times.	**type tcpConnect**

Operation Type	Function	RTR Configuration Command[1]
UDP Echo	The User Datagram Protocol (UDP) Echo operation calculates UDP response times between a Cisco router and any IP-enabled device. Response time is computed by measuring the time taken to send a datagram and receive a response from the destination device (round-trip time). If the target is a Cisco router, then SAA sends a UDP datagram to any port number specified by the user. If the destination is a non-Cisco IP host, then the user must specify a known target port number.	**type udpEcho**
Jitter/UDP Plus	The UDP Plus operation is a superset of the UDP Echo operation. In addition to measuring UDP round-trip time, the UDP Plus operation measures per-direction packet-loss, one-way delay time, and jitter. Jitter is the delay variance between received packets. Packet loss is a critical element in SLAs, and jitter statistics are useful for analyzing traffic in a voice over IP (VoIP) network. Packet loss is reported for how many packets are lost, and in which direction (source to destination or destination to source). Delay is also reported for each direction.	**type jitter**
HTTP	The HTTP operation measures the Round Trip Time (RTT) taken to connect and access data from an HTTP server. The HTTP server response time measurements consist of three types: • DNS lookup—RTT taken to perform domain name lookup. • TCP connect—RTT taken to perform a TCP connection to the HTTP server. • HTTP transaction time—RTT taken to send a request and get a response from the HTTP server (the probe retrieves the base HTML page only). For a GET request, the SAA will format the request based on the URL specified. In application self-service mode, the application controlling this probe is responsible for specifying the content of the HTTP request. SAA HTTP RAW operations allow the use of the **http-raw-request** Cisco IOS configuration submode. The SAA will send the HTTP request, receive the reply, and report RTT statistics (including the size of the page returned).	**type http operation get** or **type http operation raw**
FTP	The FTP operation throughput probe measures the time taken to transfer (download) a file from a remote host to the Cisco router using FTP (over TCP). (To test only how long taken to connect to a FTP port (port 21), use the TCP Connection operation.)	**type ftp**
DHCP	The SAA DHCP operation measures the round-trip time taken to discover a DHCP server and obtain an IP address lease from it. After obtaining an IP address, the SAA releases the IP address that was leased by the server. The default DHCP operation sends discovery packets from every available IP interface. However, if a specific DHCP server address is configured for use on the system using the **ip dhcp-server** *ipaddress* global configuration command, then discovery packets will be sent only to that DHCP server.	**type dhcp**

Operation Type	Function	RTR Configuration Command[1]
DLSw+	Data-link switching plus (DLSw+) is the enhanced Cisco version of RFC 1795. The DLSw+ operation tunnels LAN traffic over IP backbones via TCP. Many enterprise customers use the DLSw+ technology to seamlessly connect LAN media over geographically disperse locations. The routers performing the tunneling of LAN traffic into TCP/IP are referred to as DLSw peers. The SAA DLSw+ probe measures the DLSw+ protocol stack and network response time between DLSw peers. Normally DLSw peers communicate through TCP port 2065. A prerequisite to successfully running the SAA DLSw+ probe is to have a connected DLSw+ peer between the source and destination Cisco devices. On the source DLSw+ device, a probe can be defined for a DLSw+ partner peer that need not be running a Cisco image that contains SAA functionality.	**type dlsw**
DNS	The Domain Name System (DNS) operation response time is computed by measuring the difference between the time taken to send DNS request and receive a reply. The operation queries for an IP address if the user specifies a host name or queries for a host name if the user specifies an IP address.	**type dns**

1. For complete command syntax, use the ? (help) feature of the CLI, or refer to the *Cisco IOS Configuration Fundamentals Command Reference.*

Configuring SAA Operation Characteristics

To configure characteristics for SAA operations, perform the tasks described in the following sections:

- Setting General Operational Characteristics
- Enabling Data Verification for Operations
- Setting Statistics Gathering Characteristics
- Setting History Characteristics
- Setting Reaction Thresholds

Setting General Operational Characteristics

To configure optional characteristics for Cisco SAA operations, use the following commands in RTR configuration mode:

Command	Purpose
`Router(config-rtr)# frequency seconds`	Sets how often the operation should send a probe out to gather statistics. This command applies to all operation types.
`Router(config-rtr)# lsr-path {name | ipaddr} [name | ipaddr]...`	Defines a LSR path for an IP/ICMP echo probe. This command applies only to IP/ICMP Echo operations. **Note** LSR paths can be specified for IP/ICMP Echo operations, but not for IP/ICMP PathEcho operations.

Command	Purpose
`Router(config-rtr)# owner text`	Configures the SNMP owner of the operation. This command applies to all operation types.
`Router(config-rtr)# request-data-size bytes`	Sets the protocol data size in the payload of the probe request packet of the probe. This command applies to the following operation types: IP/ICMP Echo, UDP Echo, Jitter, DLSW, and SNA Echo
`Router(config-rtr)# response-data-size bytes`	Sets the protocol data size in the payload of the response packet of the operation. This command applies only to SNA Echo operations.
`Router(config-rtr)# tag text`	Logically links operations in a group. This command applies to all operations.
`Router(config-rtr)# timeout milliseconds`	Sets the amount of time the probe waits for a response from its request packet. This command applies to all operations.
`Router(config-rtr)# tos number`	Defines the IP ToS byte for request packets. This command applies to the following operation types: IP/ICMP Echo, UDP Echo, and Jitter.

Enabling Data Verification for Operations

If you suspect that data corruption is occurring for operations, you can enable data verification. To verify data for a previously configured operation, use the following commands, starting in global configuration mode:

Command	Purpose
`Router(config)# rtr operation_id`	Specifies the operation number of the SAA operation you want to configure.
`Router(config-rtr)# verify-data`	Enables data verification for IP/ICMP Echo, SNA Echo, UDP Echo, and Jitter operations. Checks each operation response for corruption. Use the **verify-data** command only when corruption may be an issue. Do not enable this feature during normal operation because it causes unnecessary overhead.
`Router(config-rtr)# data-pattern hex-pattern`	Allows you to specify an alphanumeric character string to verify that a udpEcho operation payload is not getting corrupted. The default data pattern used by the SAA is ABCD. This command allows you to specify your own hexadecimal patterns to more precisely monitor different source-to-destination and destination-to-source packets. This command works only for UDP Echo operations in this release.

Setting Statistics Gathering Characteristics

SAA operations capture statistics and collect error information. By default, the following information is captured and collected:

- Minimum and maximum response times
- Number of completions
- Sum of completion times
- Sum of the squares of completion times
- Accumulation of errors for noncompletions
- Total attempts (errors plus number of completions)
- Statistical distributions of response times

A statistical distribution of response times can be thought of as a set of buckets that holds the results of a probe. Each bucket holds the completion count that falls into that specific time interval. To modify the time intervals use the **statistics-distribution-interval** RTR configuration command. To modify the number of buckets, use the **distributions-of-statistics-kept** command. For example, if the **statistics-distribution-interval** command is set to 20 ms and the **distributions-of-statistics-kept** command is set to 3 (buckets a, b and c), and three RTT operations are performed with response times of 10 ms, 15 ms, and 30 ms, then the completion count for the buckets is 2 for a, 1 for b, and 0 for c.

In most situations, you need not change the statistical distribution interval or size. Only change the size when distributions are needed (for example, when performing statistical modeling of your network).

To control how much and which type of statistics are stored on the router, use the following commands in RTR configuration mode, as needed:

Command	Purpose
`Router(config-rtr)# distributions-of-statistics-kept` *size*	Sets the number of buckets or statistical distributions kept during the lifetime of the probe. Size is the number of buckets that contain data counts for their intervals. Applies to the following operations: IP/ICMP Echo, IP/ICMP Path Echo, UDP Echo, TcpConnect, DNS, DLSw, and SNA Echo.
`Router(config-rtr)# hops-of-statistics-kept` *size*	Collects pathEcho statistical distributions per hop per path. Size specifies the number of hops for which statistics are collected per path for each probe Applies to IP/ICMP Path Echo operations only.
`Router(config-rtr)# hours-of-statistics-kept` *hours*	Sets the number of hours for which statistics are maintained for the probe. Applies to the following operations: IP/ICMP Echo, IP/ICMP Path Echo, UDP Echo, TcpConnect, DNS, DLSw, and SNA Echo. For HTTP and Jitter operations, statistics are kept for the last 2 hours. This parameter cannot currently be reconfigured by a user.

Command	Purpose
`Router(config-rtr)# paths-of-statistics-kept` *size*	Collects statistical distributions for multiple paths. Size specifies the number of paths for which statistical distribution buckets are maintained per hour for each probe. Applies to IP/ICMP Path Echo operations only.
`Router(config-rtr)# statistics-distribution-interval` *milliseconds*	Sets the time interval for each statistical distribution. Applies to the following operations: IP/ICMP Echo, IP/ICMP Path Echo, UDP Echo, TcpConnect, DNS, DLSw, and SNA Echo.

Note When the **distribution-of-statistics-kept** command is set to the default (1), you need not set the **statistics-distribution-interval** command because it has no effect on the statistics kept. For more information, refer to the command documentation in the "Cisco Service Assurance Agent Commands" chapter of the *Cisco IOS Configuration Fundamentals Command Reference*.

Setting History Characteristics

The SAA can collect data samples for a given operation; these samples are called *history data*. By default, history data is not collected. When history collection is enabled, SAA collects the last *n* data points. The number of data points are configured using the **buckets-of-history-kept** RTR configuration command.

When collecting history, SAA also introduces the concept of *lives*. A life is defined as the operational lifetime of a probe. When a probe is stopped and restarted, data is kept in new life entries (if the number of entries is two or fewer). If the number of entries is more than two, the oldest entry is overwritten by the new entry.

History is not supported for HTTP and Jitter operations.

Note Collecting history increases the RAM usage. Collect history only when you believe there is a problem in the network. For general network response time information, use the statistics collected by the SAA. See the "Setting Statistics Gathering Characteristics" section for more information on statistics collection.

To control how much and which type of history is collected on the router, use the following commands in RTR configuration mode, as required:

Command	Purpose
`Router(config-rtr)# buckets-of-history-kept` *size*	For a pathEcho probe, sets the number of paths to store. For all other probes, sets the number (*size*) of data points to be kept. Applies to the following operations: IP/ICMP Echo, IP/ICMP Path Echo, UDP Echo, TcpConnect, DNS, DLSw, and SNA Echo.
`Router(config-rtr)# filter-for-history {none | all | overthreshold | failures}`	Defines the type of information kept in the history table for the probe. This is a required command to enable history. The **all**, **overthreshold**, or **failures** keywords must be specified for history to work. Applies to the following operations: IP/ICMP Echo, IP/ICMP Path Echo, UDP Echo, TcpConnect, DNS, DLSw, and SNA Echo.
`Router(config-rtr)# lives-of-history-kept` *lives*	Enables history collection and sets the number of lives maintained in the history table for the probe. Applies to the following operations: IP/ICMP Echo, IP/ICMP Path Echo, UDP Echo, TcpConnect, DNS, DLSw, and SNA Echo.
`Router(config-rtr)# samples-of-history-kept` *samples*	For a pathEcho probe, sets the number of hops in a path. For all other probes, RTR sets the number of samples to 1. Applies to the following operations: IP/ICMP Echo, IP/ICMP Path Echo, UDP Echo, TcpConnect, DNS, DLSw, and SNA Echo.

To disable history collection, use the default value (0) for the **lives-of-history-kept** command rather than the **filter-for-history none** command. The **lives-of-history-kept** command disables history collection before the operation of the probe is attempted, and the **filter-for-history** command with the **none** keyword checks for history inclusion after the operation of the probe attempt is made.

Setting Reaction Thresholds

You can configure the operation to send threshold notifications and use those notifications to trigger additional collection of time delay statistics. You can also configure the operation to send notifications when the probe loses connection, reestablishes connections, times out, and first succeeds after a timeout.

Thresholds can be a useful way to limit the amount of network notifications. For example, you could limit the sending of trap notifications to when a defined problem event occurs by setting a trap to be sent when a rising threshold is exceeded and another to be sent when the monitored connection recovers using the falling threshold specification.

To configure the rising threshold for an event, use the following command in RTR configuration mode when configuring the operation:

Command	Purpose
`Router(config-rtr)# threshold milliseconds`	Configures the rising threshold (hysteresis) that generates a reaction event and stores history information for the operation. This command applies to all operation types.

To configure the reaction conditions for an operation (including the falling threshold), use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# rtr reaction-configuration operation-number [connection-loss-enable] [timeout-enable] [threshold-falling milliseconds] [threshold-type option] [action-type option]`	Configures certain actions (for example, checking for connection losses or timeouts) to occur based on events controlled by the SAA.
Step 2	`Router(config)# rtr reaction-trigger operation-number target-number`	Defines an action type that will activate the operation.

Scheduling the Operation

After you have configured the operation, you must schedule the operation to begin capturing statistics and collecting error information. When scheduling an operation, it can start immediately or start at a certain month, day, and hour. Use the **pending** keyword when setting the operation to start at a later time. The **pending** keyword is also an internal state of the operation visible through SNMP. The pending state is also used when an operation is a reaction operation waiting to be triggered.

To schedule an SAA operation, use the following command in global configuration mode:

Command	Purpose
`Router(config)# rtr schedule operation-number [life {forever | seconds}] [start-time {hh:mm[:ss] [month day | day month] | pending | now | after hh:mm:ss}] [ageout seconds]`	Schedules the operation by configuring the time parameters.

Note After you schedule the operation with the **rtr schedule** command, you cannot change the configuration of the operation with the **rtr** global configuration command. To change the configuration of a scheduled operation, use the **no** form of the **rtr** command. The **no** form of the command removes all the configuration information of the operation, including the schedule, reaction configuration, and reaction triggers. You can now create a new configuration for the operation.

If the operation is in a pending state (the default), you can define the conditions under which the operation makes the transition from pending to active with the **rtr reaction-trigger** global configuration command. When the operation is in an active state it immediately begins collecting information.

Enabling the SAA Responder on Operational Targets

The SAA Responder is a component embedded in the target Cisco routing device that allows the system to anticipate and respond to SAA request packets. The responder can listen on any user-defined port for UDP and TCP protocol messages. In a client/server terminology, the SAA Responder is a Concurrent Multiservice Server.

A server listening on a user-specified port all the time on a router is prone to denial-of-service attacks. Therefore, the server should stop listening on that port after it services the request of the client. Because the ports on which to listen are unknown beforehand, there is a need for a mechanism through which the Responder can be notified on which port it should listen and respond. To meet this need, the SAA uses a Control Protocol to notify the Responder to listen on a particular port and protocol.

The Responder listens on a specific port for Control Protocol messages sent by an SAA. The control message carries information like the protocol, port number, duration, and so on. Upon receipt of the control message, the responder will enable the specified UDP or TCP port for the specified duration. During this time, the Responder accepts the requests and responds to them. The Responder disables the port once it responds to the SAA packet, or when the specified time expires. To further prevent security compromises, you can apply MD5 authentication for control messages.

The SAA Responder is needed only for the following nonnative services: TCP, UDP Echo, and Jitter (UDP +) operations. If services that are already provided by the target router (such as Telnet or HTTP) are chosen, the SAA Responder need not be enabled. For non-Cisco devices, the SA Responder cannot be configured and the SAA can send operational packets only to services native to those devices.

To enable SAA Responder functionality on a router, use the following command in global configuration mode:

Command	Purpose
`Router(config)# rtr responder [type {udpEcho | tcpConnect} [ipaddr ipaddr] {port port}]`	Enables SAA Responder functionality on a Cisco device. The optional **type**, **ipaddr**, and **port** keywords enable the SAA Responder to respond to probe packets without receiving control protocol packets. Note that if you use these keywords, however, generation of packet loss statistics will not be possible for the operation.

Configuring SAA Control Message Authentication

The SAA uses a control message protocol to communicate with the Cisco routers that are the target of SAA operations. For security reasons, users can enable authentication for the SAA Control Protocol. The authentication is provided using Message Digest 5 (MD5) authentication. This authentication requires key definition on the source and target SAA routers. Configure the key using the **keychain** global configuration command to enter key-chain configuration mode.

For details on how to configure key chains, refer to the "Managing Authentication Keys" section in the "Configuring IP Routing Protocol-Independent Features" chapter of the *Cisco IOS IP Configuration Guide*. See also the "SAA Control Protocol Authentication Example" section.

The **rtr key-chain** command notifies the SAA that it should use the previously configured key for authentication.

To configure the SAA RTR authentication, use the following command in global configuration mode:

Command	Purpose
`Router(config)# rtr key-chain name`	Specifies the key chain to be used for authentication of SAA operations.

Resetting the SAA

To perform an emergency reset of the SAA (including clearing all RTR configuration information), use the following command in global configuration mode:

Command	Purpose
`Router(config)# rtr reset`	Stops all operations and clears all SAA RTR configuration information.

Caution Use the **rtr reset** command only in extreme situations such as the incorrect configuration of a number of operations. The **rtr reset** command removes all SAA RTR configuration from the running configuration file.

In addition to stopping all operations and clearing the RTR configuration information, the **rtr reset** command returns the running configuration information to the startup condition. This command does not reread the configuration stored in NVRAM (the startup configuration file). To reconfigure the SAA, you must reenter the appropriate SAA configuration commands, or copy an existing configuration file containing your desired SAA configuration to the running configuration.

Restarting a Stopped Operation

To restart an operation, use the following command in RTR configuration mode:

Command	Purpose
`Router(config)# rtr restart operation-number`	Restarts an operation

Note that you can only restart operations in the active state; operations in the pending state cannot be restarted.

Displaying SAA Status and SAA Operational Results

To display information about the status and configuration of the SAA, use the following commands in EXEC mode, as needed. You can display information in a tabular or full format. Tabular format displays information in a column format that reduces the number of screens required to display the information. Full format displays all information using identifiers next to each displayed value.

Command	Purpose
Router> **show rtr application** [**tabular** | **full**]	Displays global information about the SAA feature.
Router> **show rtr authentication**	Displays authentication information.
Router> **show rtr collection-statistics** [*number*] [**tabular** | **full**]	Displays error totals collected for all operations or a specified operation.
Router> **show rtr configuration** [*number*] [**tabular** | **full**]	Displays configuration values including all defaults for all operations or a specified operation.
Router> **show rtr distributions-statistics** [*number*] [**tabular** | **full**]	Displays statistical distribution information (captured response times) for all operations or a specified operation.
Router> **show rtr history** [*number*] [**tabular** | **full**]	Displays history collected for all operations or a specified operation.
Router> **show rtr operational-state** [*number*] [**tabular** | **full**]	Displays the operational state of all operations or a specified operation.
Router> **show rtr reaction-trigger** [*number*] [**tabular** | **full**]	Displays the reaction trigger information for all operations or a specified operation.
Router> **show rtr responder**	Displays SAA Responder information.
Router> **show rtr totals-statistics** [*number*] [**tabular** | **full**]	Displays the total statistic values (accumulation of error counts and completions) for all operations or a specified operation.

For examples of the information displayed by these commands, and explanations of the output fields, see the "Cisco Service Assurance Agent Commands" chapter of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*.

Changing the Memory Threshold for the SAA

To specify how much memory must be available on the router to allow SAA configuration, use the following command in global configuration mode:

Command	Purpose
Router(config)# **rtr low-memory** *value*	Specifies the amount of memory (in bytes) that must be available to allow SAA configuration.

The **rtr low-memory** RTR configuration command allows you to specify the amount of memory that must be available to allow SAA configuration. The default value is 25 percent of the memory available on the system at startup. If the amount of available free memory falls below the value specified in the **rtr low-memory** command, then the SAA will not allow new operations to be configured. (The low-memory value is also referred to as the lowWaterMark.) If the value is set to 0, then SAA operations can be created until the system runs out of memory.

The value of the **rtr low-memory** command should not exceed the amount of free memory available on the system. To determine the amount of free memory available on the system, use the **show memory** EXEC command.

For example, if there is 6 MB of free memory when system starts up, and the default lowWaterMark is used, then SAA can use up to 4.5 MB memory for creating operations. If the free memory drops below 1.5 MB, then SAA cannot create any more operations.

Before every new operation is created, the SAA checks the lowWaterMark to ensure that it does not consume more memory than it is configured for.

The SAA also provides a MIB variable (rttMonApplProbeCapacity) to track how many operations (probes) can be created with the available memory. Each operation takes about 14 KB of memory (when default options are used). The equation to compute rttMonApplProbeCapacity is:

rttMonApplProbeCapacity =

MIN(((Free_Bytes_on_the_Router–rttMonApplFreeMemLowWaterMark) / Memory_required_by_each_probe), (rttMonApplNumCtrlAdminEntry–Num_of_Probes_already_configured))

For example, when the system boots, the rttMonApplProbeCapacity variable might show that 200 operations can be configured. But if other subsystems in the router start using up more memory when network gets busy, the SAA may only be able to configure 150 probes.

The **show rtr application** EXEC command will display the number of operations that can be run on the device in the "System Max Number of Entries" field.

Configuring Specific Operations

The following sections provide more information on configuring the operations:

- Configuring a Jitter Operation
- Configuring a DHCP Operation

Configuring a Jitter Operation

The primary function of Jitter (UDP+) operations is to measure response times for real-time traffic, such as VoIP.

When multiple packets are sent consecutively from source to destination, for example, 10 ms apart, and if the network is behaving ideally, the destination should be receiving them 10 ms apart. But if there are delays in the network (like queueing, arriving through alternate routes, and so on) the arrival delay between packets might be greater than or less than 10 ms. Using this example, a positive jitter value indicates that the packets arrived greater than 10 ms apart. If the packets arrive 12 ms apart, then positive jitter is 2 ms; if the packets arrive 8 ms apart, then negative jitter is 2 ms. For delay-sensitive networks like VoIP, positive jitter values are undesirable, and a jitter value of 0 is ideal.

In theory the delay incurred by packets traversing a route from host A to host B is equal in each direction. However, the delay in one direction may be much higher. There might also be asymmetric sending and receiving paths between host A and host B. The one-way delay statistics provided by the UDP+ operation addresses this need for per-direction information, and allows you to more readily identify where bottlenecks occur.

The Jitter operation was designed to measure the delay variance and packet loss in IP networks by generating synthetic UDP traffic. The Jitter operation sends *n* packets, each of size *s*, from source router to a target router (which should have SAA Responder enabled on it) each *t* ms apart. Each parameter is user-configurable. By default, ten packet-frames are generated every 10 ms with an RTP payload size of 10 bytes (Cisco gateways combine two such frames and send them every 20 ms) to simulate voice traffic.

The packets SAA sends out to measure Jitter carry packet sending sequence and receiving sequence information, and sending and receiving time stamps from source and responder. Based on these, Jitter operations are capable of measuring the following:

- Per-direction delay variance between packets (jitter)
- One-way delay
- Per-direction packet-loss
- Average round trip time

Jitter

Based on time stamps from consecutive packets, the sender can calculate the jitter value, which is the difference in the latency (interpacket delay). Note that for the interpacket delay the clocks on the two devices need not be synchronized.

One-Way Delay

The Jitter probe packet contains four time stamps for its journey from host A to host B and back again: time sent from A, time received at B, time sent from B, and time received at A. The one-way delay is computed using the following simple formulae:

one-way delay from A to B = time received at B–time sent from A

one-way delay from B to A = time received at A–time sent from B

Therefore, to accurately measure per-direction delay between two devices, you must synchronize the clocks on each device. To synchronize the clocks on each device, you must configure the Cisco IOS Network Time Protocol (NTP) feature on both the source and destination devices.

If the sum of the source to destination (SD) and the destination to source (DS) values is not within 10 percent of the round-trip time, the one-way measurement value is considered faulty and is discarded. This type of inaccuracy usually is found in devices that are not time-synchronized.

To configure a Jitter operation, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# rtr operation-number`	Specifies an operation and enters RTR configuration mode.
Step 2	`Router(config-rtr)# type jitter dest-ipaddr {name | ipaddr} dest-port port-number [source-ipaddr {name | ipaddr}] [source-port port-number] [control {enable | disable}] [num-packets number-packets] [interval inter-packet-interval]`	Defines a Jitter operation.
Step 3	`Router(config-rtr)# frequency seconds`	(Optional) Sets how often the operation should send a probe out to gather statistics.
Step 4	`Router(config-rtr)# owner text`	(Optional) Configures the SNMP owner of the operation.
Step 5	`Router(config-rtr)# tag text`	(Optional) Logically links operations in a group.
Step 6	`Router(config-rtr)# timeout milliseconds`	(Optional) Sets the amount of time the probe waits for a response from its request packet.

	Command	Purpose
Step 7	`Router(config-rtr)# tos number`	(Optional) Defines the IP ToS byte for request packets. (This command applies to the IP/ICMP Echo, UDP Echo, and Jitter operation types only.)
Step 8	`Router(config)# rtr schedule operation-number [life seconds] [start-time {pending | now | hh:mm [month day | day month]}] [ageout seconds]`	Schedules the operation by configuring the time parameters.

You must also enable the SAA Responder on the destination device. To enable the Responder, configure the following global configuration command on the destination device:

Command	Purpose
`Router(config)# rtr responder`	Enables SAA Responder functionality on a device.

Alternatively, you can enable the SAA Responder using SNMP with the **rttMonApplResponder.0 -Integer 1** command from a network management application.

To monitor the operational state of the Jitter operation, use the **show rtr operational-state** EXEC command. To view the statistics gathered by Jitter operation, use the **show rtr collection-statistics** EXEC command.

Disabling SAA RTR Control Protocol for Jitter Operations

In some cases, you may wish to disable SAA RTR control protocol traffic while still allowing a Jitter operation. You can avoid control message traffic by configuring a specific port to be permanently open for the Responder.

Note If you disable the SAA RTR Control Procotol, only round-trip-time statistics will be gathered for the Jitter operation.

To disable RTR control protocol traffic, use the **control disable** option when configuring the Jitter operation on the origin device:

Command	Purpose
`Router(config-rtr)# type jitter dest-ipaddr ipaddr dest-port port-number control disable`	Configures the operation as a Jitter operation. • Use of the **control disable** keywords disable RTR control protocol messages.

Then enable the SAA Responder on the destination device. With the control protocol disabled, you must specify the operation type, as shown here:

Command	Purpose
`Router(config)# rtr responder type udpEcho ipaddr ipaddr port port-number`	Enables SAA Responder functionality on a device. • Note that the **udpEcho** keyword corresponds to the Jitter operation type.

The following example shows the configuration for the sending device and the target device. The same port number is specified on each.

RouterA (Sending Device) Configuration

```
RouterA(config)# rtr 10
RouterA(config-rtr)# type jitter dest-ipaddr 172.24.132.100 dest-port 99 control disable
```

RouterB (Target Device) Configuration

```
RouterB(config)# rtr responder type udpEcho ipaddr 172.24.132.100 port 99
```

Note that in this configuration, most of the standard Jitter statistics can not be collected.

Configuring a DHCP Operation

To test how long is taken for the system to obtain an IP address from any DHCP server on the network, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# rtr` *operation-number*	Specifies an operation and enters RTR configuration mode.
Step 2	`Router(config-rtr)# type dhcp [source-ipaddr` *source-ipaddr*`] [dest-ipaddr` *dest-ipaddr*`] [option 82 [circuit-id` *circuit-id*`] [remote-id` *remote-id*`] [subnet-mask` *subnet-mask*`]]`	Defines a DHCP operation. DHCP option 82 allows you to specify the circuit ID, remote ID, or the subnet mask for the destination DHCP server.
Step 3	`Router(config-rtr)# frequency` *seconds*	(Optional) Sets how often the operation should be performed (for example, **frequency 60** indicates once every 60 seconds).
Step 4	`Router(config-rtr)# owner` *text*	(Optional) Configures the SNMP owner of the operation.
Step 5	`Router(config-rtr)# tag` *text*	(Optional) Logically links operations in a group.
Step 6	`Router(config-rtr)# timeout` *milliseconds*	(Optional) Sets the amount of time the probe waits for a response from its request packet.
Step 7	`Router(config)# rtr schedule` *operation-number* `[life` *seconds*`] start-time {pending | now | hh:mm [month day | day month]} [ageout` *seconds*`]`	Schedules the operation by configuring the time parameters.

To test how long is taken for the system to obtain an IP address from a specific DHCP server, use the following global configuration command in addition to the commands above:

Command	Purpose
`Router(config)# ip dhcp-server` *ipaddress*	Specifies a single DHCP server to be used for DHCP requests from the system.

Configuring SAA Operations Using SNMP

SAA operations using SNMP can be created in two ways: through a createAndGo operation and through a createAndWait operation. Each operation has a specific set of variables that need to be defined before it can be operational. The following variables should be specified in the order shown:

- Set rttMonCtrlAdminStatus
- Set rttMonCtrlAdminRttType
- Set rttMonEchoAdminProtocol
- Set the rest of the configuration variables

Additionally, each operation requires specific variables to be set before it can be activated. The following is a comprehensive list of variables based on operation type:

- For Echo, pathEcho and DLSW operations:
  - rttMonEchoAdminTargetAddress
- For udpEcho, tcpConnect and Jitter operations:
  - rttMonEchoAdminTargetAddress
  - rttMonEchoAdminTargetPort
- For HTTP operations:
  - rttMonEchoAdminURL
- For DNS operations:
  - rttMonEchoAdminTargetAddressString
  - rttMonEchoAdminNameServer

Note DHCP operations do not require any additional variables.

For examples of configuring the SAA using SNMP, see the"SAA Configuration Using SNMP Examples" section.

Modifying Configuration Parameters While an Operation Is Active

Operation configuration parameters (except for trigger admin variables) cannot be changed while the probe is running because the data storage gets affected by configuration changes. For example, if an operation is configured to poll every 60 seconds, then a statsCapture table will store 60 samples in an hour. But if the polling interval is changed while the operation is still running, the statsCapture data aggregation will become corrupted.

To change the configuration of an active operation using SNMP, perform the following steps:

Step 1 Set rttMonCtrlOperState to immediateStop (3).

Step 2 Set rttMonCtrlAdminStatus to notInService (2).

Step 3 Set required variables.

Step 4 Set rttMonCtrlAdminStatus to Active (1).

Accessing SAA Data Using SNMP

The results of SAA operations are stored in different tables of the CISCO-RTTMON-MIB.

Historical information is saved in the rttMonHistoryCollectionTable. Currently HTTP and Jitter operations do not support the rttMonHistoryCollectionTable.

The rttMonAppl table contains the general SAA information, such as number of operations supported on the device and the maximum capacity of the device.

For Echo, PathEcho, UDP Echo, TCP Connect, DLSw, DNS, FTP and DHCP operations:

- Latest sample (updated after each frequency cycle) is stored in the rttMonCtrlOperTable.
- Statistical distribution information (aggregate of each sample, updated after each frequency cycle) is stored in the rttMonStatsCaptureTable and rttMonStatsTotalsTable.
- Error information (aggregate of each sample, updated after each frequency cycle) is stored in the rttMonStatsCollectTable.

For HTTP operations:

- Latest sample is stored in the rttMonLatestHTTPOperTable and the rttMonCtrlOperTable (only the rttMonLatestRttOperCompletionTime variable is updated).
- Statistical information is stored in the rttMonHTTPStatsTable and the rttMonStatsTotalsTable.

For Jitter operations:

- Latest sample is stored in the rttMonLatestJitterOperTable and in the rttMonCtrlOperTable (only the rttMonLatestRttOperCompletionTime variable is updated).
- Statistical information is stored in the rttMonJitterStatsTable and the rttMonStatsTotalsTable.

The following operations are not supported in the CISCO-RTTMON-MIB:

- HTTP first byte operations
- Frame Relay operations
- Path Jitter operations
- ATM interface service level monitoringoperations
- Frame Relay service level monitoring operations
- Interface service level monitoring operations (Serial/hssi/dds/imaE1/imaT1)
- Controller service level monitoring operations (Ds3/E3/T1/E1/ft1)

Enabling SAA SNMP Notifications

To enable SNMP notifications (traps or informs) to be sent from your system, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# snmp-server enable traps rtr`	Enables the sending of SAA RTR notifications.
Step 2	`Router(config)# snmp-server host` *host* `[traps | informs] [version {1 | 2c | 3 [auth | noauth | priv]}]` *community-string* `[rtr]`	Specifies whether you want the SNMP notifications sent as traps or informs, the version of SNMP to use, the security level of the notifications (for SNMPv3), and the recipient (host) of the notifications.

The RTR notifications are defined in the CISCO-RTTMON-MIB (enterprise 1.3.6.1.4.1.9.9.42.2) and are as follows:

- 1 rttMonConnectionChangeNotification
- 2 rttMonTimeoutNotification
- 3 rttMonThresholdNotification
- 4 rttMonVerifyErrorNotification

For further SNMP configuration steps, see the "Configuring SNMP Support" chapter in this document.

SAA Configuration Using the CLI Examples

This section provides the following configuration examples for creating operations on a Cisco device using the Cisco IOS CLI:

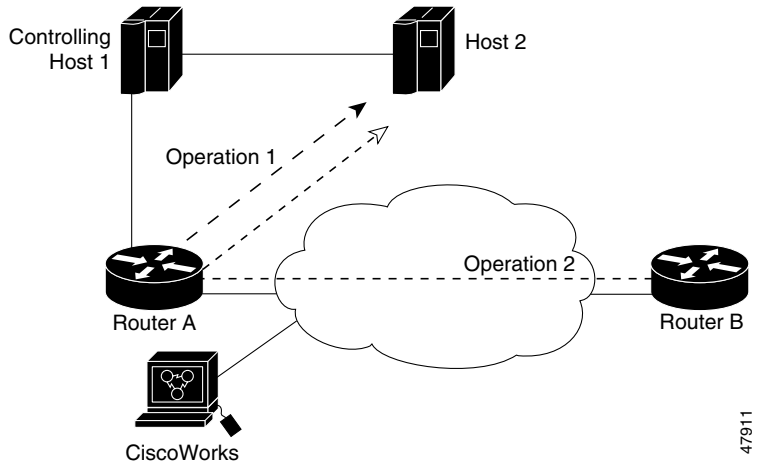
- SNA Echo Example
- IP/ICMP Path Echo Example
- TcpConnect Example
- SAA Control Protocol Authentication Example
- Jitter Operation Example
- HTTP GET Operation Example
- HTTP RAW Operation Using RAW Submode Example
- HTTP RAW Operation Through a Proxy Server Example
- FTP Operation Example
- DNS Operation Example
- DLSw Operation Example
- DHCP Operation Example
- Connection Loss Trigger Example

SNA Echo Example

The example in Figure 19 shows probe 1 configured from Router A to Host 2, and probe 2 configured from Router B to Host 2. This configuration allows normative analysis of the network to determine a baseline from which triggers (and general reactions) are configured. Also, two SNA physical units (PUs)

must be configured: CWBC0A and CWBC0B. For information on configuring SNA PUs, see the **dspu host** or the **sna host** command in the *Cisco IOS Bridging and IBM Networking Command Reference, Volume 2*.

Figure 19 SNA LU2 Echo Operation



Router A Configuration

```
RouterA(config)# rtr 1
RouterA(config-rtr)# type echo protocol snaLU2EchoAppl CWBC0A
RouterA(config-rtr)# exit
RouterA(config)# rtr schedule 1 start-time now
RouterA(config)#
```

Router B Configuration

```
RouterB(config)# rtr 2
RouterB(config-rtr)# type echo protocol snaLU2EchoAppl CWBC0B
RouterB(config-rtr)# exit
RouterB(config)# rtr schedule 1 start-time now
RouterB(config)#
```

Configuration Files for Router A and Router B

After you save the configurations for Router A and Router B (using the **copy running-config startup-config** EXEC command), the following information is stored in the configuration files:

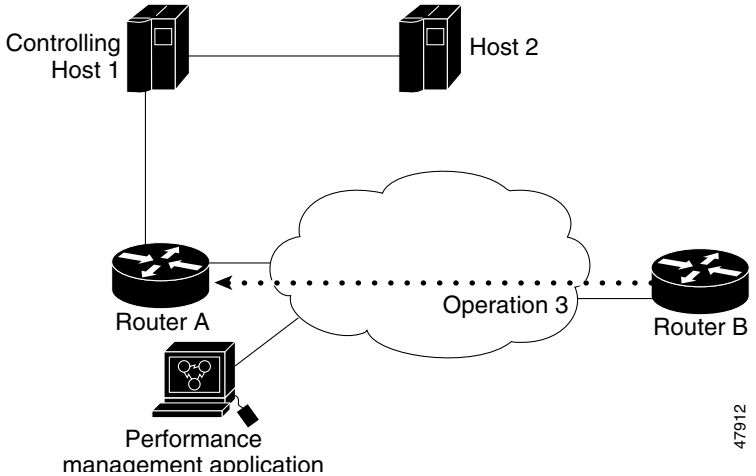
```
!Router A Configuration File
! Router A's PU Configuration
sna host CWBC0A xid-snd 05dcc00a rmac 4001.3745.1088 rsap 4 lsap 12 focalpoint
rtr 1
 type echo protocol snaLU2EchoAppl CWBC0A
 paths-of-statistics-kept 1
 hops-of-statistics-kept 1
 samples-of-history-kept 1
rtr schedule 1 start-time now

!Router B Configuration File
!Router B's PU Configuration from the Configuration File:
sna host CWBC0B xid-snd 05dcc00b rmac 4001.3745.1088 rsap 4 lsap 12 focalpoint
rtr 2
 type echo protocol snaLU2EchoAppl CWBC0B
```

```
 paths-of-statistics-kept 1
 hops-of-statistics-kept 1
 samples-of-history-kept 1
rtr schedule 2 start-time now
```

IP/ICMP Path Echo Example

The example in Figure 20 shows that Operation 3 is configured on Router B to record statistics for each hop along the path that the operation takes to reach it destination (Router A).

Figure 20 IP/ICMP Path Echo Operation



This example sets up a pathEcho (with history) pending entry from Router B to Router A via IP/ICMP. It attempts to execute three times in 25 seconds (first attempt starts at 0 seconds) and keeps statistics for those three attempts in three history buckets. The entry can be started five times before wrapping over stored history (**lives-of-history-kept** = 5).

Router B Configuration

```
RouterB(config)# rtr 3
RouterB(config-rtr)# type pathEcho protocol ipIcmpEcho RouterA
RouterB(config-rtr)# frequency 10
RouterB(config-rtr)# lives-of-history-kept 5
RouterB(config-rtr)# buckets-of-history-kept 3
RouterB(config-rtr)# filter-for-history all
RouterB(config-rtr)# exit
RouterB(config)# rtr schedule 3 life 25
RouterB(config)# exit
```

Router B Configuration

After you save the configuration (using the **copy running-config startup-config** EXEC command), the information is stored in the configuration file. Some necessary default forms of commands are automatically included if they are not specified in the configuration setting, based on their necessity for operation execution. In this example, the default **response-data-size** command is added to the configuration file automatically.

```
RouterB# show startup-config
```

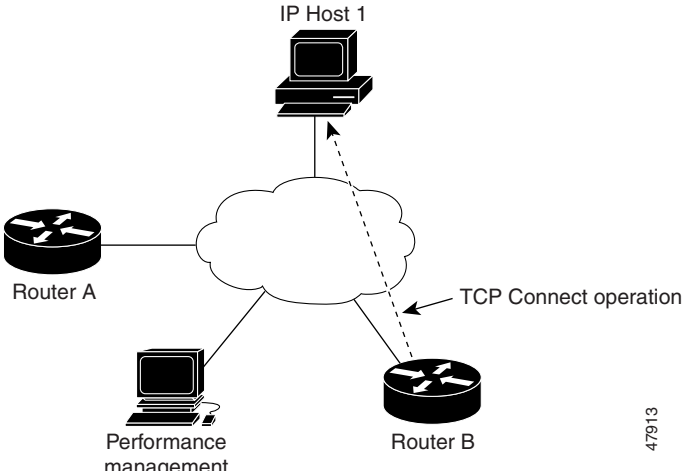
```
.
.
.
!
rtr 3
 type pathEcho protocol ipIcmpEcho 172.28.161.21
 frequency 10
 response-data-size 1
 lives-of-history-kept 5
 buckets-of-history-kept 3
 filter-for-history all
rtr schedule 3 life 25 start-time pending
!
.
.
.
```

Note that the SAA Responder need not be enabled on Router A for this operation.

TcpConnect Example

The example in Figure 21 shows a tcpConnect operation configured from Router B to the Telnet port (TCP port 23) of IP Host 1 (IP address 10.0.0.1).

Figure 21 TcpConnect Operation



Router B Configuration

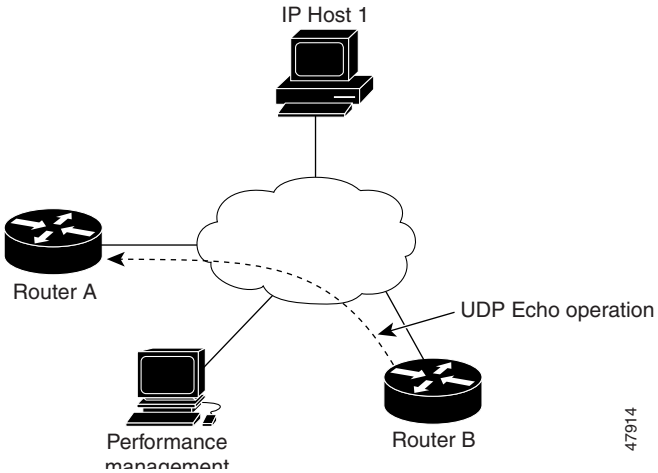
```
RouterB(config)# rtr 5
RouterB(config-rtr)# type tcpConn dest-ipaddr 10.0.0.1 dest-port 23 control disable
RouterB(config-rtr)# exit
RouterB(config)# rtr schedule 5 start now
```

In the example, the Control Protocol for the probe is disabled. RTR Collector uses the RTR control protocol to notify the RTR Responder on the responder router to enable the target port temporarily. This action allows the Responder to respond to the probe packet. In this case, because the target is not a router and a well-known TCP port is used, there is no need to send the control message.

SAA Control Protocol Authentication Example

The example in Figure 22 shows a udpEcho probe configured from Router B to UDP port 888 on Router A (IP address 20.0.0.1).

Figure 22 udpEcho Operation



Note Configuring the SAA Control Protocol authentication is optional. However, if you configure authentication for Router B, you must configure the same authentication for Router A.

In the following configuration example, a key chain named csaa-key is configured on both routers. The **rtr key-chain** global configuration command enables RTR MD5 authentication on the control messages.

Router A Configuration

```
RouterA(config)# key chain csaa-key
RouterA(config-keychain)# key 1
RouterA(config-keychain-key)# key-string secret
RouterA(config-keychain-key)# exit
RouterA(config-keychain)# exit
RouterA(config)# rtr key-chain csaa-key
RouterA(config)# rtr responder
```

Router B Configuration

```
RouterB(config)# key chain csaa-key
RouterB(config-keychain)# key 1
RouterB(config-keychain-key)# key-string secret
RouterB(config-keychain-key)# exit
RouterB(config-keychain)# exit
RouterB(config)# rtr key-chain csaa-key
RouterB(config)# rtr 6
RouterB(config-rtr)# type udpEcho dest-ipaddr 20.0.0.1 dest-port 888 control enable
RouterB(config-rtr)# exit
RouterB(config)# rtr schedule 6 start now
```

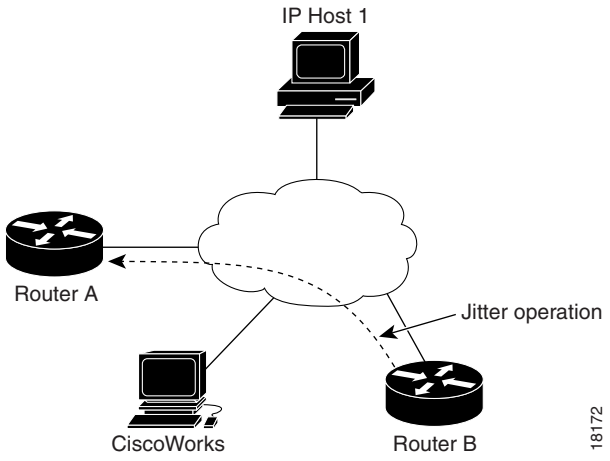
Jitter Operation Example

In order to perform a Jitter operation (also known as a UDP+ operation), the SAA Responder must be enabled on the target router using the **rtr responder** global configuration command.

A Jitter operation consists of a train of packets sent at a constant interval. The numbers of packets sent and the interval are user-configurable. When the SAA Responder receives the packets, it time-stamps the reception time and then sends the packet back.

In the example shown in Figure 23, SAA operation number 200 is created and configured as a Jitter (UDP+) operation using the destination IP address 172.24.132.100, destination UDP port number 99. The operation will send 20 packets at 20-ms intervals.

Figure 23 Jitter Operation



Router A Configuration

```
RouterA(config)# rtr responder
```

Router B Configuration

```
RouterB(config)#rtr 200
RouterB(config-rtr)#type jitter dest-ip 172.24.132.100 dest-port 99 num-packets 20
interval 20
```

After the Jitter operation has run, you can display the results with the **show rtr collection-statistics** EXEC command. The following example shows sample output:

```
Entry Number: 200
Target Address: 172.24.132.100, Port Number: 31337
Start Time: *14:14:14.000 EST Thu Apr 6 2000
RTT Values:
NumOfRTT: 2800  RTTSum: 4792    RTTSum2: 8830
Packet Loss Values:
PacketLossSD: 0 PacketLossDS: 0
PacketOutOfSequence: 0  PacketMIA: 0    PacketLateArrival: 0
InternalError: 0        Busies: 0
Jitter Values:
MinOfPositivesSD: 1     MaxOfPositivesSD: 1
NumOfPositivesSD: 249   SumOfPositivesSD: 249   Sum2PositivesSD: 249
MinOfNegativesSD: 1     MaxOfNegativesSD: 2
NumOfNegativesSD: 238   SumOfNegativesSD: 239   Sum2NegativesSD: 241
MinOfPositivesDS: 1     MaxOfPositivesDS: 1
```

```
NumOfPositivesDS: 97    SumOfPositivesDS: 97    Sum2PositivesDS: 97
MinOfNegativesDS: 1     MaxOfNegativesDS: 1
NumOfNegativesDS: 92    SumOfNegativesDS: 92    Sum2NegativesDS: 92
```

The values shown indicate the aggregated values for the current hour. RTT stands for Round-Trip-Time. Table 25 describes the significant fields shown in the display.

Table 25 show rtr collection-statistics Field Descriptions

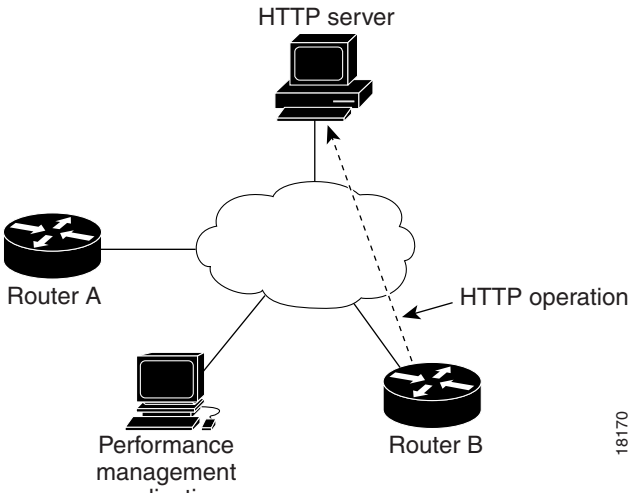
Field	Description
NumOfRTT	The number of successful round trips.
RTTSum	The sum of those round-trip values (in ms).
RTTSum2	The sum of squares of those round-trip values (in ms).
PacketLossSD	The number of packets lost from source to destination.
PacketLossDS	The number of packets lost from destination to source.
PacketOutOfSequence	The number of packets returned out of order.
PacketMIA	The number of packets lost where the direction (SD/DS) cannot be determined.
PacketLateArrival	The number of packets that arrived after the timeout.
InternalError	The number of times an operation could not be started due to other internal failures.
Busies	The number of times this operation could not be started because the previously scheduled run was not finished.
MinOfPositivesSD MaxOfPositivesSD	The minimum and maximum positive jitter values from source to destination (in ms). Positive jitter values indicate delays in receiving time from one packet to another.
NumOfPositivesSD	The number of jitter values from source to destination that are positive (that is, network latency increases for two consecutive test packets).
SumOfPositivesSD	The sum of those positive values (in ms).
Sum2PositivesSD	The sum of the squares of the positive values.
MinOfNegativesSD MaxOfNegativesSD	The minimum and maximum negative jitter values from source to destination. The absolute value is given.
NumOfNegativesSD	The number of jitter values from source to destination that are negative (that is, network latency decreases for two consecutive test packets).
SumOfNegativesSD	The sum of the negative values.
Sum2NegativesSD	The sum of the squares of the negative values.

The DS values show the same information as above for DS Jitter values.

HTTP GET Operation Example

In the example shown in Figure 24, operation 5 is created and configured as an HTTP GET operation. The destination URL is http://www.cisco.com:

Figure 24 HTTP Operation



Router B Configuration

```
RouterB(config)#rtr 5
RouterB(config-rtr)#type http operation get url http://www.cisco.com
RouterB(config-rtr)# exit
RouterB(config)# rtr schedule 5 start-time now
```

HTTP RAW Operation Using RAW Submode Example

In the following example, SAA operation 6 is created and configured as an HTTP RAW operation. To use the raw commands, enter HTTP-RAW submode through use of the **http-raw-request** RTR configuration command. The RTR HTTP-RAW configuration submode is indicated by the `(config-rtr-http)` router prompt.

```
(config)# rtr 6
(config-rtr)# type http operation raw url http://www.cisco.com
(config-rtr)# http-raw-request
(config-rtr-http)# GET /index.html HTTP/1.0\r\n
(config-rtr-http)# \r\n
(config-rtr-http)# exit
(config)# rtr schedule 6 start-time now
```

HTTP RAW Operation Through a Proxy Server Example

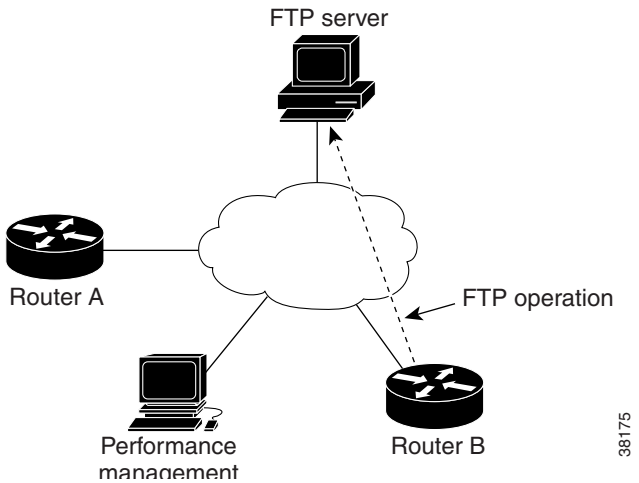
In this example, http://www.proxy.cisco.com is the proxy server and http://www.yahoo.com is the HTTP server:

```
(config)# rtr 6
(config-rtr)# type http operation raw url http://www.proxy.cisco.com
(config-rtr)# http-raw-request
(config-rtr-http)# GET http://www.yahoo.com HTTP/1.0\r\n
(config-rtr-http)# \r\n
(config-rtr-http)# exit
(config)# rtr schedule 6 start-time now
```

FTP Operation Example

An FTP operation is configured as shown in Figure 25.

Figure 25 FTP Operation



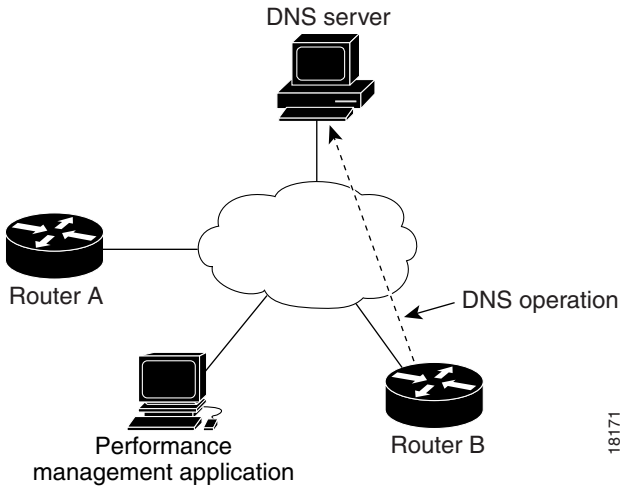
In this example, SAA operation 20 is configured as an FTP operation; ira is the user, smith is the password, zxq is the host name or address, and test is the file name.

```
RouterB(config)# rtr 20
RouterB(config-rtr)# type ftp operation get url ftp://ira:smith@zxq/test
RouterB(config-rtr)# exit
RouterB(config)# rtr schedule 20 start-time now
```

DNS Operation Example

In the example following Figure 26, SAA operation 7 is created and configured as a DNS operation using the name server IP address 172.20.2.132:

Figure 26 DNS Operation




```
RouterB(config)#rtr 7
RouterB(config-rtr)#type dns target-addr lethe name-server 172.20.2.132
RouterB(config-rtr)# exit
RouterB(config)# rtr schedule 7 start-time now
```

DLSw Operation Example

In the example following Figure 27, DLSw peers 172.20.26.10 and 172.21.27.11 are configured:

Figure 27 DLSw Operation

Router A Configuration File

```
RouterA# show running-config
.
.
.
dlsw local-peer peer-id 172.20.26.10
dlsw remote-peer 0 tcp 172.21.27.11
.
.
.
rtr 1
 type dlsw peer-ipaddr 172.21.27.11
 rtr schedule 1 start-time now
.
.
.
```

Router B Configuration File

```
RouterB# show running-config
.
.
.
dlsw local-peer peer-ip 172.21.27.11
dlsw remote-peer 0 tcp 172.20.26.10
.
.
.
```

DHCP Operation Example

In the following example, SAA operation number 4 is configured as a DHCP operation enabled for DHCP server 172.16.20.3. Note that DHCP option 82 is used to specify the circuit ID.

```
Router(config)# rtr 4
Router(config-rtr)# type dhcp option 82 circuit-id 10005A6F1234
Router(config-rtr)# exit
Router(config)# ip dhcp-server 172.16.20.3
```

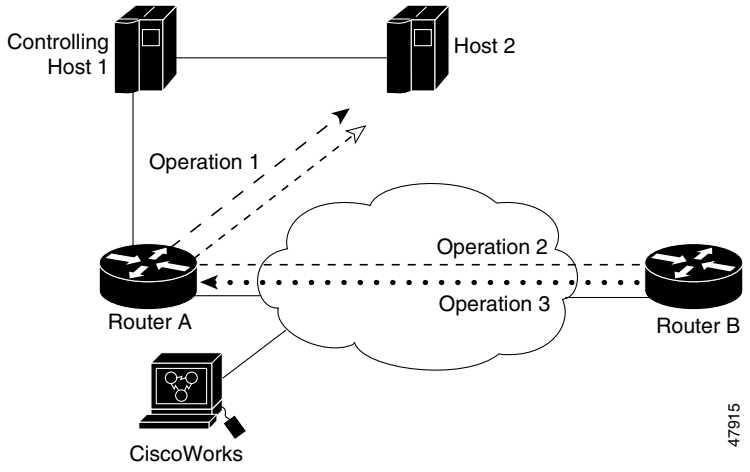
Option 82 is is called the Relay Agent Information option and is inserted by the DHCP relay agent when forwarding client-originated DHCP packets to a DHCP server. Servers recognizing the Relay Agent Information option may use the information to implement IP address or other parameter assignment policies. The DHCP server echoes the option back verbatim to the relay agent in server-to-client replies, and the relay agent strips the option before forwarding the reply to the client.

The Relay Agent Information option is organized as a single DHCP option that contains one or more suboptions that convey information known by the relay agent. The initial suboptions are defined for a relay agent that is co-located in a public circuit access unit. These include a **circuit-id** for the incoming circuit, a **remote-id** that provides a trusted identifier for the remote high-speed modem, and the **subnet-mask** of the logical IP subnet from which the relay agent received the client DHCP packet.

Connection Loss Trigger Example

Figure 28 shows SAA operations 1, 2, and 3 in the network. This example shows how to configure a trigger if operation 2 encounters a connection loss from Router B to Host 2. If a connection loss occurs between Router B and Host 2, a trap is issued, an SNA network management vector transport (NMVT) alert is issued, and the operation 3 state is changed to active.

Figure 28 Configuring a Trigger for Connection Loss



Router B Configuration

```
RouterB(config)# rtr reaction-configuration 2 connection-loss-enable action-type
trapNmvtAndTrigger
RouterB(config)# rtr reaction-trigger 2 3
```

Note The operation numbers must be unique within only one router. The examples shown use three different probe operation numbers for clarity.

SAA Configuration Using SNMP Examples

This section shows examples of configuring SAA operations using SNMP. In these examples, objects in the CISCO-RTTMON-MIB are set. Note that these are not Cisco IOS software commands, and that these types of configurations must be done from an NMS using the appropriate SNMP software.

Creating an Echo Operation Example

In the following example, an LSR path is specified to compute the response time for a specific path. The source address on the source router is specified and the operation is scheduled to run forever.

```
rttMonCtrlAdminStatus.<index> -Integer 4  \
rttMonCtrlAdminRttType.<index> -Integer 1 \
rttMonEchoAdminProtocol.<index> -Integer 2 \
rttMonEchoAdminTargetAddress.<index> -OctetString "04 00 00 01" \
rttMonEchoAdminSourceAddress.<index> -OctetString "01 00 00 01" \
rttMonEchoPathAdminHopAddress.<index>.1 -OctetString "02 00 00 01" \
rttMonEchoPathAdminHopAddress.<index>.2 -OctetString "03 00 00 01" \
rttMonScheduleAdminRttStartTime.<index> -TimeTicks 1 \
rttMonScheduleAdminRttLife.<index> -Integer 2147483647
```

Creating a Path Echo Operation Example

```
rttMonCtrlAdminStatus.<index> -Integer 4  \
```

```
rttMonCtrlAdminRttType.<index> -Integer 2 \
rttMonEchoAdminProtocol.<index> -Integer 2 \
rttMonEchoAdminTargetAddress.<index> -OctetString "05 00 00 02" \
rttMonScheduleAdminRttStartTime.<index> -TimeTicks 1 \
rttMonScheduleAdminConceptRowAgeout.<index> -Integer 0
```

Creating a UDP Operation Example

In this example the UDP operation computes response time to communicate with the SAA Responder running on a Cisco router using port number 4444. The ToS bits are set to use QoS metrics on the network.

```
rttMonCtrlAdminStatus.<index> -Integer 4  \
rttMonCtrlAdminRttType.<index> -Integer 5 \
rttMonEchoAdminProtocol.<index> -Integer 3 \
rttMonEchoAdminTargetAddress.<index> -OctetString "05 00 00 02" \
rttMonEchoAdminTargetPort.<index> -Integer 4444 \
rttMonEchoAdminTOS.<index> -Integer 5 \
rttMonScheduleAdminRttStartTime.<index> -TimeTicks 1
```

Creating a TCP Operation Example

In this example the TCP operation computes response time to communicate with a host using the well-known HTTP server port number. Note that the Control Protocol is disabled, meaning that we will go to a well-known port instead of a Responder.

```
rttMonCtrlAdminStatus.<index> -Integer 4  \
rttMonCtrlAdminRttType.<index> -Integer 6 \
rttMonEchoAdminProtocol.<index> -Integer 24 \
rttMonEchoAdminTargetAddress.<index> -OctetString "05 00 00 02" \
rttMonEchoAdminTargetPort.<index> -Integer 80 \
rttMonEchoAdminControlEnable.<index> -Integer 2 \
rttMonScheduleAdminRttStartTime.<index> -TimeTicks 1
```

Creating a Jitter Operation Example

```
rttMonCtrlAdminStatus.<index> -Integer 4  \
rttMonCtrlAdminRttType.<index> -Integer 9 \
rttMonEchoAdminProtocol.<index> -Integer 27 \
rttMonEchoAdminTargetAddress.<index> -OctetString "05 00 00 02" \
rttMonEchoAdminTargetPort.<index> -Integer 8000 \
rttMonEchoAdminInterval.<index> -Integer 20 \
rttMonEchoAdminNumPackets.<index> -Integer 100 \
rttMonScheduleAdminRttStartTime.<index> -TimeTicks 1
```

Creating an HTTP Get Operation Example

```
rttMonCtrlAdminStatus.<index> -Integer 4  \
rttMonCtrlAdminRttType.<index> -Integer 7 \
rttMonEchoAdminProtocol.<index> -Integer 25 \
rttMonEchoAdminOperation.<index> -Integer 1 \
rttMonEchoAdminURL.<index> -DisplayString "http://www.cisco.com:80/index.html" \
rttMonEchoAdminHTTPVersion.<index> -DisplayString "1.0" \
rttMonEchoAdminCache.<index> -Integer 2 \
rttMonScheduleAdminRttStartTime.<index> -TimeTicks 1
```

Creating an HTTP RAW Operation Example

```
rttMonCtrlAdminStatus.<index> -Integer 4  \
rttMonCtrlAdminRttType.<index> -Integer 7 \
rttMonEchoAdminProtocol.<index> -Integer 25 \
rttMonEchoAdminOperation.<index> -Integer 2 \
rttMonEchoAdminURL.<index> -DisplayString "http://www.cisco.com" \
rttMonEchoAdminString1.<index> -DisplayString "GET /index.html HTTP/1.0\r\n\r\n" \
rttMonEchoAdminNameServer.<index> -OctetString "01 05 07 09" \
rttMonScheduleAdminRttStartTime.<index> -TimeTicks 1
```

Creating a DNS Operation Example

```
rttMonCtrlAdminStatus.<index> -Integer 4  \
rttMonCtrlAdminRttType.<index> -Integer 8 \
rttMonEchoAdminProtocol.<index> -Integer 26 \
rttMonEchoAdminTargetAddressString.<index> -DisplayString "www.cisco.com" \
rttMonEchoAdminNameServer.<index> -OctetString "11 05 07 09" \
rttMonScheduleAdminRttStartTime.<index> -TimeTicks 1
```

Creating a DLSw Operation Example

```
rttMonCtrlAdminStatus.<index> -Integer 4  \
rttMonCtrlAdminRttType.<index> -Integer 10 \
rttMonEchoAdminProtocol.<index> -Integer 28 \
rttMonEchoAdminTargetAddress.<index> -OctetString "05 00 00 02" \
rttMonScheduleAdminRttStartTime.<index> -TimeTicks 1
```

Creating a DHCP Operation Example

```
rttMonCtrlAdminStatus.<index> -Integer 4  \
rttMonCtrlAdminRttType.<index> -Integer 11 \
rttMonEchoAdminProtocol.<index> -Integer 29
rttMonScheduleAdminRttStartTime.<index> -TimeTicks 1
```

Creating an FTP Operation Example

```
rttMonCtrlAdminStatus.<index> -Integer 4  \
rttMonCtrlAdminRttType.<index> -Integer 12 \
rttMonEchoAdminProtocol.<index> -Integer 30 \
rttMonEchoAdminOperation.<index> -Integer 3 \
rttMonEchoAdminURL.<index> -DisplayString
"ftp://anonymous@test:www.cisco.com/temp/temp.txt" \
rttMonScheduleAdminRttStartTime.<index> -TimeTicks 1
```

SAA Command List

The following is a list of commands described in this chapter, used for the configuration and management of SAA. This list is provided to assist you in locating commands in the Cisco IOS Command Reference documents.

- **atm-slm statistics**
- **buckets-of-history-kept**
- **clear saa apm cache**
- **data-pattern**
- **distributions-of-statistics-kept**
  filter-for-history
- **frequency**
- **hops-of-statistics-kept**
- **http-raw-request**
- **hours-of-statistics-kept**
- **lives-of-history-kept**
- **lsr-path**
- **owner**
- **paths-of-statistics-kept**
- **request-data-size**
- **response-data-size**
- **rtr**
- **rtr key-chain**
- **rtr low-memory**
- **rtr reaction-configuration**
- **rtr reaction-trigger**
- **rtr reset**
- **rtr responder**
- **rtr responder type tcpConnect**
- **rtr responder type udpEcho**
- **rtr responder type frame-relay**
- **rtr restart**
- **rtr schedule**
- **saa apm cache-size**
- **saa apm copy**
- **saa apm lowWaterMark**
- **saa apm operation**
- **samples-of-history-kept**
- **show rtr application**

- **show rtr authentication**
- **show rtr collection-statistics**
- **show rtr configuration**
- **show rtr distributions-statistics**
- **show rtr history**
- **show rtr operational-state**
- **show rtr reaction-trigger**
- **show rtr responder**
- **show rtr totals-statistics**
- **show saa apm cache**
- **show saa apm information**
- **show saa apm operation**
- **show saa apm results**
- **statistics-distribution-interval**
- **tag**
- **threshold**
- **timeout**
- **tos**
- **type atm-slm**
- **type dhcp**
- **type dlsw**
- **type dns**
- **type echo**
- **type frame-relay**
- **type ftp**
- **type http**
- **type jitter**
- **type pathEcho**
- **type pathJitter**
- **type slm**
- **type t1-slm**
- **type tcpConnect**
- **type udpEcho**
- **verify-data**
- **vrfName**

Configuring Web Cache Services Using WCCP

The Web Cache Communication Protocol (WCCP) is a Cisco-developed content-routing technology that allows you to integrate cache engines (such as the Cisco Cache Engine 550) into your network infrastructure. Cisco IOS Release 12.1 and later releases allow the use of either Version 1 (WCCPv1) or Version 2 (WCCPv2) of the WCCP. This chapter describes how to configure your router to redirect traffic to cache engines (web caches), describes how to manage cache engine clusters (cache farms), and outlines the benefits of using WCCPv2.

For a complete description of the WCCP configuration commands in this chapter, refer to the "WCCP Commands" chapter in the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

The tasks in this chapter assume that you have already configured cache engines on your network. For specific information on hardware and network planning associated with Cisco Cache Engines and WCCP, see the Product Literature and Documentation links available on the Cisco.com Web Scaling site at http://www.cisco.com/warp/public/cc/pd/cxsr/ces/index.shtml.

Note Cisco Systems replaced the Cache Engine 500 Series platforms with Content Engine Platforms in July 2001. Cache Engine Products were the Cache Engine 505, 550, 570, and 550-DS3. Content Engine Products are the Content Engine 507, 560, 590, and 7320.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Understanding WCCP

The Cisco IOS WCCP feature allows utilization of Cisco Cache Engines (or other caches running WCCP) to localize web traffic patterns in the network, enabling content requests to be fulfilled locally. Traffic localization reduces transmission costs and download time.

WCCP enables Cisco IOS routing platforms to transparently redirect content requests. The main benefit of transparent redirection is that users need not configure their browsers to use a web proxy. Instead, they can use the target URL to request content, and have their requests automatically redirected to a cache engine. The word "transparent" is this case means that the end user does not know that a requested file (such as a web page) came from the cache engine instead of from the originally specified server.

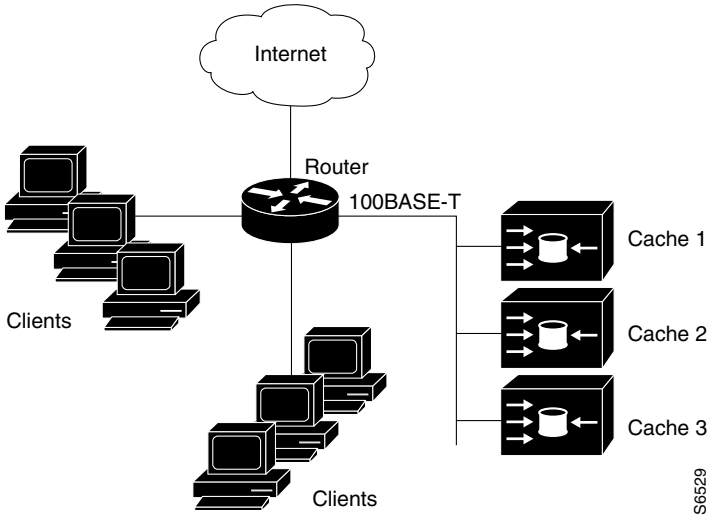
When a cache engine receives a request, it attempts to service it from its own local cache. If the requested information is not present, the cache engine issues its own request to the originally targeted server to get the required information. When the cache engine retrieves the requested information, it forwards it to the requesting client and caches it to fulfill future requests, thus maximizing download performance and substantially reducing transmission costs.

WCCP enables a series of cache engines, called a *cache engine cluster*, to provide content to a router or multiple routers. Network administrators can easily scale their cache engines to handle heavy traffic loads through these clustering capabilities. Cisco clustering technology enables each cache member to work in parallel, resulting in linear scalability. Clustering cache engines greatly improves the scalability, redundancy, and availability of your caching solution. You can cluster up to 32 cache engines to scale to your desired capacity.

Understanding WCCPv1 Configuration

With WCCP-Version 1, only a single router services a cluster. In this scenario, this router is the device that performs all the IP packet redirection. Figure 29 illustrates how this configuration appears.

Figure 29 Cisco Cache Engine Network Configuration Using WCCP-Version 1



Content is not duplicated on the cache engines. The benefit of using multiple caches is that you can scale a caching solution by clustering multiple physical caches to appear as one logical cache.

The following sequence of events details how WCCPv1 configuration works:

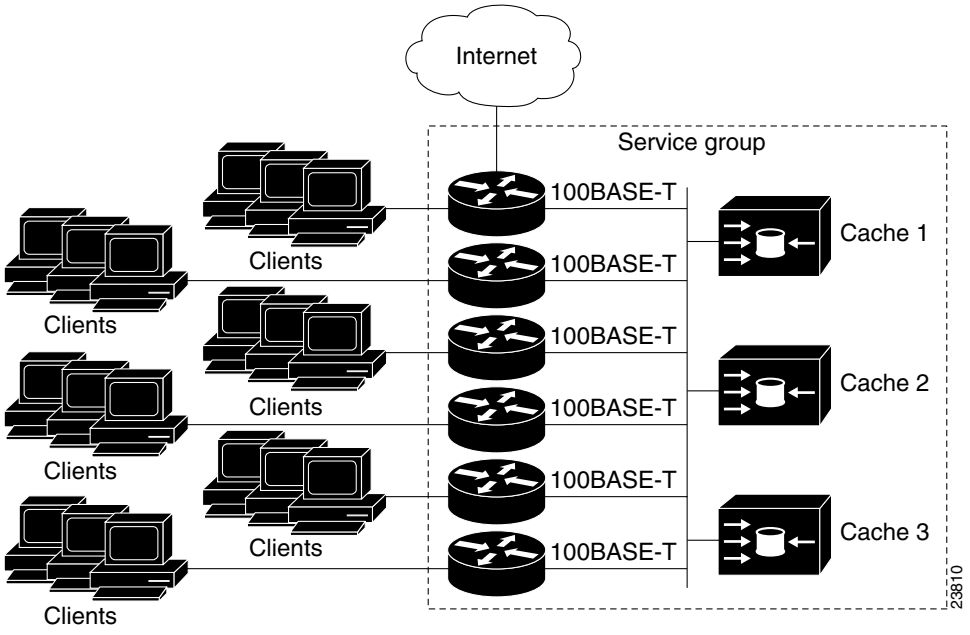
1. Each cache engine is configured by the system administrator with the IP address of the control router. Up to 32 cache engines can connect to a single control router.
2. The cache engines send their IP addresses to the control router using WCCP, indicating their presence. Routers and cache engines communicate to each other via a control channel; this channel is based on UDP port 2048.
3. This information is used by the control router to create a cluster view (a list of caches in the cluster). This view is sent to each cache in the cluster, essentially making all the cache engines aware of each other. A stable view is established after the membership of the cluster remains the same for a certain amount of time.

4. Once a stable view has been established, one cache engine is elected as the lead cache engine. (The lead is defined as the cache engine seen by all the cache engines in the cluster with the lowest IP address). This lead cache engine uses WCCP to indicate to the control router how IP packet redirection should be performed. Specifically, the lead cache engine designates how redirected traffic should be distributed across the cache engines in the cluster.

Understanding WCCPv2 Configuration

Multiple routers can use WCCPv2 to service a cache cluster. This is in contrast to WCCPv1, in which only one router could redirect content requests to a cluster. Figure 30 illustrates a sample configuration using multiple routers.

Figure 30 Cisco Cache Engine Network Configuration Using WCCP v2



The subset of cache engines within a cluster and routers connected to the cluster that are running the same service is known as a *service group*. Available services include TCP and User Datagram Protocol (UDP) redirection.

Using WCCPv1, the cache engines were configured with the address of the single router. WCCPv2 requires that each cache engine be aware of all the routers in the service group. To specify the addresses of all the routers in a service group, you must choose one of the following methods:

- Unicast—A list of router addresses for each of the routers in the group is configured on each cache engine. In this case the address of each router in the group must be explicitly specified for each cache engine during configuration.
- Multicast—A single multicast address is configured on each cache engine. In the multicast address method, the cache engine sends a single-address notification that provides coverage for all routers in the service group. For example, a cache engine could indicate that packets should be sent to a

multicast address of 224.0.0.100, which would send a multicast packet to all routers in the service group configured for group listening using WCCP (see the **ip wccp group-listen** interface configuration command for details).

The multicast option is easier to configure because you need only specify a single address on each cache engine. This option also allows you to add and remove routers from a service group dynamically, without needing to reconfigure the cache engines with a different list of addresses each time.

The following sequence of events details how WCCPv2 configuration works:

1. Each cache engine is configured with a list of routers.
2. Each cache engine announces its presence and a list of all routers with which it has established communications. The routers reply with their view (list) of cache engines in the group.
3. Once the view is consistent across all cache engines in the cluster, one cache engine is designated as the lead and sets the policy that the routers need to deploy in redirecting packets.

The following sections describe how to configure WCCPv2 on routers so they may participate in a service group.

WCCPv2 Features

WCCPv2 provides the features described in the following sections:

- Support for Services Other than HTTP
- Support for Multiple Routers
- MD5 Security
- Web Cache Packet Return
- Load Distribution

Support for Services Other than HTTP

WCCPv2 allows redirection of traffic other than HTTP (TCP port 80 traffic), including a variety of UDP and TCP traffic. WCCPv1 supported the redirection of HTTP (TCP port 80) traffic only. WCCPv2 supports the redirection of packets intended for other ports, including those used for proxy-web cache handling, File Transfer Protocol (FTP) caching, FTP proxy handling, web caching for ports other than 80, and Real Audio, video, and telephony applications.

To accommodate the various types of services available, WCCPv2 introduces the concept of multiple *service groups*. Service information is specified in the WCCP configuration commands using dynamic services identification numbers (such as “98”) or a predefined service keywords (such as “web-cache”). This information is used to validate that service group members are all using or providing the same service.

The cache engines in service group specify traffic to be redirected by protocol (TCP or UDP) and port (source or destination). Each service group has a priority status assigned to it. Packets are matched against service groups in priority order.

Support for Multiple Routers

WCCPv2 allows multiple routers to be attached to a cluster of cache engines. The use of multiple routers in a service group allows for redundancy, interface aggregation, and distribution of the redirection load.

MD5 Security

WCCPv2 provides optional authentication that enables you to control which routers and cache engines become part of the service group using passwords and the HMAC MD5 standard. Shared-secret MD5 one-time authentication (set using the **ip wccp** [**password** [**0-7**] *password*] global configuration command) enables messages to be protected against interception, inspection, and replay.

Web Cache Packet Return

If a cache engine is unable to provide a requested object it has cached due to error or overload, the cache engine will return the request to the router for onward transmission to the originally specified destination server. WCCPv2 provides a check on packets that determines which requests have been returned from the cache engine unserviced. Using this information, the router can then forward the request to the originally targeted server (rather than attempting to resend the request to the cache cluster). This provides error handling transparency to clients.

Typical reasons why a cache engine would reject packets and initiate the packet return feature include the following:

- Instances when the cache engine is overloaded and has no room to service the packets
- Instances when the cache engine is filtering for certain conditions that make caching packets counterproductive (for example, when IP authentication has been turned on)

Load Distribution

WCCPv2 can be used to adjust the load being offered to individual cache engines to provide an effective use of the available resources while helping to ensure high quality of service (QoS) to the clients. WCCPv2 allows the designated cache to adjust the load on a particular cache and balance the load across the caches in a cluster. WCCPv2 uses three techniques to perform load distribution:

- Hot Spot Handling—Allows an individual hash bucket to be distributed across all the cache engines. Prior to WCCPv2, information from one hash bucket could only go to one cache engine.
- Load Balancing—Allows the set of hash buckets assigned to a cache engine to be adjusted so that the load can be shifted from an overwhelmed cache engine to other members that have available capacity.
- Load Shedding—Enables the router to selectively redirect the load to avoid exceeding the capacity of a cache engine.

The use of these hashing parameters prevents one cache from being overloaded and reduces the potential for bottlenecking.

Restrictions for WCCPv2

The following limitations apply to WCCP v2:

- WCCP works only with IP networks.
- For routers servicing a multicast cluster, the Time To Live (TTL) value must be set at 15 or fewer.
- Because the messages may now be IP multicast, members may receive messages that will not be relevant or are duplicates. Appropriate filtering needs to be performed.
- Service groups can comprise up to 32 cache engines and 32 routers.
- All cache engines in a cluster must be configured to communicate with all routers servicing the cluster.
- Multicast addresses must be from 224.0.0.0 to 239.255.255.255.

Configuring WCCP

The following configuration tasks assume that you have already installed and configured the cache engines you want to include in your network. You must configure the cache engines in the cluster before configuring WCCP functionality on your routers. Refer to the *Cisco Cache Engine User Guide* for cache engine configuration and setup tasks.

IP must be configured on the router interface connected to the cache engines and on the router interface connected to the Internet. Note that Cisco Cache Engines require use of a Fast Ethernet interface for a direct connection. Examples of router configuration tasks follow this section. For complete descriptions of the command syntax, refer to the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*.

Perform the tasks found in the following sections to configure WCCP on a router:

- Specifying a Version of WCCP (Optional)
- Configuring a Service Group Using WCCPv2 (Required)
- Excluding Traffic on a Specific Interface from Redirection (Optional)
- Registering a Router to a Multicast Address (Optional)
- Using Access Lists for a WCCP Service Group (Optional)
- Setting a Password for a Router and Cache Engines (Optional)

Specifying a Version of WCCP

Until you configure a WCCP service using the **ip wccp** {**web-cache** | *service-number*} global configuration command, WCCP is disabled on the router. The first use of a form of the **ip wccp** command enables WCCP. By default WCCPv2 is used for services, but you can use WCCPv1 functionality instead. To change the running version of WCCP from Version 2 to Version 1, or to return to WCCPv2 after an initial change, use the following command in EXEC mode:

Command	Purpose
`Router# ip wccp version {1 | 2}`	Specifies which version of WCCP to configure on a router. WCCPv2 is the default running version.

WCCPv1 does not use the WCCP commands from earlier Cisco IOS versions. Instead, use the WCCP commands documented in this chapter. If a function is not allowed in WCCPv1, an error prompt will be printed to the screen. For example, if WCCPv1 is running on the router and you try to configure a dynamic service, the following message will be displayed: "WCCP V1 only supports the web-cache service." The **show ip wccp** EXEC command will display the WCCP protocol version number that is currently running on your router.

Configuring a Service Group Using WCCPv2

WCCPv2 uses service groups based on logical redirection services, deployed for intercepting and redirecting traffic. The standard service is web cache, which intercepts TCP port 80 (HTTP) traffic and redirects that traffic to the cache engines. This service is referred to as a *well-known service*, because the characteristics of the web cache service are known by both the router and cache engines. A description of a well-known service is not required beyond a service identification (in this case, the Command Line Interface (CLI) provides a **web-cache** keyword in the command syntax).

In addition to the web cache service, there can be up to seven dynamic services running concurrently in a service group.

Note More than one service can run on a router at the same time, and routers and cache devices can be part of multiple service groups at the same time.

The dynamic services are defined by the cache engines; the cache instructs the router which protocol or ports to intercept, and how to distribute the traffic. The router itself does not have information on the characteristics of the dynamic service group's traffic, because this information is provided by the first web cache to join the group. In a dynamic service, up to eight ports can be specified within a single protocol.

Cisco Cache Engines, for example, use dynamic service 99 to specify a reverse-proxy service. However, other cache devices may use this service number for some other service. The following configuration information deals with enabling general services on Cisco routers. Refer to the cache server documentation for information on configuring services on cache devices.

To enable a service on a router, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip wccp {web-cache | service-number} [group-address groupaddress] [redirect-list access-list] [group-list access-list] [password password]`	Specifies a web-cache or dynamic service to enable on the router, specifies the IP multicast address used by the service group, specifies any access lists to use, specifies whether to use MD5 authentication, and enables the WCCP service.
Step 2	`Router(config)# interface type number`	Specifies an interface to configure and enters interface configuration mode.
Step 3	`Router(config-if)# ip wccp {web-cache | service-number} redirect {out | in}`	Enables WCCP redirection on the specified interface.

As indicated by the **out** and **in** keyword options in the **ip wccp** *service* **redirect** command, redirection can be specified for outbound interfaces or inbound interfaces. Inbound traffic can be configured to use Cisco Express Forwarding (CEF), Fast Forwarding, or Process Forwarding.

Configuring WCCP for redirection for inbound traffic on interfaces allows you to avoid the overhead associated with CEF forwarding for outbound traffic. Setting an output feature on any interface results in the slower switching path of the feature being taken by all packets arriving at all interfaces. Setting an input feature on an interface results in only those packets arriving at that interface taking the configured feature path; packets arriving at other interfaces will use the faster default path. Configuring WCCP for inbound traffic also allows packets to be classified before the routing table lookup, which translates into faster redirection of packets.

Specifying a Web Cache Service

Using the specific forms of the above commands, you can configure a web-cache service as follows:

	Command	Purpose
Step 1	`Router(config)# ip wccp web-cache`	Enables the web cache service on the router.
Step 2	`Router(config)# interface` *type number*	Targets an interface number for which the web cache service will run, and enters interface configuration mode.
Step 3	`Router(config-if)# ip wccp web-cache redirect {out | in}`	Enables the check on packets to determine if they qualify to be redirected to a web cache, using the interface specified in Step 2.

Excluding Traffic on a Specific Interface from Redirection

To exclude any interface from redirecting inbound traffic, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface` *type number*	Specifies an interface to configure, and enters interface configuration mode.
Step 2	`Router(config-if)# ip wccp redirect exclude in`	Allows inbound packets on this interface to be excluded from redirection.

Registering a Router to a Multicast Address

If you decide to use the multicast address option for your service group, you must configure the router to listen for the multicast broadcasts on an interface using the following commands:

	Command	Purpose
Step 1	`Router(config)# ip wccp {web-cache | service-number} group-address` *groupaddress*	Specifies the multicast address for the service group.

	Command	Purpose
Step 2	`Router(config)# interface type number`	Specifies the interface to be configured for multicast reception.
Step 3	`Router(config-if)# ip wccp {web-cache | service-number} group-listen`	Enables the reception of IP multicast packets (content originating from the cache engines) on the interface specified in Step 2.

For network configurations where redirected traffic needs to traverse an intervening router, the router being traversed must be configured to perform IP multicast routing. You must configure the following two components to enable traversal over an intervening router:

- Enable IP multicast routing using the **ip multicast-routing** global configuration mode command.
- Enable the interfaces to which the cache engines will connect to receive multicast transmissions using the **ip wccp group-listen** interface configuration mode command (note that earlier Cisco IOS versions required the use of the **ip pim** interface configuration command).

Using Access Lists for a WCCP Service Group

To configure the router to use an access list to determine which traffic should be directed to which cache engines, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# access-list access-list permit ip host host-address [destination-address | destination-host | any]`	Creates an access list that enables or disables traffic redirection to the cache engine.
Step 2	`Router(config)# ip wccp web-cache group-list access-list`	Indicates to the router from which IP addresses of cache engines to accept packets.

To disable caching for certain clients, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# access-list access-list permit ip host host-address [destination-address | destination-host | any]`	Creates an access list that enables or disables traffic redirection to the cache engine.
Step 2	`Router(config)# ip wccp web-cache redirect-list access-list`	Sets the access list used to enable redirection.

Setting a Password for a Router and Cache Engines

MD5 password security requires that each router and cache engine that wants to join a service group be configured with the service group password. The password can consist of up to seven characters. Each cache engine or router in the service group will authenticate the security component in a received WCCP packet immediately after validating the WCCP message header. Packets failing authentication will be discarded.

To configure an MD5 password for use by the router in WCCP communications, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip wccp web-cache password password`	Sets an MD5 password on the router.

Verifying and Monitoring WCCP Configuration Settings

Use the following commands in EXEC mode, as needed to verify and monitor the configuration settings for WCCP:

Command	Purpose
`Router# show ip wccp [web-cache | service-number]`	Displays global information related to WCCP, including the protocol version currently running, the number of cache engines in the routers service group, which cache engine group is allowed to connect to the router, and which access list is being used.
`Router# show ip wccp {web-cache | service-number} detail`	Queries the router for information on which cache engines of a specific service group the router has detected. The information can be displayed for either the web cache service or the specified dynamic service.
`Router# show ip interface`	Displays status about whether any ip wccp redirection commands are configured on an interface. For example, "Web Cache Redirect is enabled / disabled."
`Router# show ip wccp {web-cache | service-number} view`	Displays which devices in a particular service group have been detected and which cache engines are having trouble becoming visible to all other routers to which the current router is connected. The **view** keyword indicates a list of addresses of the service group. The information can be displayed for either the web cache service or the specified dynamic service. For further troubleshooting information, use the **show ip wccp** {**web-cache** | *service number*} **service** command.

WCCP Configuration Examples

This section provides the following configuration examples:

- Changing the Version of WCCP on a Router Example
- Performing a General WCCPv2 Configuration Example
- Running a Web Cache Service Example
- Running a Reverse Proxy Service Example
- Registering a Router to a Multicast Address Example
- Using Access Lists Example
- Setting a Password for a Router and Cache Engines Example
- Verifying WCCP Settings Example

Changing the Version of WCCP on a Router Example

The following example shows the process of changing the WCCP version from the default of WCCPv2 to WCCPv1, and enabling the web-cache service in WCCPv1:

```
Router# show ip wccp
% WCCP version 2 is not enabled
Router# configure terminal
Router(config)# ip wccp version 1
Router(config)# end
Router# show ip wccp
% WCCP version 1 is not enabled

Router# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip wccp web-cache
Router(config)# end
Router# show ip wccp
Global WCCP information:
    Router information:
        Router Identifier:                  10.4.9.8
        Protocol Version:                   1.0
. . .
```

Performing a General WCCPv2 Configuration Example

The following example shows a general WCCPv2 configuration session:

```
Router# configure terminal
Router(config)# ip wccp web-cache group-address 224.1.1.100 password alaska1
Router(config)# interface ethernet0
Router(config-if)# ip wccp web-cache redirect out
```

Running a Web Cache Service Example

The following example shows a web cache service configuration session:

```
router# configure terminal
router(config)# ip wccp web-cache
router(config)# interface ethernet 0
router(config-if)# ip wccp web-cache redirect out
Router(config-if)# ^Z
Router# copy running-config startup-config
```

The following example shows a configuration session in which redirection of HTTP traffic arriving on interface 0/1 is enabled:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface ethernet 0/1
Router(config-if)# ip wccp web-cache redirect in
Router(config-if)# ^Z
Router# show ip interface ethernet 0/1
.
.
.
WCCP Redirect inbound is enabled
WCCP Redirect exclude is disabled
```

.
.
.

Running a Reverse Proxy Service Example

The following example assumes you a configuring a service group using Cisco Cache Engines, which use dynamic service 99 to run a reverse proxy service:

```
router# configure terminal
router(config)# ip wccp 99
router(config)# interface ethernet 0
router(config-if)# ip wccp 99 redirect out
```

Registering a Router to a Multicast Address Example

The following example shows how to register a router to a multicast address of 224.1.1.100:

```
Router(config)# ip wccp web-cache group-address 224.1.1.100
Router(config)# interface ethernet 0
Router(config-if)# ip wccp web cache group-listen
```

The following example shows a router configured to run a reverse proxy service, using the multicast address of 224.1.1.1. Redirection applies to packets outgoing via interface ethernet 0:

```
Router(config)# ip wccp 99 group-address 224.1.1.1
Router(config)# interface ethernet 0
Router(config-if)# ip wccp 99 redirect out
```

Using Access Lists Example

To achieve better security, you can use a standard access list to notify the router which IP addresses are valid addresses for a cache engine attempting to register with the current router. The following example shows a standard access list configuration session where the access list number is 10 for some sample hosts:

```
router(config)# access-list 10 permit host 11.1.1.1
router(config)# access-list 10 permit host 11.1.1.2
router(config)# access-list 10 permit host 11.1.1.3
router(config)# ip wccp web-cache group-list 10
```

To disable caching for certain clients, servers, or client/server pairs, you can use WCCP access lists. The following example shows that any requests coming from 10.1.1.1 to 12.1.1.1 will bypass the cache, and that all other requests will be serviced normally:

```
Router(config)# ip wccp web-cache redirect-list 120
Router(config)# access-list 120 deny tcp host 10.1.1.1 any
Router(config)# access-list 120 deny tcp any host 12.1.1.1
Router(config)# access-list 120 permit ip any any
```

The following example configures a router to redirect web-related packets received via interface ethernet 0/1, destined to any host except 209.165.196.51:

```
Router(config)# access-list 100 deny ip any host 209.165.196.51
Router(config)# access-list 100 permit ip any any
```

```
Router(config)# ip wccp web-cache redirect-list 100
Router(config)# interface Ethernet 0/1
Router(config-if)# ip wccp web-cache redirect in
```

Setting a Password for a Router and Cache Engines Example

The following example shows a WCCPv2 password configuration session where the password is alaska1:

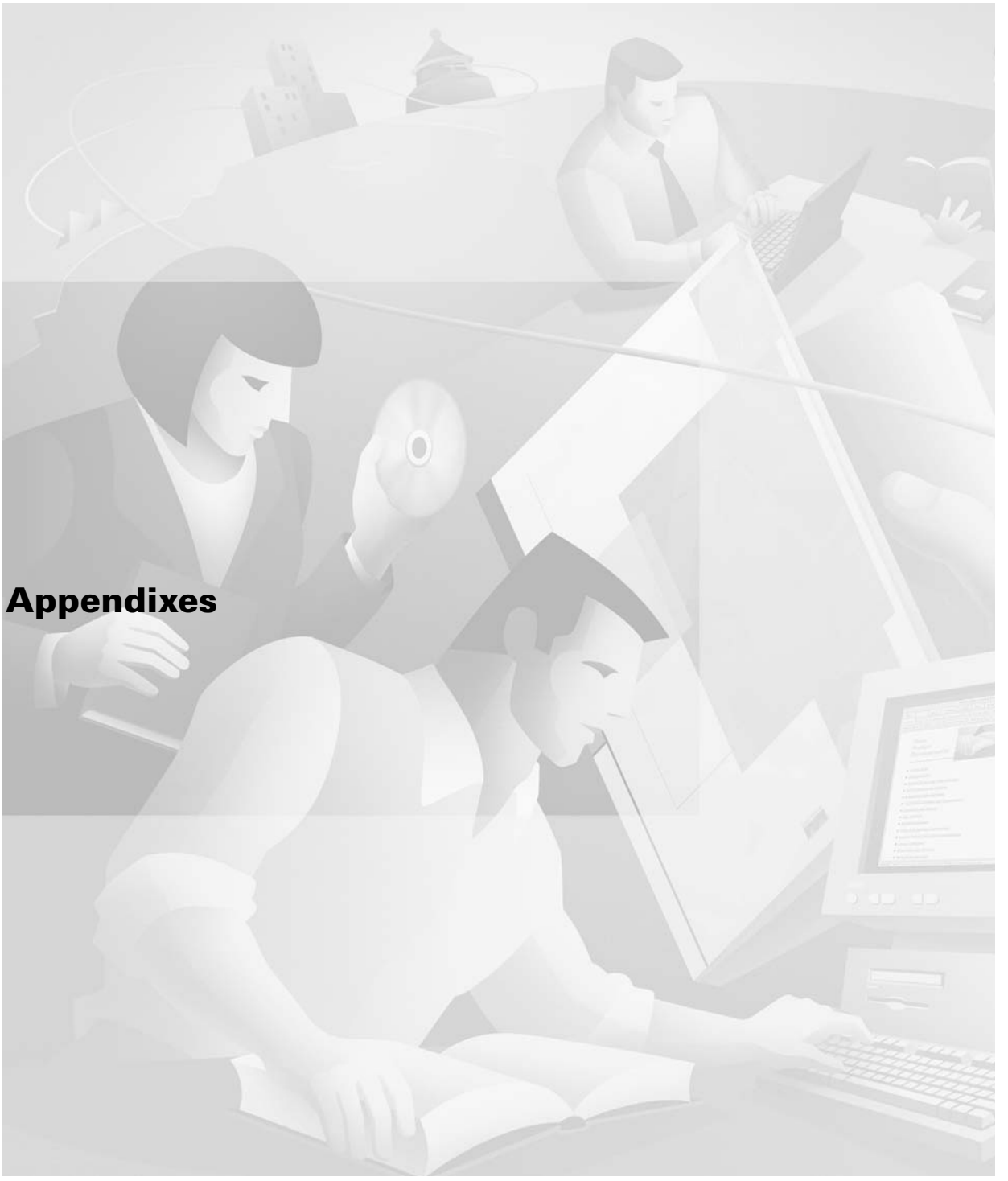
```
router# configure terminal
router(config)# ip wccp web-cache password alaska1
```

Verifying WCCP Settings Example

To verify your configuration changes, use the **more system:running-config** EXEC command. The following example shows that the both the web cache service and dynamic service 99 are enabled on the router:

```
router# more system:running-config

Building configuration...
Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname router4
!
enable secret 5 $1$nSVy$faliJsVQXVPW.KuCxZNTh1
enable password alabama1
!
ip subnet-zero
ip wccp web-cache
ip wccp 99
ip domain-name cisco.com
ip name-server 10.1.1.1
ip name-server 10.1.1.2
ip name-server 10.1.1.3
!
!
!
interface Ethernet0
ip address 10.3.1.2 255.255.255.0
no ip directed-broadcast
ip wccp web-cache redirect out
ip wccp 99 redirect out
no ip route-cache
no ip mroute-cache
!

interface Ethernet1
ip address 10.4.1.1 255.255.255.0
no ip directed-broadcast
```

```
ip wccp 99 redirect out
no ip route-cache
no ip mroute-cache
!
interface Serial0
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
!
interface Serial1
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
!
ip default-gateway 10.3.1.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.3.1.1
no ip http server
!
!
!
line con 0
transport input none
line aux 0
transport input all
line vty 0 4
password alaska1
login
!
end
```



Appendixes

Cisco IOS Command Modes

This appendix contains summaries of the command and configuration modes used in the Cisco IOS Command-Line Interface (CLI) in Cisco IOS Release 12.2. The availability of configuration modes will depend on the feature set found in your system image and on which router platform you are using. For specific information on any particular configuration mode, see the documentation references given in the mode summaries.

This appendix lists command modes in the following categories:

- Base Command Modes
- Configuration Modes and Submodes

These lists include short summaries of the modes.

Following the configuration mode summary list, Table 26 presents the configuration mode summaries organized by router prompt, and includes examples of entering each mode.

Base Command Modes

Base command modes are used for navigating the CLI and performing basic router startup, configuration, and monitoring tasks. For more information on the base command modes, see the "Using the Command-Line Interface" chapter of this document. For details about setup mode, see the "Using Configuration Tools" chapter.

User EXEC Mode

The default command mode for the CLI is user EXEC mode. The EXEC commands available at the user EXEC level are a subset of those available at the privileged EXEC level. In general, the user EXEC commands allow you to connect to remote devices, change terminal settings on a temporary basis, perform basic tests, and list system information. The prompt for user EXEC mode is the name of the device followed by an angle bracket: `Router>`.

Privileged EXEC Mode

Privileged EXEC mode is password protected, and allows the use of all EXEC mode commands available on the system. To enter privileged EXEC mode from user EXEC mode, use the **enable** command. Privileged EXEC mode allows access to global configuration mode through the use of the **enable** command. The privileged EXEC mode prompt consists of the devices's host name followed by the pound sign: `Router#` .

Global Configuration Mode

Global configuration commands generally apply to features that affect the system as a whole, rather than just one protocol or interface. You can also enter any of the specific configuration modes listed in the following section from global configuration mode.

To enter global configuration mode, use the **configure terminal** privileged EXEC command. The router prompt for global configuration mode is indicated by the term *config* in parenthesis: `Router(config)#` .

ROM Monitor Mode

If your router or access server does not find a valid system image to load, the system will enter read-only memory (ROM) monitor mode. ROM monitor (ROMMON) mode can also be accessed by interrupting the boot sequence during startup. From ROM monitor mode, you can boot the device or perform diagnostic tests.

To enter ROM monitor mode, use the Break key (Cntl-C) during the first 60 seconds of start-up. The router prompt is indicated by an angle bracket by itself or the term ROMMON followed by a number and an angle bracket: `>` or `rommon1>` .

Setup Mode

Setup mode is not, strictly speaking, a command mode. Setup mode is rather an interactive facility that allows you to perform first-time configuration and other basic configuration procedures on all routers. The facility prompts you to enter basic information needed to start a router functioning. Setup mode uses the System Configuration Dialog, which guides you through the configuration process. It prompts you first for global parameters and then for interface parameters. The values shown in brackets next to each prompt are the default values. For more information on setup mode, see the "Using AutoInstall and Setup" chapter of this book.

To enter setup mode after the router has been configured for the first time, use the **setup** command in privileged EXEC mode. The router prompt for setup mode is indicated by a configuration question, followed by the default answer in brackets and a colon (:), as shown in the following example:

```
Continue with configuration dialog? [yes]:
Enter host name [Router]:
```

Configuration Modes and Submodes

Configuration modes are entered from global configuration mode. Configuration submodes are entered from other configuration modes. Configuration subsubmodes are configuration modes entered from configuration submodes.

The following configuration mode short summaries list the basic characteristics of each mode and where you can find details on the configuration tasks associated with each mode. Configuration modes and configuration submodes are listed here alphabetically.

All prompts listed are shown as they appear after the host name prompt on the system (for example, if the host name is "Router", the prompt for CA Identity configuration mode would be `Router(ca-identity)#`).

On most systems, a field of 30 characters is used for the host name and the prompt. Note that the length of your host name may cause longer configuration mode prompts to be truncated. For example, the full prompt for service profile configuration mode is:

```
(config-service-profile)#
```

However, if you are using the host-name of "Router", you will only see the following prompt (on most systems):

```
Router(config-service-profil)#
```

If the hostname is longer, you will see even less of the prompt:

```
Basement-rtr2(config-service)#
```

Keep this behavior in mind when assigning a name to your system (using the **hostname** global configuration command). If you expect that users will be relying on mode prompts as a CLI navigation aid, you should assign host names of no more than nine characters.

AAA Preauthentication Configuration Mode

Prompt: (config-preauth)

To enter AAA preauthentication configuration mode from global configuration mode, use the **aaa preauth** command. AAA preauthentication configuration mode allows you to configure preauthentication on the basis of the called number, the calling number, and the call type. This mode provides access to the following commands: **group**, **clid**, **ctype**, **dnis**, and **dnis bypass**.

For details, refer to the "Configuring Authentication" chapter of the Release 12.2 *Cisco IOS Security Configuration Guide*.

Access List Configuration Mode

See "Standard Named Access List (NACL) Configuration Mode" and "Extended Named Access List (NACL) Configuration Mode".

Access-point Configuration Mode

Prompt: (config-access-point)

To enter access-point configuration mode from access-point list configuration mode, use the **access-point** command. Use access-point configuration mode to specify the configuration characteristics of a GGSN access point.

For details, refer to the "Configuring Network Access to the GGSN" chapter in the "General Packet Radio Service (GPRS)" part of the Release 12.2 *Cisco IOS Mobile Wireless Configuration Guide*.

Access-point List Configuration Mode

Prompt: (config-ap-list)

To enter access-point list configuration mode from global configuration mode, use the **gprs access-point-list** command. Use access-point list configuration mode to define the general packet radio service (GPRS) access point list on a Gateway GPRS Support Node (GGSN).

The following submode is accessible from access-point list configuration mode:

- Access-point Configuration Mode

For details, refer to the "Configuring Network Access to the GGSN" chapter in the "General Packet Radio Service (GPRS)" part of the Release 12.2 *Cisco IOS Mobile Wireless Configuration Guide*.

Address Family Configuration Mode

Prompt: (config-router-af)

To enter address family configuration mode from router configuration mode, use the **address-family** command. Within this mode, you can configure address-family specific parameters for routing protocols, such as BGP, RGP, and static routing, that can accommodate multiple Layer 3 address families. The address family configuration mode commands include the **neighbor-activate** command and the **neighbor as-override** command. To exit address family configuration mode, use the **exit-address-family** command.

For details, refer to the "Configuring Multiprotocol Label Switching" chapter of the Release 12.2 *Cisco IOS Switching Services Configuration Guide*.

ALPS Circuit Configuration Mode

Prompt: (config-alps-circuit)

To enter Airline Product Set (ALPS) circuit configuration mode from global configuration mode, use the **alps circuit** command. Within ALPS circuit configuration mode, you can configure the tunneling mechanism that transports airline protocol data across a Cisco router-based TCP/IP network to an X.25-attached mainframe. This feature provides connectivity between agent set control units (ASCUs) and a mainframe host that runs the airline reservation system database.

For details, refer to the "Configuring the Airline Product Set" chapter of the Release 12.2 *Cisco IOS Bridging and IBM Networking Configuration Guide*.

ALPS ASCU Configuration Mode

Prompt: (config-alps-ascu) or (config-if-alps-ascu)

To enter Airline Product Set (ALPS) Agent Set Control Unit (ASCU) configuration mode from interface configuration mode, use the **apls ascu** command. Use ALPS ASCU configuration mode to configure ASCU characteristics on a specific interface.

For details, refer to the "Configuring the Airline Product Set" chapter of the Release 12.2 *Cisco IOS Bridging and IBM Networking Configuration Guide*.

Annex G Configuration Mode

Prompt: (config-annexg)

To enter Annex G Configuration Mode from global configuration mode, use the **call-router h323-annexg** command. Use Annex G Configuration Mode to configure an H.323 annex G border element (BE).

For details, refer to the 12.2(4)T "Cisco H.323 Scalability and Interoperability Enhancements" feature module.

The following example configures an Annex G BE that advertises both static and dynamic descriptors to its neighbors:

```
Router(config)# call-router h323-annexg be20
Router(config-annexg)# advertise all
```

The following submodes are accessible from Annex G Configuration Mode:

- Annex G Neighbor BE Configuration Mode

APPN Configuration Modes

Prompt: (appn)

The Advanced Peer-to-Peer Networking (APPN) configuration modes and configuration submodes were removed from the software in Cisco IOS Release 12.1. The configuration functionality that was previously provided by the APPN configuration modes has been replaced with SNA Switching Services (SNASw) functionality. SNA Switching uses existing configuration modes.

For details, refer to the "Configuring SNA Switching Services" chapter of the Release 12.2 *Cisco IOS Bridging and IBM Networking Configuration Guide*.

ATM VC Configuration Mode

Prompt: (config-if-atm-vc)

To enter ATM virtual circuit (VC) configuration mode from interface configuration mode or subinterface configuration mode, use the **pvc** command or the **svc nsap** command. Use ATM VC configuration mode to configure VC characteristics for an ATM permanent virtual circuit (PVC) or switched virtual circuit (SVC).

For details, refer to the "Configuring ATM" chapter of the Release 12.2 *Cisco IOS Wide-Area Networking Configuration Guide*.

ATM VC Bundle Configuration Mode

Prompt: (config-atm-bundle)

To enter ATM virtual circuit (VC) bundle configuration mode from interface configuration mode or subinterface configuration mode, use the **bundle** command. Use ATM bundle configuration mode to create and assign attributes and parameters to a bundle and all of its member virtual circuits (VCs).

The following configuration submode is accessible from ATM VC bundle configuration mode:

- ATM VC Bundle-Member Configuration Mode

For details, refer to the "Configuring IP to ATM Class of Service" chapter in the "Quality of Service Solutions" part of the Release 12.2 *Cisco IOS Quality of Service Solutions Configuration Guide*.

ATM VC Bundle-Member Configuration Mode

Prompt: (config-if-atm-member)

To enter ATM virtual circuit (VC) bundle-member configuration mode from ATM VC bundle configuration mode, use the **pvc-bundle** command. Use ATM VC bundle-member configuration mode to add a Virtual Circuit (VC) to a bundle as a bundle member, and configure the characteristics of that bundle member.

For details, refer to the "Configuring IP to ATM Class of Service" chapter in the "Quality of Service Solutions" part of the Release 12.2 *Cisco IOS Quality of Service Solutions Configuration Guide*.

ATM VC CES Configuration Mode

Prompt: (config-if-ces-vc)

To enter ATM virtual circuit (VC) circuit emulation service (CES) configuration mode from interface configuration mode, use the **pvc** or **svc** commands with the **ces** keyword. Use ATM VC CES configuration mode to configure VC parameters for an ATM CES permanent virtual circuit (PVC) or switched virtual circuit (SVC).

For details, refer to the "Configuring ATM" chapter of the Release 12.2 *Cisco IOS Wide-Area Networking Configuration Guide*.

ATM VC Class Configuration Mode

Prompt: (config-vc-class)

To enter ATM virtual circuit (VC) class configuration mode from global configuration mode, use the **vc-class atm** command.

Use ATM VC class configuration mode to configure a set of VC parameters that will apply to an ATM main interface, subinterface, PVC, or SVC. For example, you can create a VC class that contains VC parameter configurations that you will apply to a particular PVC or SVC. You might create another VC class that contains VC parameter configurations that you will apply to all VCs configured on a particular ATM main interface or subinterface.

For details, refer to the "Configuring ATM" chapter of the Release 12.2 *Cisco IOS Wide-Area Networking Configuration Guide*.

ATM-FR VC Group Configuration Mode

Prompt: (config-vc-group)

To enter ATM-Frame Relay (FR) virtual circuit (VC) group configuration mode from global configuration mode, use the **vc-group** command. Use ATM VC group configuration mode to map Frame Relay DLCIs to ATM VC groups for Frame Relay-ATM interworking.

For details, refer to the **vc-group** command documentation in the "Frame Relay-ATM Interworking Commands" chapter of the Release 12.2 *Cisco IOS Wide-Area Networking Command Reference*.

ATM PVC Range Configuration Mode

Prompt: (config-if-atm-range)

To enter ATM permanent virtual circuit (PVC) range configuration mode from subinterface configuration mode, use the **range** [*range-name*] **pvc** command. Use PVC range configuration mode to configure a number of ATM PVCs all at once rather than configuring the PVCs individually. PVC range configuration applies to multi-point sub-interfaces only.

For details, refer to the "Configuring Broadband Access: PPP and Routed Bridge Encapsulation" chapter in the Release 12.2 *Cisco IOS Wide-Area Networking Configuration Guide*.

ATM PVC-in-range Configuration Mode

Prompt: (cfg-if-atm-range-pvc)

To enter ATM permanent virtual circuit (PVC)-in-range configuration mode from ATM PVC range configuration mode, use the **pvc-in-range** command. Use ATM PVC-in-range configuration mode to explicitly configure an individual ATM PVC within a PVC range.

For details, refer to the "Configuring Broadband Access: PPP and Routed Bridge Encapsulation" chapter in the Release 12.2 *Cisco IOS Wide-Area Networking Configuration Guide*.

CA Identity Configuration Mode

Prompt: (ca-identity)

To enter certificate authority (CA) identity configuration mode from global configuration mode, use the **crypto ca identity** command. Use CA identity configuration mode to specify characteristics for certificate authorities.

For details, refer to the "Configuring Certification Authority Interoperability " chapter in the "IP Security and Encryption" part of the Release 12.2 *Cisco IOS Security Configuration Guide*.

CA Trusted-Root Configuration Mode

Prompt: (ca-root)

To enter certificate authority (CA) trusted-root configuration mode from global configuration mode, use the **crypto ca trusted-root** command. Use CA trusted-root configuration mode to specify the source for a root certificate.

For details, refer to the "Configuring Certification Authority Interoperability " chapter in the "IP Security and Encryption" part of the Release 12.2 *Cisco IOS Security Configuration Guide*.

Call Discriminator Configuration Mode

See (Resource-Pool) Call Discriminator Profile Configuration Mode.

Called-Group Configuration Mode

See Dialer DNIS Group Configuration Mode.

CASA Configuration Mode

Prompt: (config-casa)

To enter Cisco Appliance Services architecture (CASA) configuration mode from global configuration mode, use the **ip casa** command. Use CASA configuration mode to configure CASA listen ports, such as the MNLB forwarding agent.

For details, refer to the "Configuring IP Services" chapter in the "IP Addressing and Services" part of the Release 12.2 *Cisco IOS IP Configuration Guide*. For further background information, refer to the white paper "High Availability Web Services" and the *MultiNode Load Balancing Feature Set for LocalDirector User Guide* (available on Cisco.com).

CAS Custom Configuration Mode

Prompt: (config-ctrl-cas)

To enter CAS custom configuration mode from controller configuration mode, use the **cas-custom** command. Use CAS custom configuration mode to customize E1 R2 signaling parameters for a particular E1 channel group on a channelized E1 line.

Some switches require you to fine tune your R2 settings. However, do not tamper with these special signaling commands unless you understand exactly how your switch will be effected.

For details, refer to the "Configuring ISDN PRI" chapter in the "Signaling Configuration" part of the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

CES Configuration Mode

Prompt: (config-ces)

To enter circuit emulation service (CES) configuration mode from global configuration mode, use the **ces** command. Use CES configuration mode to configure CES parameters such as the CES clock.

For details, refer to the "Configuring ATM" chapter of the Release 12.2 *Cisco IOS Wide-Area Networking Configuration Guide*.

Certificate Chain Configuration Mode

Prompt: (config-cert-chain)

To enter certificate chain configuration mode from global configuration mode, use the **crypto ca certificate chain** command. Use certificate chain configuration mode to delete certificates using the **no certificate** command.

For details, refer to the "Configuring Certification Authority Interoperability " chapter in the "IP Security and Encryption" part of the Release 12.2 *Cisco IOS Security Configuration Guide*.

Class Map Configuration Mode

See QoS Class-Map Configuration Mode.

Controller Configuration Mode

Prompt: (config-controller)

To enter controller configuration mode from global configuration mode, use the **controller** command. Use controller configuration mode to configure channelized T1 or E1.

The following submode is accessible through controller configuration mode:

- CAS Custom Configuration Mode

For details, refer to the "Configuring ISDN PRI" chapter in the "Signaling Configuration" part of the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

Crypto Map Configuration Mode

Prompt: (config-crypto-map)

To enter crypto map configuration mode from global configuration mode, use the **cypto map** command. Use crypto map configuration mode to create or alter the definition of a crypto-map. Crypto-maps are part of an authentication and encryption router configuration.

For details, refer to the "Configuring IPSec Network Security " chapter in the Release 12.2 *Cisco IOS Security Configuration Guide*.

Crypto Transform Configuration Mode

Prompt: (config-crypto-trans)

To enter crypto transform configuration mode from global configuration mode, use the **crypto ipsec transform-set** command. Use crypto transform configuration mode to change the initialization vector length for the esp-rfc1829 transform, or to change the transform-set to tunnel or transport mode.

For details, refer to the "Configuring IPSec Network Security" chapter in the Release 12.2 *Cisco IOS Security Configuration Guide*.

Customer Profile Configuration Mode

See (Resource-Pool) Customer Profile Configuration Mode.

DHCP Pool Configuration Mode

Prompt: (config-dhcp)

To enter DHCP pool configuration mode from global configuration mode, use the **ip dhcp pool** command. Use DHCP pool configuration mode to configure DHCP pool parameters, such as the IP subnet number and the default router list.

For details, refer to the "Configuring DHCP" chapter in the Release 12.2 *Cisco IOS IP Configuration Guide*.

Dial Peer Voice Configuration Mode

Prompt: (config-dialpeer)

To enter dial peer voice configuration mode from global configuration mode, use the **dial peer voice** command. Use dial-peer configuration mode to configure dial peers for Voice over IP, Voice over ATM, Voice over Frame Relay, and Voice over HDLC.

For details, refer to the chapters on the above technologies in the "Voice" part of the Release 12.2 *Cisco IOS Voice, Video, and Fax Configuration Guide*.

Dial Peer COR List Configuration Mode

Prompt: (config-cor)

To enter dial peer class of restrictions (COR) list configuration mode from global configuration mode, use the **dial-peer cor list** *list-name* command. Use dial peer COR list configuration mode to add members to the list of restrictions.

For details, refer to the **dial peer cor list** command description in the Release 12.2 *Cisco IOS Dial Technologies Command Reference*.

Dialer DNIS Group Configuration Mode

Prompt: (config-dnis-group)

To enter dialer called group configuration mode from global configuration mode, use the **dialer dnis group** command. Use dialer called group configuration mode to add a DNIS number to a dialer-called-group (DNIS group). (DNIS groups can be used to accept or reject calls when used with other Cisco software features, such as resource pool management.)

For details, refer to the description of the **dialer dnis group** command in the Release 12.2 *Cisco IOS Dial Technologies Command Reference*.

DLUR Configuration Mode

See TN3270 DLUR Configuration Mode.

DNIS Group Configuration Mode

See Dialer DNIS Group Configuration Mode.

Extended Named Access List (NACL) Configuration Mode

Prompt: (config-ext-nacl)

To enter extended named access list configuration mode from global configuration mode, use the **ip access-list** or **ipx access list** command. Use access-list configuration mode to create a named IP or IPX access list.

For information on creating a named IP access list, refer to the "Configuring IP Services" chapter in the "IP Addressing and Services" part of the Release 12.2 *Cisco IOS IP Configuration Guide*. For information on creating a named IPX access list, refer to the "Configuring Novell IPX" chapter in the Release 12.2 *Cisco IOS AppleTalk and Novell IPX Configuration Guide* .

Frame Relay DLCI Configuration Mode

Prompt: (config-fr-dlci)

To enter Frame Relay DLCI configuration mode from interface configuration mode, use the **frame-relay interface-dlci** command. Use Frame Relay DLCI configuration mode to assign a Voice over Frame Relay (VoFR) FRF.11 encapsulation to a Frame Relay DLCI using the **vofr** Frame Relay DLCI configuration command.

For details, refer to the **frame-relay interface-dlci**, **frame-relay interface-dlci switched**, and **vofr** command documentation in the Release 12.2 *Cisco IOS Wide-Area Networking Configuration Guide* and the Release 12.2 *Cisco IOS Voice, Video, and Fax Command Reference.*

Frame Relay Congestion Management Configuration Mode

Prompt: (config-fr-congest)

To enter Frame Relay congestion management configuration mode from interface configuration mode, use the **frame-relay congestion-management** command. Use Frame Relay congestion management configuration mode to configure Frame Relay congestion management parameters for switched PVCs on a Frame Relay interface.

For details, refer to the "Configuring Frame Relay" chapter in the Release 12.2 *Cisco IOS Wide-Area Networking Configuration Guide*.

FRF.5 / FRF.8 Configuration Mode

Prompt: (config-frf5) or (config-frf8)

To enter FRF.5 or FRF.8 configuration mode from global configuration mode, use the **connect** command. Use FRF configuration mode to create a connection between a Frame Relay DLCI and an ATM PVC, to configure Frame Relay DE field mapping, or to set ATM CLP fields.

For details, refer to the **connect** command documentation in the "Configuring Frame Relay-ATM Interworking" chapter of the Release 12.2 *Cisco IOS Wide-Area Networking Configuration Guide*.

Gatekeeper Configuration Mode

Prompt: (config-gk)

Use gatekeeper configuration mode to configure a Cisco 2500 series, Cisco 3620, Cisco 3640, or Cisco MC3810A router as a multimedia conference manager Gatekeeper. On these platforms, use the **gatekeeper** command in global configuration mode to enter gatekeeper configuration mode.

For details, refer to the "Configuring Gatekeepers (Multimedia Conference Manager)" chapter in the "Voice" part of the Release 12.2 *Cisco IOS Voice, Video, and Fax Configuration Guide*. For additional details, refer to the 12.0(3)T "Multimedia Conference Manager" feature module.

Gateway Configuration Mode

Prompt: (config-gateway)

To enter gateway configuration mode from global configuration mode, use the **gateway** command. Use gateway configuration mode to configure gateway operating characteristics, such as security.

For details, refer to the "Configuring Voice over IP" chapter in the "Voice" part of the Release 12.2 *Cisco IOS Voice, Video, and Fax Configuration Guide*.

Hex Input Mode

See Public-Key Hex Input Configuration Mode.

HTTP Raw Request Configuration Mode

See SAA HTTP Raw Request Configuration Mode.

Hub Configuration Mode

Prompt: (config-hub)

To enter hub configuration mode from global configuration mode, use the **hub** command. Use hub configuration mode to configure hub functionality for an Ethernet interface on the Cisco 2500 series.

For details, refer to the the "Configuring LAN Interfaces" chapter in the Release 12.2 *Cisco IOS Interface Configuration Guide*.

IBM Channel Configuration Mode

IBM channel configuration mode is the same as interface configuration mode. Enter interface channel configuration mode from global configuration mode by using the **interface channel** form of the interface command.

For details, refer to the "Configuring Cisco Mainframe Channel Connection (CMCC) Adapters" chapter in the "IBM Networking" part of the Release 12.2 *Cisco IOS Bridging and IBM Networking Configuration Guide*.

IBM Channel Internal Adapter Configuration Mode

Prompt: (cfg-adap-*type n-m*)

To enter IBM channel internal adapter configuration mode from IBM channel internal LAN interface configuration mode, use the **adapter** command. Use internal adapter configuration mode to configure the link characteristics for the internal LAN adapter and name the internal LAN adapter. To configure an internal adapter interface, you must first use the **bridge-group** internal LAN configuration command or the **source-bridge** internal LAN configuration command to configure bridging type.

For details, refer to the **adapter** command documentation in the "Cisco Mainframe Channel Connection (CMCC) Commands" chapter in the "IBM Networking" part of the Release 12.2 *Cisco IOS Bridging and IBM Networking Command Reference, Volume 2*.

IBM Channel Internal LAN Interface Configuration Mode

Prompt: (cfg-lan-*type n*)

To enter internal LAN configuration mode from interface configuration mode, use the **lan** command. Use the IBM channel internal LAN configuration mode to configure an internal LAN on a CIP interface and configure Cisco Systems Network Architecture (CSNA) parameters.

The following configuration mode is accessible through internal LAN configuration mode:

- IBM Channel Internal Adapter Configuration Mode

For details, refer to the "Configuring Cisco Systems Network Architecture (CSNA) and Cisco Multipath Channel (CMPC)" chapter in the "IBM Networking" part of the Release 12.2 *Cisco IOS Bridging and IBM Networking Configuration Guide*.

Interface Configuration Mode

Prompt: (config-if)

To enter interface configuration mode from global configuration mode, use an **interface** command. Many features are enabled on a per-interface basis. Subinterface configuration mode is accessible through interface configuration mode.

In addition to subinterface configuration mode, the following configuration submodes are accessible through interface configuration mode:

- ATM VC Configuration Mode
- ATM VC Bundle Configuration Mode
- ATM VC Bundle-Member Configuration Mode
- Frame Relay DLCI Configuration Mode
- Frame Relay Congestion Management Configuration Mode
- IP Host Backup Configuration Mode
- IBM Channel Configuration Mode
- IBM Channel Internal LAN Interface Configuration Mode
  - IBM Channel Internal Adapter Configuration Mode
- RLM Group Configuration Mode
  - RLM Device Configuration Mode

Note Many configuration modes available through interface configuration mode are also available in subinterface configuration mode.

IP Host Backup Configuration Mode

Prompt: (config-if-path)

To enter IP host backup configuration mode from interface configuration mode, use the **path** command. IP host backup mode is used to configure the IP host backup paths on an interface.

For details, refer to the descriptions of the **path**, **claw**, and **offload** commands in the "CLAW and TCP/IP Offload Commands" chapter in the "IBM Networking" part of the Release 12.2 *Cisco IOS Bridging and IBM Networking Command Reference, Volume 2*.

IPv6 Access List Configuration Mode

Prompt: (config-ipv6-acl)

Introduced in "IPv6 Extended ACL Support," 12.2(13)T

To enter IPv6 access list configuration mode from global configuration mode, use the ipv6 access-list command. Use the IPv6 access list configuration mode to specify the permit and deny parameters for an IPv6 access list.

For details, refer to the *Implementing Security for IPv6* documentation module at:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t13/ipv6/ipv6imp/sa_secv6.htm

The following example configures an IPv6 access list named outbound that defines HTTP access to be restricted to certain hours during the day, and to log any activity outside of the permitted hours.

```
Router(config)# time-range lunchtime
Router(config)# periodic weekdays 12:00 to 13:00
Router(config)# ipv6 access-list OUTBOUND
Router(config-ipv6-acl)# permit tcp any any eq www time-range lunchtime
Router(config-ipv6-acl)# deny tcp any any eq www log-input
Router(config-ipv6-acl)# permit tcp 2000:1::/64 any
Router(config-ipv6-acl)# permit udp 2000:1::/64 any
```

The ipv6 access-list command changed its syntax in Cisco IOS Release 12.2(13)T as the IPv6 access list configuration mode was added. Previous T train releases contained the permit and deny keywords, and associated arguments, within the ipv6 access-list command syntax. Examples of the syntax in the previous releases are documented in the *Implementing Security for IPv6* module. If an IPv6 access list configuration from a previous release is used in Cisco IOS Release 12.2(13)T, the configuration is translated to use the new IPv6 access list configuration mode.

IP VPN Routing/Forwarding (VRF) Instance Configuration Mode

See VRF Configuration Mode.

IPX Router Configuration Mode

Prompt: (config-ipx-router)

To enter Novell Internet Packet Exchange (IPX) router configuration mode from global configuration mode, use the **ipx router** command. Use IPX router configuration mode to configure IPX routing characteristics, such as route distribution. Note that IPX must first be enabled using the **ipx routing** command.

For details, refer to the "Configuring Novell IPX" chapter in the Release 12.2 *Cisco IOS AppleTalk and Novell IPX Configuration Guide* .

ISAKMP Policy Configuration Mode

Prompt: (config-isakmp)

To enter Internet Security Association and Key Management Protocol (ISAKMP) policy configuration mode from global configuration mode, use the **crypto isakmp policy** command. Use ISAKMP to define an Internet Key Exchange (IKE) policy (ISAKMP is a security protocol implemented by IKE). IKE policies define a set of parameters to be used during the IKE negotiation.

For details, refer to the "Configuring Internet Key Exchange Security Protocol" chapter of the "IP Security and Encryption" part of the Release 12.2 *Cisco IOS Security Configuration Guide*.

Key-Chain Configuration Mode

Prompt: (config-keychain)

To enter key-chain configuration mode from global configuration mode, use the **keychain** command. Use key-chain configuration mode to configure authentication keys.

The following submode is accessible though key-chain configuration mode:

- Key-Chain Key Configuration Mode

For details, refer to the "Managing Authentication Keys" section in the "Configuring IP Routing Protocol-Independent Features" chapter of the Release 12.2 *Cisco IOS IP Configuration Guide*.

Key-Chain Key Configuration Mode

Prompt: (config-keychain-key)

To enter key-chain key configuration mode from key-chain configuration mode, use the **key** command. Use key-chain key configuration mode to configure a specific authentication key in a key-chain.

For details, refer to the "Managing Authentication Keys" section in the "Configuring IP Routing Protocol-Independent Features" chapter of the Release 12.2 *Cisco IOS IP Configuration Guide*.

LANE Database Configuration Mode

Prompt: (lane-config-database)

To enter LAN emulation (LANE) database configuration mode from global configuration mode, use the **lane database** command.

A LANE database contains entries that bind an emulated LAN name to the ATM address of the LANE server, bind LANE client MAC addresses to an emulated LAN name, and bind LANE client ATM address templates to an emulated LAN name.Use LANE database configuration mode to create entries for a specified database.

For details, refer to the "Configuring LAN Emulation" chapter in the "LAN Emulation" part of the Release 12.2 *Cisco IOS Switching Services Configuration Guide*.

Line Configuration Mode

Prompt: (config-line)

To enter line configuration mode from global configuration mode, use a form of the **line** command. Use line configuration mode to modify the operation of an auxiliary, console, physical, or virtual terminal line. Line configuration commands always follow a **line** command, which defines a line number. These commands are generally used to connect to remote routers or access servers, change terminal parameter settings either on a line-by-line basis or for a range of line, and set up the auxiliary port modem configuration to support dial-on-demand routing (DDR).

For common line configuration tasks, refer to the "Modem and Dial Shelf Configuration and Management" part of the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

Listen-Point Configuration Mode

See TN3270 Listen-Point Configuration Mode.

Map Class Configuration Mode

See Static Maps Class Configuration Mode.

Map-List Configuration Mode

See Static Maps List Configuration Mode.

Modem Pool Configuration Mode

Prompt: (config-modem-pool)

To enter modem pool configuration mode from global configuration mode, use the **modem-pool** command. A modem pool is a group of modems inside an access server that are assigned a single dialed number identification service number (DNIS). Use modem pool configuration mode to create multiple pools of physical modems, assign unique DNIS numbers to each modem pool, and set maximum simultaneous connect limits.

For details, refer to the "Managing Modems" chapter in the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

MPOA Client (MPC) Configuration Mode

Prompt: (mpoa-client-config)

To enter Multiprotocol over ATM (MPOA) client (MPC) configuration mode from global configuration mode, use the **mpoa client config name** command. Use MPOA client configuration mode to optionally change MPOA client operating parameters.

For details, refer to the "Configuring the Multiprotocol over ATM Client" chapter in the "LAN Emulation" part of the Release 12.2 *Cisco IOS Switching Services Configuration Guide*.

MPOA Server (MPS) Configuration Mode

Prompt: (mpoa-server-config)

To enter Multiprotocol over ATM (MPOA) server configuration mode from global configuration mode, use the **mpoa server config name** command. Use MPOA server configuration mode to optionally change MPOA server operating parameters.

For details, refer to the "Configuring the Multiprotocol over ATM Server" chapter in the "LAN Emulation" part of the Release 12.2 *Cisco IOS Switching Services Configuration Guide*.

MRM Manager Configuration Mode

Prompt: (config-mrm-manager)

To enter Multicast Routing Monitor (MRM) manager configuration mode from global configuration mode, use the **ip mrm manager** command. Use MRM manager configuration mode to configure a router interface to be a Manager for a MRM test. MRM manager configuration mode commands also configure beacon message characteristics, Test Sender parameters, and Test Receiver parameters.

For details, refer to the "Using IP Multicast Tools" chapter of the Release 12.2 *Cisco IOS IP Configuration Guide*.

Policy-Map Configuration Mode

See QoS Policy-Map Configuration Mode and QoS Policy-Map Class Configuration Mode.

Poll-Group Configuration Mode

See System Controller Poll-Group Configuration Mode.

Public-Key Chain Configuration Mode

Prompt: (config-pubkey-c)

To enter public key chain configuration mode from global configuration mode, use the **crypto key pubkey-chain rsa** command. Use public key chain configuration mode to manually specify other IPSec peers' RSA or DSS public keys.

From public-key chain configuration mode, you can enter the following submodes:

- Public-Key Key Configuration Mode
  - Public-Key Hex Input Configuration Mode

For details, refer to the "Configuring Internet Key Exchange Security Protocol" chapter in the "IP Security and Encryption" part of the Release 12.2 *Cisco IOS Security Configuration Guide*.

Public-Key Key Configuration Mode

Prompt: (config-pubkey-k)

To enter public-key key configuration mode from public-key chain configuration mode, use the **addressed-key** or **named-key** public key chain configuration commands puts you into public key configuration mode. In this mode you can specify RSA or DSS public keys. The following submode is accessible through public-key key configuration mode:

- Public-Key Hex Input Configuration Mode

For details, refer to the "Configuring Internet Key Exchange Security Protocol" chapter in the "IP Security and Encryption" part of the Release 12.2 *Cisco IOS Security Configuration Guide*.

Public-Key Hex Input Configuration Mode

Prompt: (config-pubkey)

To enter public-key hex input configuration mode from public-key key configuration mode, use the **key-string** command. Use public-key hex input configuration mode to manually specify a remote peer's RSA public key for an encrypting peer router. The public key data is entered in hexadecimal form, and it will take more than one command line to enter. To continue entering the public key data on a new line, press Return. When the public key hex data is completely entered, press Return to get a new line, then type **quit** to return to public-key key configuration mode.

For details, refer to the "Configuring Internet Key Exchange Security Protocol" chapter in the "IP Security and Encryption" part of the Release 12.2 *Cisco IOS Security Configuration Guide*.

QoS Class-Map Configuration Mode

Prompt: (config-cmap)

To enter Quality of Service (QoS) class-map configuration mode from global configuration mode, enter the **class-map** command. Use class-map configuration mode to define a traffic class.

Also referred to as "QoS Class-map Configuration Mode," this mode was introduced in Cisco IOS Releases 12.0(5)XE and 12.1(5)T. The mode is related to the QoS Policy-Map Configuration Mode.

For details, refer to the "Configuring Multiprotocol Label Switching" chapter in the "Multiprotocol Label Switching" part of the Release 12.2 *Cisco IOS Switching Services Configuration Guide* and the "Configuring the Modular Quality of Service Command-Line Interface" chapter of the Release 12.2 *Cisco IOS Quality of Service Solutions Configuration Guide*.

QoS Policy-Map Configuration Mode

Prompt: (config-pmap)

To enter Quality of Service (QoS) policy map configuration mode from global configuration mode, enter the **policy-map** command. The **policy-map** command is used to define the characteristics of a service policy. The first step in creating a service policy is associating a traffic class with one or more Quality of Service (QoS) policies. The associated traffic class is defined by using the **class** command in policy map configuration mode.

For details, refer to the "Configuring Multiprotocol Label Switching" chapter in the "Multiprotocol Label Switching" part of the Release 12.2 *Cisco IOS Switching Services Configuration Guide* and the "Configuring Weighted Fair Queueing" chapter in the "Congestion Management" part of the Release 12.2 *Cisco IOS Quality of Service Solutions Configuration Guide*.

QoS Policy-Map Class Configuration Mode

Prompt ID: (config-pmap-c)

To enter Quality of Service (QoS) policy-map class configuration mode from policy-map configuration mode, enter the **class** command. After defining the associated traffic class, the router is automatically in policy-map class configuration mode. Use policy-map class configuration mode to define the Quality of Service (QoS) policies for a particular service policy.

For details, refer to the "Configuring Multiprotocol Label Switching" chapter in the "Multiprotocol Label Switching" part of the Release 12.2 *Cisco IOS Switching Services Configuration Guide* and the "Configuring Weighted Fair Queueing" chapter in the "Congestion Management" part of the Release 12.2 *Cisco IOS Quality of Service Solutions Configuration Guide*.

RADIUS Server Group Configuration Mode

See Server Group RADIUS Configuration Mode.

RED Group Configuration Mode

Prompt: (config-red-group)

To enter Random Early Detection (RED) configuration mode from global configuration mode, use the **random-detect-group** command. Use RED configuration mode to define the Weighted Random Early Detection (WRED) parameter group. (Note that the **service-policy output** and **random-detect-group** commands are mutually exclusive; before you can configure one command, you must disable the other if it is configured.)

For details, refer to the "Configuring IP to ATM Class of Service" chapter in the "Quality of Service Solutions" part of the Release 12.2 *Cisco IOS Quality of Service Solutions Configuration Guide*.

RLM Group Configuration Mode

Prompt: (config-rlm-group)

To enter Redundant Link Manager (RLM) group configuration mode from interface configuration mode, use the **rlm group** command. Use RLM group configuration mode to configure the RLM group (network access server).

The following configuration submode is accessible through RLM group configuration mode:

- RLM Device Configuration Mode

For details, refer to the "Configuring the Cisco SS7/C7 Dial Access Solution System" chapter in the "Dial Access Specialized Features" part of the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

RLM Device Configuration Mode

Prompt: (config-rlm-group-sc)

To enter Redundant Link Manager (RLM) device configuration mode from RLM group configuration mode, use the **server** command. Use RLM device configuration mode to specify configuration options for the RLM network access server, such as link addresses and weighting preferences.

For details, refer to the "Configuring the Cisco SS7/C7 Dial Access Solution System" chapter in the "Dial Access Specialized Features" part of the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

Resource Group Configuration Mode

See (Resource-Pool) Resource Group Configuration Mode.

(Resource-Pool) Call Discriminator Profile Configuration Mode

Prompt: (config-call-discriminator)

To enter resource-pool call discriminator profile configuration mode from global configuration mode, use the **resource-pool profile discriminator** command. Use call discriminator profile configuration mode to specify a list of calling party numbers to be rejected for inbound calls.

For details, refer to the "Configuring Resource Pool Management (RPM)" chapter in the "Dial Access Specialized Features" part of the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

(Resource-Pool) Customer Profile Configuration Mode

Prompt: (config-customer-profile)

To enter resource-pool customer profile configuration mode from global configuration mode, use the **resource-pool profile customer** command. (To use resource-pool configuration modes, use should first enable resource pool management using the **resource-pool enable** global configuration command.) Use the customer profile configuration mode to include a group of DNIS numbers in a customer profile.

For details, refer to the "Configuring Resource Pool Management (RPM)" chapter in the "Dial Access Specialized Features" part of the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

(Resource-Pool) Resource Group Configuration Mode

Prompt: (config-resource-group)

To enter resource-pool resource group configuration mode from global configuration mode, use the **resource-pool group resource** command. (To use resource-pool configuration modes, use should first enable resource pool management using the **resource-pool enable** global configuration command.) Use resource group configuration mode to associate a range of modems or other physical resources with a resource group for Resource Pool Management.

For details, refer to the "Configuring Resource Pool Management (RPM)" chapter in the "Dial Access Specialized Features" part of the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

(Resource-Pool) Service Profile Configuration Mode

Prompt: (config-service-profile)

To enter resource-pool service profile configuration mode from global configuration mode, use the **resource-pool profile service** command. (To use resource pool configuration modes, use should first enable resource pool management using the **resource-pool enable** global configuration command.) Use service profile configuration mode to configure modem service parameters for devices used by the Resource Pool Manager (RPM).

For details, refer to the "Configuring Resource Pool Management (RPM)" chapter in the "Dial Access Specialized Features" part of the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

(Resource-Pool) VPDN Profile Configuration Mode

Prompt: (config-vpdn-profile)

To enter resource-pool virtual private dialup network (VPDN) profile configuration mode from global configuration mode, use the **resource-pool profile vpdn** command. (To use resource-pool configuration modes, use should first enable resource pool management using the **resource-pool enable** global configuration command.) Use call VPDN profile configuration mode to configure a VPDN resource pool managment profile.

For details, refer to the "Configuring Resource Pool Management (RPM)" chapter in the "Dial Access Specialized Features" part of the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

Route-Map Configuration Mode

Prompt: (config-route-map)

To enter route-map configuration mode from global configuration mode, use the **route-map** (IP) command. Use the route-map configuration mode to configure routing table source and destination information. For details, refer to the "Configuring IP Routing Protocol-Independent Features" chapter in the "IP Routing Protocols" part of the Release 12.2 *Cisco IOS IP Configuration Guide*.

Router Configuration Mode

Prompt: (config-router)

Router configuration commands configure an IP routing protocol and always follow a **router** command.

The following submodes are accessible from router configuration mode:

- Address Family Configuration Mode

For details, refer to the relevant protocol chapter in the "IP Routing Protocols" part of the Release 12.2 *Cisco IOS IP Configuration Guide*.

RTR Entry Configuration Mode

Prompt: (config-rtr)

To enter response time reporter (RTR) entry configuration mode from global configuration mode, use the **rtr** command. Use RTR configuration mode to configure Cisco Service Assurance Agent (SAA) operations for the measurement of response times and availability.

The following submode is accessible from RTR configuration mode:

- SAA HTTP Raw Request Configuration Mode

For details, refer to the "Network Monitoring Using Cisco Service Assurance Agent" chapter in this book.

SAA HTTP Raw Request Configuration Mode

Prompt: (config-rtr-http)
Aliases: RTR HTTP Raw Request Configuration Mode

To enter SAA HTTP raw configuration mode from RTR configuration mode or RTR Entry Configuration Mode, use the **http-raw-request** command. Use HTTP Raw Request Configuration Mode to explicitly specify the options for an SAA HTTP operation using HTTP 1.0 commands. The SAA HTTP operation determines the amount of time it takes for an HTTP request from your device to be serviced.

For details, see the "Network Monitoring Using Cisco Service Assurance Agent" chapter in this book.

Server Group RADIUS Configuration Mode

Prompt: (config-sg-radius)

To enter server group RADIUS configuration mode from global configuration mode, use the **aaa group server radius** command.

For details on the **aaa group server radius** command, refer to the "RADIUS Commands" chapter in the "Security Server Protocols" part of the Release 12.2 *Cisco IOS Security Command Reference*. For additional information, refer to the corresponding chapter in the Release 12.2 *Cisco IOS Security Configuration Guide*.

Server Group TACACS+ Configuration Mode

Prompt: (config-sg-tacac)

To enter server group TACACS+ configuration mode from global configuration mode, use the **aaa group server tacacs+** command.

For details on the **aaa group server tacacs+** command, refer to the "TACACS+ Commands" chapter in the "Security Server Protocols" part of the Release 12.2 *Cisco IOS Security Command Reference*. For additional information, refer to the corresponding chapter in the Release 12.2 *Cisco IOS Security Configuration Guide*.

Service Profile Configuration Mode

See (Resource-Pool) Service Profile Configuration Mode.

SLB DFP Configuration Mode

Prompt: (config-slb-dfp)

To enter server load balancing (SLB) dynamic feedback protocol (DFP) configuration mode from global configuration mode, use the **ip slb dfp** command. Use server load balancing DFP configuration mode to configure the Dynamic Feedback Protocol, which is a mechanism that allows host agents in load-balanced environments to dynamically report the change in status of the host systems providing a virtual service.

For details, refer to the "Configuring Server Load Balancing" chapter in the Release 12.2 *Cisco IOS IP Configuration Guide*.

SLB Real Server Configuration Mode

Prompt: (config-slb-real)

To enter server load balancing (SLB) real server configuration mode from server load balancing server-farm configuration mode, use the **real** command. Use real server configuration mode to identify a real server in your network. A *virtual* server can be defined that represents a group of *real* servers in a cluster of networks called a *server farm*. The real servers are the physical devices that provide the load-balanced services.

For details, refer to the "Configuring Server Load Balancing" chapter in the Release 12.2 *Cisco IOS IP Configuration Guide*..

SLB Server-Farm Configuration Mode

Prompt: (config-slb-sfarm)

To enter server load balancing (SLB) server-farm configuration mode from global configuration mode, use the **ip slb serverfarm** command. Use server farm configuration mode to group real servers into server farms. Using server farms enables Cisco IOS server load balancing to assign new connections to the real servers based on their weighted capacities, and on the load algorithms used.

For details, refer to the "Configuring Server Load Balancing" chapter in the Release 12.2 *Cisco IOS IP Configuration Guide*.

SLB Virtual Server Configuration Mode

Prompt ID: (config-slb-vserver)

To enter server load balancing (SLB) virtual server configuration mode from global configuration mode, use the **ip slb vserver** command. Use virtual server configuration mode to specify a virtual server that represents a group of real servers.

For details, refer to the "Configuring Server Load Balancing" chapter in the Release 12.2 *Cisco IOS IP Configuration Guide*.

SPE Configuration Mode

Prompt ID: (config-spe)

To enter Service Processing Element (SPE; also referred to as the Software Port Entity) configuration mode from global configuration mode, use the **spe** command. Use SPE configuration mode to copy firmware upgrades to a specified modem or modems. The modems are identified in the CLI using their SPE numbers.

For details, refer to the "Configuring and Managing Cisco Access Servers and Dial Shelves" chapter in the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

Standard Named Access List (NACL) Configuration Mode

Prompt: (config-std-nacl)

All IP and IPX access lists can be identified by a number. Alternatively, some IP and IPX access lists can be identified by a name. Use access-list configuration mode when you are creating a named IP or IPX access list.

For information on creating a named IP access list, refer to the "Configuring IP Services" chapter in the Release 12.2 *Cisco IOS IP Configuration Guide*. For information on creating a named IPX access list, refer to the "Configuring Novell IPX" chapter in the Release 12.2 *Cisco IOS AppleTalk and Novell IPX Configuration Guide* .

Static Maps Class Configuration Mode

Prompt: (config-map-class)

To enter static maps class configuration mode from global configuration mode, use the **map-class** global configuration mode command. Use static maps class configuration mode to configure parameters for Frame Relay, ATM, or Dialer encapsulation protocols.

The **map-class dialer** command allows you to specify different characteristics for different types of calls on a per-call-destination basis. For example, you can specify higher priority and a lower wait-for-carrier time for an ISDN-calls map class than for a modem-calls map class. You can also specify a different speed for some ISDN calls than for other ISDN calls. For details, refer to the "Configuring PPP Callback" chapter in the Callback and Bandwidth Allocation Configuration part of the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

The **map-class frame-relay** command allows you to specify parameters that control the traffic that the source router will send over a switched virtual circuit (SVC). For details, refer to the "Configuring Frame Relay" chapter of the Release 12.2 *Cisco IOS Wide-Area Networking Configuration Guide*.

Note that the **map-class atm** command is not supported in Cisco IOS Release 12.0 and later.

Use the **exit-class** command to exit from static maps class configuration mode.

Static Maps List Configuration Mode

Prompt: (config-map-list)

To enter static maps list configuration mode from global configuration mode, use the **map-list** command. Use static maps list configuration mode to define the protocol addresses and associate each protocol address with a specific map class. Static maps list configuration mode commands take the form *protocol* [*address*] **class** (for example, **aarp class**, **apollo 1.2 class**, **cdp class**, **ip 1.2.3.4 class**, and so on).

For details, refer to the **class (map-list)** command documentation in the "Frame Relay Commands" chapter of the Release 12.2 *Cisco IOS Wide-Area Networking Command Reference*.

Subinterface Configuration Mode

Prompt: (config-subif)

To enter subinterface configuration mode from interface configuration mode, use an **interface** command. Use subinterface configuration mode to configure multiple virtual interfaces (called subinterfaces) on a single physical interface.

Subinterfaces appear to be distinct physical interfaces to the various protocols. For example, Frame Relay networks provide multiple point-to-point links called permanent virtual circuits (PVCs). PVCs can be grouped under separate subinterfaces that in turn are configured on a single physical interface. From a bridging spanning-tree viewpoint, each subinterface is a separate bridge port, and a frame arriving on one subinterface can be sent out on a another subinterface.

For details on how to configure subinterfaces, refer to the appropriate documentation module for a specific protocol in the Cisco IOS software documentation.

System Controller Poll-Group Configuration Mode

Prompt: (config-poll-group)

To enter system controller poll-group configuration mode from global configuration mode, use the **syscon poll-group** command. Use system controller poll-group configuration mode to configure data collection for a specific poll group using a system controller. The poll-group configuration mode is required for Performance Data Collection, which allows a system controller to collect and store SNMP MIB data from its managed router and dial shelves.

For details, refer to the "Configuring and Managing Cisco Access Servers and Dial Shelves" chapter in the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

Time Range Configuration Mode

Prompt: (config-time-range)

To enter time range configuration mode from global configuration mode, use the **time-range** command. Use time range configuration mode to define a time range consisting of specific times of the day and week. Apply the time range to a function that accepts time ranges to control when that function will occur. For example, the time range that you define can be referenced in IP extended access lists and IPX extended access lists.

For details, see the "Performing Basic System Management" chapter in this book.

TN3270 Server Configuration Mode

Prompt: (cfg-tn3270)

The TN3270 server provides a set of configuration modes and submodes for configuring the TN3270 Server feature on a CMCC adapter. For CIP adapters, the TN3270 server is configured on the virtual interface, which is always port 2. For CPA adapters, the TN3270 server feature is always configured on port 0.

To enter TN3270 server configuration mode from interface configuration mode, enter the **tn3270-server** command.

The following configuration submodes are accessible through TN3270 configuration mode:

- TN3270 DLUR Configuration Mode
  - TN3270 DLUR PU Configuration Mode
  - TN3270 DLUR Linked SAP Configuration Mode
- TN3270 Listen-Point Configuration Mode
  - TN3270 Listen-Point PU Configuration Mode
- TN3270 PU Configuration Mode
- TN3270 Response-Time Configuration Mode
- TN3270 Security Configuration Mode
  - TN3270 Security Profile Configuration Mode

For details, refer to the "Configuring the TN3270 Server" chapter in the "IBM Networking" part of the Release 12.2 *Cisco IOS Bridging and IBM Networking Configuration Guide* and the "TN3270 Server Commands" chapter of the Release 12.2 *Cisco IOS Bridging and IBM Networking Command Reference, Volume 2*.

TN3270 DLUR Configuration Mode

Prompt: (tn3270-dlur)

To enter dependent LU requester (DLUR) configuration mode from TN3270 server configuration mode, use the **dlur** command. Use DLUR configuration mode to enable the SNA session switch function on a CMCC adapter.

For details, refer to the "Configuring the TN3270 Server" chapter in the "IBM Networking" part of the Release 12.2 *Cisco IOS Bridging and IBM Networking Configuration Guide*.

TN3270 DLUR PU Configuration Mode

Prompt: (tn3270-dlur-pu)

To enter dependent LU requester (DLUR) PU configuration mode from DLUR configuration mode, use the **pu (DLUR)** command. Use DLUR PU configuration mode to create a PU entity that has no direct link to a host.

Note DLUR PU configuration mode is a legacy configuration mode whose function to define DLUR PUs can be replaced by using theTN3760 listen-point configuration modes in Cisco IOS Release 12.0(5)T and later. When you define listen-point configurations, you can create DLUR PUs within listen-point PU configuration mode using the **pu dlur** command instead.

For details, refer to the "Configuring the TN3270 Server" chapter in the "IBM Networking" part of the Release 12.2 *Cisco IOS Bridging and IBM Networking Configuration Guide*.

TN3270 DLUR Linked SAP Configuration Mode

Prompt: (tn3270-dlur-lsap)

To enter dependent LU requester (DLUR) linked service access point (SAP) configuration mode from DLUR configuration mode, use the **lsap** command. Use DLUR linked SAP configuration mode to create an SAP in the SNA session switch.

For details, refer to the "Configuring the TN3270 Server" chapter in the "IBM Networking" part of the Release 12.2 *Cisco IOS Bridging and IBM Networking Configuration Guide*.

TN3270 Listen-Point Configuration Mode

Prompt: (tn3270-lpoint)

To enter listen-point configuration mode from TN3270 server configuration mode, use the **listen-point** command. Use listen-point configuration mode to specify the IP address and TCP port number to create a listen point.

For details, refer to the "Configuring the TN3270 Server" chapter in the "IBM Networking" part of the Release 12.2 *Cisco IOS Bridging and IBM Networking Configuration Guide*.

TN3270 Listen-Point PU Configuration Mode

Prompt: (tn3270-lpoint-pu)

To enter listen-point PU configuration mode from listen-point configuration mode, use the **pu (listen-point)** command. Use listen-point PU configuration mode to create a PU entity that has a direct link to a host.

For details, refer to the "Configuring the TN3270 Server" chapter in the "IBM Networking" part of the Release 12.2 *Cisco IOS Bridging and IBM Networking Configuration Guide*.

TN3270 PU Configuration Mode

Prompt: (tn3270-pu)

To enter PU configuration mode from TN3270 server configuration mode, use the **pu (TN3270)** command. Use PU configuration mode to create a PU entity that has its own direct link to a host.

Note PU configuration mode is a legacy configuration mode whose function to define direct PUs can be replaced by using the listen-point configuration modes in Cisco IOS Release 12.0(5)T and later. When you define listen-point configurations, you can create direct PUs within listen-point PU configuration mode using the **pu (listen-point)** command instead.

For details, refer to the "Configuring the TN3270 Server" chapter in the "IBM Networking" part of the Release 12.2 *Cisco IOS Bridging and IBM Networking Configuration Guide*.

TN3270 Response-Time Configuration Mode

Prompt: (tn3270-resp-time)

To enter response-time configuration mode from TN3270 server configuration mode, use the **response-time group** command. Use response-time configuration mode to configure a client subnet group for response-time measurements.

For details, refer to the "Configuring the TN3270 Server" chapter in the "IBM Networking" part of the Release 12.2 *Cisco IOS Bridging and IBM Networking Configuration Guide*.

TN3270 Security Configuration Mode

Prompt ID: (tn3270-security)

To enter TN3270 security configuration mode from TN3270 server configuration mode, use the **security** command. Use security configuration mode to configure security on the TN3270 server.

For details, refer to the "Configuring the TN3270 Server" chapter in the "IBM Networking" part of the Release 12.2 *Cisco IOS Bridging and IBM Networking Configuration Guide*.

TN3270 Security Profile Configuration Mode

Prompt ID: (tn3270-sec-profile)

To enter TN3270 security profile configuration mode from TN3270 security configuration mode, use the profile command. Use profile configuration mode to configure a security profile on the TN3270 server.

For details, refer to the "Configuring the TN3270 Server" chapter in the "IBM Networking" part of the Release 12.2 *Cisco IOS Bridging and IBM Networking Configuration Guide*.

Translation-Rule Configuration Mode

Prompt: (config-translate)

To enter translation-rule configuration mode from global configuration mode, use the **translation-rule** command. Use translation-rule configuration mode to define a translation-rule tag number.

For details, refer to the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter in the Release 12.2 *Cisco IOS Voice, Video, and Fax Configuration Guide*.

Voice-Card Configuration Mode

Prompt: (config-voicecard)

To enter voice-card configuration mode from global configuration mode, use the **voice-card** command.Use voice-card configuration mode to specify the HCM codec complexity for a voice card.

For details, refer to the "Configuring Voice Ports" chapter in the Release 12.2 *Cisco IOS Voice, Video, and Fax Configuration Guide*.

Voice Class Configuration Mode

Prompt: (config-class)

To enter voice class configuration mode from global configuration mode, use one of the forms of the **voice class** command. Use the **voice class busyout** command to use voice class configuration mode to define busyout conditions to be applied to a voice port. Use the **voice class permanent** command to use voice class configuration mode to create a voice class for Cisco trunk (private line) or FRF.11 trunk calls. You can assign the voice class to network dial peers and to voice ports. Use the **voice class dualtone** command to use voice class configuration mode to define a supervisory disconnect tone or tones to be detected. You can assign the voice class to an FXO voice port.

For details, refer to the "Configuring Voice over ATM" chapter in the Release 12.2 *Cisco IOS Voice, Video, and Fax Configuration Guide*.

Voice-Port Configuration Mode

Prompt: (config-voiceport)

To enter voice port configuration mode from global configuration mode, use the **voice-port** command. Use voice port configuration mode to configure voice port settings for voice over ATM, voice over Frame Relay, and other related protocols.

For details, refer to the **voice-port** command description in the Release 12.2 *Cisco IOS Voice, Video, and Fax Command Reference*.

Voice Service Configuration Mode

Prompt: (conf-voi-serv)

To enter voice-service configuration mode from global configuration mode, use the **voice service {pots | voatm | vofr | voip}** command. Use voice-service configuration mode to specify POTS, voice over ATM (voatm), voice over Frame Relay (vofr), or Voice over IP (voip) options.

For details on specifying voice over ATM options, refer to the "Configuring Voice over ATM" chapter in the Release 12.2 *Cisco IOS Voice, Video, and Fax Configuration Guide*.

Voice Service Session Configuration Mode

Prompt: (conf-voi-serv-s)

To enter voice-service session configuration mode from voice-service configuration mode, use the **session protocol aal2** command. Use voice-service session configuration mode to configure call admission control (CAC) and subcell multiplexing.

For details, refer to the "Configuring Voice over ATM" chapter in the Release 12.2 *Cisco IOS Voice, Video, and Fax Configuration Guide*.

VoIP Dial Peer Configuration Mode

See Dial Peer Voice Configuration Mode.

VPDN Group Mode and Submodes

Prompt: (config-vpdn)

To enter virtual private dial-up network (VPDN) group configuration mode, first enable VPDN by using the **vpdn enable** global configuration mode command, and then use the **vpdn-group** *number* global configuration mode command. The VPDN group configuration mode is used to configure VPDN services on Cisco routers and access servers. In VPDN group configuration mode, you can configure generic information for the entire VPDN group. You can also enter the VPDN configuration submodes, and configure specific information for the VPDN services.

Refer to the "Configuring Virtual Private Networks" chapter in the "Virtual Templates, Profiles, and Networks" part of the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide* for information on the following VPDN group configuration submodes:

- VPDN Accept-dialin group configuration mode (config-vpdn-acc-in)
- VPDN Accept-dialout group configuration mode (config-vpdn-acc-ou)
- VPDN Request-dialin group configuration mode (config-vpdn-req-in)
- VPDN Request-dialout group configuration mode (config-vpdn-req-ou)

VPDN Profile Configuration Mode

See (Resource-Pool) VPDN Profile Configuration Mode.

VPDN Template Configuration Mode

Prompt: (config-vpdn-templ)

To enter VPDN template configuration mode from global configuration mode, use the **vpdn-template** command. Use the VPDN template configuration mode to configure a VPDN group configuration template.

For details, refer to the 12.2(4)T "Default VPDN Group Template" feature module.

VRF Configuration Mode

Prompt: (config-vrf)

Aliases: IP VPN Routing/Forwarding instance Configuration Mode; IP VRF (ip-vrf) Configuration Mode

To enter VPN routing/forwarding (VRF) configuration mode from global configuration mode or router configuration mode, use the **ip vrf** command. Use VRF configuration mode to specify attributes for an MPLS VPN routing/forwarding instance (VRF).

For details, refer to the "Configuring Multiprotocol Label Switching" chapter in the "Multiprotocol Label Switching" part of the Release 12.2 *Cisco IOS Switching Services Configuration Guide*.

Note This mode appears as the **ip-vrf** option in the **show parser dump ?** command.

X.25 Profile Configuration Mode

Prompt: (config-x25)

To enter X.25 configuration mode from global configuration mode, use the **x25 profile** command. X.25 profiles streamline X.25 and LAPB configuration. X.25 profiles can contain existing X.25 and LAPB commands and, once created and named, can be simultaneously associated with more than one DLCI connection, using just the profile name. X.25 Layers 2 and 3 are transparently supported over Annex G. LAPB treats the Frame Relay network like an X.25 network link and passes all of the data and control messages over the Frame Relay network.

For details, refer to the **x25 profile** command documentation in the Release 12.2 *Cisco IOS Wide-Area Networking Configuration Guide* and the Release 12.2 *Cisco IOS Wide-Area Networking Command Reference* for more information.

Configuration Modes Summary Table

Table 26 lists the configuration modes available using the Cisco IOS CLI. The availability of any particular mode will depend on the features in your system software image and which platform you are using. For example, some configuration modes are specifically for configuring access servers, and will not be available on most routers.

Configuration modes are listed alphabetically by router prompt. All prompts listed are shown as they appear after the host-name prompt on the system (for example, if the host-name is "Router", the prompt for CA Identity configuration mode would be `Router(ca-identity)#`). The examples in the table assume the general default host-name of "Router."

Unless otherwise indicated, the **exit** command will bring you back to the mode you were in before you entered the current mode. For example, using the **exit** command in *subinterface configuration mode* will bring you back to *interface configuration mode*, using the **exit** command in *interface configuration mode* will bring you back to *global configuration mode*, and using the **exit** command in *global configuration mode* will bring you back to *privileged EXEC mode*.

At any time you can enter the **end** command to end your configuration session and return to privileged EXEC mode.

Table 26 Configuration Mode Summaries

Prompt	Configuration Mode Name	Access Method	Example
(ca-identity)	CA Identity Configuration Mode	From global configuration mode, use the **crypto ca identity** command.	Router(config)# **crypto ca identity** Router(ca-identity)#
(ca-root)#	CA Trusted-Root Configuration Mode	From global configuration mode, use the **crypto ca trusted-root** command.	Router(config)# **crypto ca trusted-root** Router(ca-root)#
(cfg-adap-*type n-m*)	IBM Channel Internal Adapter Configuration Mode	From IBM channel internal LAN configuration mode, enter the **adapter** command. In the router prompt syntax, *type* is the specified internal LAN type, *n* is the specified lan-id, and *m* is the adapter number.	Router(config)# **lan ethernet 10** Router(cfg-lan-Ether 10)# **adapter 1 4.5.6** Router(cfg-adap-Ether 10-1)#
(cfg-atm-range-p)#	ATM PVC-in-range Configuration Mode	From PVC range configuration mode, use the **pvc-in-range** command.	Router(config-if-atm-range)# **pvc-in-range** [*pvc-name*] [*vpi*][**/***vci*] Router(cfg-if-atm-range-pvc)#
(cfg-lan-*type n*)#	IBM Channel Internal LAN Interface Configuration Mode	From interface configuration mode, use the **lan** command. In the router prompt syntax, *type* is the specified internal LAN type and *n* is the specified LAN ID.	Router(config-if)# **lan ethernet 10** Router(cfg-lan-Ether 10)#
(cfg-tn3270)#	TN3270 Server Configuration Mode	From interface configuration mode, use the **tn3270-server** command.	Router(config)# **interface** *type slot***/***port* Router(config-if)# **tn3270-server** Router(cfg-tn3270)#

Table 26 Configuration Mode Summaries (continued)

Prompt	Configuration Mode Name	Access Method	Example
`(config-access-point)`	Access-point Configuration Mode	From access-point list configuration mode, use the **access-point** command.	`Router(config-ap-list)# access-point` `Router(config-access-point)#`

`(config-alps-ascu)`	ALPS ASCU Configuration Mode	From interface configuration mode, use the **alps ascu** command.	`Router(config)# interface` *type slot/port*
			`Router(config-if)# alps ascu 4B`
			`Router(config-alps-ascu)#`

`(config-alps-circuit)`	ALPS Circuit Configuration Mode	From global configuration mode, use the **alps circuit** command.	`Router(config)# alps circuit CKT_NAME` `Router(config-alps-circuit)#`
`(config-annexg)`	Annex G Configuration Mode	From global configuration mode, use the **call-router h323-annexg** command.	`Router(config)# call-router h323-annexg be20` `Router(config-annexg)# advertise all`
`(config-ap-list)`	Access-point List Configuration Mode	From global configuration mode, use the **gprs access-point-list** command.	`Router(config)# gprs access-point-list` `Router(config-ap-list)#`

`(config-atm-bundle)`
or
`(atm-bundle-config)`
	ATM VC Bundle Configuration Mode	From interface or subinterface configuration mode, use the **bundle** command.	`Router(config-subif)# bundle newyork`
			`Router(config-atm-bundle)#`

`(config-call-discriminator)`	(Resource-Pool) Call Discriminator Profile Configuration Mode	From global configuration mode, use the **resource-pool profile discriminator** command.	`Router(config)# resource-pool profile discriminator profile1`
			`Router(config-call-discrimin)# ?`
			`Call Discriminator Profile Commands:`
			`  call-type  Call-type to be rejected`
			`  clid       CLID entity to be rejected`
			`  dnis       DNIS entity to be rejected`

`(config-casa)#`	CASA Configuration Mode	From global configuration mode, use the **ip casa** command.	`Router(config)# ip casa 10.10.4.1 224.0.1.2` `Router(config-casa)#`
`(config-cert-chain)#`	Certificate Chain Configuration Mode	From global configuration mode, use the **crypto ca certificate** chain command.	`Router(config)# crypto ca certificate` `Router(config-cert-chain)#`

Table 26 Configuration Mode Summaries (continued)

Prompt	Configuration Mode Name	Access Method	Example
`(config-ces)#`	CES Configuration Mode	From global configuration mode, use the **ces** command.	`Router(config)# ces 1/0` `Router(config-ces)#`

`(config-class)#`	Voice Class Configuration Mode	From global configuration mode, use the **voice class** command.	`Router(config)# voice class busyout bsyout1`
			`Router(config-class)# ?`
			`voiceclass configuration commands:`
			`busyout Configure busyout trigger event & procedure`

`(config-cmap)#`	QoS Class-Map Configuration Mode	From global configuration mode, use the **class-map** command.	`Router(config)# class-map` `Router(config-cmap)#`
`(config-controller)#`	Controller Configuration Mode	From global configuration mode, use the **controller** command.	`Router(config)# controller t1 0/0` `Router(config-controll)#`
`(config-cor)#`	Dial Peer COR List Configuration Mode	From global configuration mode, use the **dial-peer cor list** *list-name* command.	`Router(config)# dial-peer cor list corlist1` `Router(config-cor)#`
`(config-crypto-map)#`	Crypto Map Configuration Mode	From global configuration mode, use the **crypto map** command.	`Router(config)# crypto map Research 10` `Router(config-crypto-map)#`
`(config-crypto-trans)#`	Crypto Transform Configuration Mode	From global configuration mode, use the **crypto ipsec transform-set** command.	`Router(config)#crypto ipsec transform-set` `Router(config-crypto-trans)#`
`(config-ctrl-cas)#`	CAS Custom Configuration Mode	From controller configuration mode, use the **cas-custom** command.	`Router(config-controller)# cas-custom 1` `Router(config-ctrl-cas)#`

`(config-customer-profile)#`	(Resource-Pool) Customer Profile Configuration Mode	From global configuration mode, use the **resource-pool profile customer** command.	`Router(config)# resource-pool profile customer name1`
			`Router(config-customer-profi)#?`
			`Customer Profile Configuration Commands:`
			`dnis Assign DNIS group with this profile`
			`limit Configure limits for the profile`
			`resource Assign resource and supported call-type`
			`source Assign Template with this profile`
			`vpdn Assign VPDN group/profile with this profile`

Table 26 Configuration Mode Summaries (continued)

Prompt	Configuration Mode Name	Access Method	Example
`(config-dhcp)#`	DHCP Pool Configuration Mode	From global configuration mode, use the **ip dhcp pool** command.	`Router(config)# ip dhcp pool pname1` `Router(config-dhcp)#`
`(config-dialpeer)#`	Dial Peer Voice Configuration Mode	From global configuration mode, use the **dial peer voice** command.	`Router(config)# dial peer voice 1 pots` `Router(config-dialpeer)#`

`(config-dnis-group)#`	Dialer DNIS Group Configuration Mode	From global configuration mode, use the **dialer dnis group** command.	`Router(config)# dialer dnis group dnis_isp_1`
			`Router(config-dnis-group)# ?`
			`Dialer DNIS Configuration Commands:`
			`call-type set call-type override`
			`number Enter number in DNIS group`
			`range Enter a range of numbers in DNIS group`

`(config-ext-nacl)#`	Extended Named Access List (NACL) Configuration Mode	From global configuration mode, use the **ip access-list** or **ipx access-list** command.	`Router(config)# ip access-list extended flag` `Router(config-ext-nacl)#`
`(config-fr-congest)#`	Frame Relay Congestion Management Configuration Mode	From interface configuration mode, use the **frame-relay congestion-management** command.	`Router(config-if)# frame-relay congestion-management` `Router(config-fr-congest)#`

`(config-fr-dlci)#`	Frame Relay DLCI Configuration Mode	From interface configuration mode, use the **frame-relay interface-dlci [switched]** command.	`Router(config)# interface serial 1/1`
			`Router(config-if)# frame-relay interface-dlci 100`
			`Router(config-fr-dlci)# vofr`
			`Router(config-fr-dlci)#`

`(config-frf5)#`
or
`(config-frf8)#`
	FRF.5 / FRF.8 Configuration Mode	From global configuration mode, use the **connect** command.	`router(config)# connect serial0 100 atm3/0 0/32 network-interworking`
			`router(config-frf5)# clp-bit 1`
			or
			`router(config)# connect serial0 100 atm1/0 0/32 service-interworking`
			`router(config-frf8)# efci-bit map-fecn`

`(config-gateway)#`	Gateway Configuration Mode	From global configuration mode, use the **gateway** command.	`Router(config )# gateway` `Router(config-gateway)#`
`(config-gk)#`	Gatekeeper Configuration Mode	From global configuration mode, use the **gatekeeper** command.	`Router(config)# gatekeeper` `Router(config-gk)#`

Table 26 Configuration Mode Summaries (continued)

Prompt	Configuration Mode Name	Access Method	Example
`(config-hub)#`	Hub Configuration Mode	From global configuration mode, use **hub** command.	`Router(config)#hub ethernet 0 1 3` `Router(config-hub)#`
`(config-if)#`	Interface Configuration Mode	From global configuration mode, enter by specifying an interface with an **interface** command.	`Router(config)# interface serial 2` `Router(config-if)#`

`(config-if-atm-member)#`	ATM VC Bundle-Member Configuration Mode	From ATM bundle configuration mode, use the **pvc-bundle** command.	`Router(config-if)# bundle chicago`
			`Router(config-if-atm-bundle)# pvc-bundle chicago-control 207`
			`Router(config-if-atm-member)# class control-class`
			`Router(config-if-atm-bundle)# pvc-bundle chicago-premium 206`

`(config-if-atm-range-pvc)#`	ATM PVC Range Configuration Mode	From subinterface configuration mode, use the **range** [*name*] **pvc** command.	`Router(config-subif)# range [range-name] pvc start-vpi/start-vci end-vpi/end-vci` `Router(config-if-atm-range-p)#`

`(config-if-atm-vc)#`	ATM VC Configuration Mode	From interface configuration mode, use the **pvc** or **svc nsap** command.	`Router(config-if)# pvc 0/33`
			`Router(config-if-atm-vc)#`
			
			or
			`Router(config-if)# svc nsap AB.CDEF.01.234567.890A.BCDE.F012.3456.7890.1234.12`
			`Router(config-if-atm-vc)#`

`(config-if-ces-vc)#`	ATM VC CES Configuration Mode	From interface configuration mode, use the **pvc** or **svc** command with the **ces** keyword, or the **ces pvc** command.	`Router(config-if)# svc [name] ces` `Router(config-if-ces-vc)#`

`(config-if-path)#`	IP Host Backup Configuration Mode	From interface configuration mode, use the **path** command.	`Router(config)# interface channel 3/1`
			`Router(config-if)# ip address 198.92.5.1 255.255.255.128`
			`Router(config-if)# path c010 c110 c210`
			`Router(config-if-path)# claw 30 198.92.5.2 lpar1 cip1 tcpip tcpip`
			`. . .`

Table 26 Configuration Mode Summaries (continued)

Prompt	Configuration Mode Name	Access Method	Example

`(config-ipv6-acl)#`	IPv6 Access List Configuration Mode	From global configuration mode, use the **ipv6 access-list** command.	`Router(config)# time-range lunchtime`
			`Router(config)# periodic weekdays 12:00 to 13:00`
			`Router(config)# ipv6 access-list OUTBOUND`
			`Router(config-ipv6-acl)# permit tcp any any eq www time-range lunchtime`
			`Router(config-ipv6-acl)# deny tcp any any eq www log-input`
			`Router(config-ipv6-acl)# permit tcp 2000:1::/64 any`
			`Router(config-ipv6-acl)# permit udp 2000:1::/64 any`

`(config-ipx-router)#`	IPX Router Configuration Mode	From global configuration mode, use the **ipx router** command.
		(IPX must first be enabled using the **ipx routing** command.)
			`Router(config)# ipx routing`
			`Router(config)# ipx router rip`
			`Router(config-ipx-router)#`

`(config-isakmp)#`	ISAKMP Policy Configuration Mode	From global configuration mode, use the **crypto isakmp policy** command.	`Router(config)# crypto isakmp policy` `Router(config-isakmp)#`
`(config-keychain)#`	Key-Chain Configuration Mode	From global configuration mode, use the **keychain** command.	`Router(config)# keychain blue` `Router(config-keychain)#`
`(config-keychain-key)#`	Key-Chain Key Configuration Mode	From keychain configuration mode, use the **key** command.	`Router(config-keychain)# key 10` `Router(config-keychain-key)#`
`(config-line)#`	Line Configuration Mode	From global configuration mode, enter by specifying a line with a **line** {**aux** | **con** | **tty** | **vty**} *line-number* [*ending-line-number*] command.	`Router(config)# line vty 0 4` `Router(config-line)#`

`(config-map-class)#`	Static Maps Class Configuration Mode	From global configuration mode, use the **map-class** *encapsulation class-name* command.	`Router(config)# map-class frame-relay map1`
			`Router(config-map-class)# ?`
			`Static maps class configuration commands:`
			`  frame-relay  Configure Map parameters`
			`  service-policy  class-based service policy`

Table 26 Configuration Mode Summaries (continued)

Prompt	Configuration Mode Name	Access Method	Example

`(config-map-list)#`	Static Maps List Configuration Mode	From global configuration mode, use the **map-list** *name* command.	`Router(config)# map-list map2`
			`Router(config-map-list)# ?`
			`Static maps list configuration commands:`
			`A.B.C.D Protocol specific address`
			`aarp AppleTalk ARP`
			`apollo Apollo Domain`
			`appletalk AppleTalk`
			`. . .`

`(config-modem-pool)#`	Modem Pool Configuration Mode	From global configuration mode, use the **modem-pool** *name* command.	`Router(config)# modem-pool pool1`
			`Router(config-modem-pool)# ?`
			`Modem pool configuration commands:`
			`called-number Map a called number to modem pool`
			`pool-range Configure a group range for the modem pool`

`(config-mpoa-client)`
 	
`See (mpoa-client-config)#`
	See MPOA Client configuration mode (below)		

`(config-mpoa-server)`
 	
`See (mpoa-server-config)#`
	See MPOA Server configuration mode (below)		

`(config-mrm-manager)#`	MRM Manager Configuration Mode	From global configuration mode, use the **ip mrm manager** command.	`Router(config)# ip mrm manager test1` `Router(config-mrm-manager)#`
`(config-pmap)#`	QoS Policy-Map Configuration Mode	From global configuration mode, use the **policy-map** command.	`Router(config)# policy-map policyA` `Router(config-pmap)#`

`(config-pmap-c)#`	QoS Policy-Map Class Configuration Mode	From policy-map configuration mode, use the **class** command.	`Router(config)# policy-map policyA`
			`Router(config-pmap)# class first`
			`Router(config-pmap-c)#`

`(config-poll-group)#`	System Controller Poll-Group Configuration Mode	From global configuration mode, enter poll-group configuration mode with the **syscon poll-group** command.	`Router(config)# syscon poll-group cmlineinfo` `Router(config-poll-gr)#`
`(config-preauth)#`	AAA Preauthentication Configuration Mode	From global configuration mode, use the **aaa preauth** command.	`Router(config)# aaa preauth` `Router(config-preauth)#`

Table 26 Configuration Mode Summaries (continued)

Prompt	Configuration Mode Name	Access Method	Example
`(config-pubkey-chain)#`	Public-Key Chain Configuration Mode	From global configuration mode, use the **crypto key pubkey-chain {dss | rsa}** command.	`Router(config)# crypto key pubkey-chain rsa` `Router(config-pubkey-c)#`

`(config-pubkey-hex)#`	Public-Key Hex Input Configuration Mode	From public-key key configuration mode, use the **key-string** command.	`Router(config-pubkey-key)# address 10.5.5.1`
			`Router(config-pubkey-key)# key-string 005C300D 06092A86`
			`Router(config-pubkey-hex)# 4886F70D 01010105`
			`. . .`

`(config-pubkey-key)#`	Public-Key Key Configuration Mode	From public-key chain configuration mode, use the **addressed-key** command or **named-key** command.	`Router(config-pubkey-c)# named-key otherpeer.domain.com` `Router(config-pubkey-k)#`
`(config-red-group)#`	RED Group Configuration Mode	From global configuration mode, use the **random-detect-group** command.	`Router(config)# random-detect-group sanjose` `Router(config-red-group)#`
`(config-resource-group)#`	(Resource-Pool) Resource Group Configuration Mode	From global configuration mode, use the **resource-pool group resource** command.	`Router(config)# resource-pool group resource groupname1` `Router(config-resource-group)# range limit 48`
`(config-rlm-group)#`	RLM Group Configuration Mode	From interface configuration mode, use the **rlm group** command.	`Router(config-if)# rlm group 1` `Router(config-rlm-group)#`
`(config-rlm-group-sc)#`	RLM Device Configuration Mode	From RLM group configuration mode, use the **server** command.	`Router(config-rlm-group)# server r1-server` `Router(config-rlm-group-sc)#`

`(config-route-map)#`	Route-Map Configuration Mode	From global configuration mode, use the **route-map** command.	`Router(config)# route-map arizona`
			`Router(config-route-map)# ?`
			`Route Map configuration commands:`
			`  match  Match values from routing table`
			`  set  Set values in destination routing protocol`

Table 26 Configuration Mode Summaries (continued)

Prompt	Configuration Mode Name	Access Method	Example
`(config-router)#`	Router Configuration Mode	From global configuration mode, enter by issuing the **router** *protocol* command (such as **router igrp**).	`Router(config)# router rip` `Router(config-router)#`

`(config-router-af)#`	Address Family Configuration Mode	From router configuration mode, use the **address-family** command.
		To exit, use the **exit-address-family** command.
			`Router(config)# router bgp 100`
			`Router(config-router)# address-family vpnv4`
			`Router(config-router-af)#`

`(config-rtr)#`	RTR Entry Configuration Mode	From global configuration mode, use the **rtr** command.	`Router(config)# rtr 1`
			`Router(config-rtr)# ?`
			`  RTR Entry Commands:`
			`. . .`

`(config-rtr-http)#`	SAA HTTP Raw Request Configuration Mode	From RTR configuration mode, use the **http-raw-request** command.	`Router(config-rtr)# type http operation raw url http://www.cisco.com`
			`Router(config-rtr)# http-raw-request`
			`Router(config-rtr-http)# ?`
			`HTTP Raw Request Configuration:`
			`  LINE  http raw request; enter 'exit' to end the request`
			`Router(config-rtr-http)# GET /index.html HTTP/1.0\r\n`
			`Router(config-rtr-http)# \r\n`
			`Router(config-rtr-http)# exit`
			`Router(config-rtr)#`

`(config-service-profile)#`	Service Profile Configuration Mode	From global configuration mode, use the **resource-pool profile service** command.	`Router(config)# resource-pool profile service user1`
			`Router(config-service-profil)# ?`
			`Service Profile Configuration Commands:`
			`  modem  Configure modem service parameters`

`(config-sg)#`
 	
`or`
`(config-sg-radius)#`
	Server Group RADIUS Configuration Mode	From global configuration mode or interface configuration mode, use the **aaa group server radius** command.	`Router(config-if)# aaa group server radius sg1`
			`Router(config-sg-radius)#`

Table 26 Configuration Mode Summaries (continued)

Prompt	Configuration Mode Name	Access Method	Example
`(config-sg)#` or `(config-sg-tacacs)#`	Server Group TACACS+ Configuration Mode	From global configuration mode or interface configuration mode, use the **aaa group server tacacs+** command.	`Router(config-if)# aaa group server tacacs+ sg1` `Router(config-sg-tacacs)#`
`(config-slb-dfp)#`	SLB DFP Configuration Mode	From global configuration mode, use the **ip slb dfp** command.	`Router(config)# ip slb dfp` `Router(config-slb-dfp)#`

`(config-slb-real)#`	SLB Real Server Configuration Mode	From server farm configuration mode, use the **real** command.	`Router(config)# ip slb serverfarm sfarm1`
			`Router(config-slb-sfarm)# real ip-address`
			`Router(config-slb-real)#`

`(config-slb-sfarm)#`	SLB Server-Farm Configuration Mode	From global configuration mode, use the **ip slb serverfarm** command.	`Router(config)# ip slb serverfarm sfarm1` `Router(config-slb-sfarm)#`
`(config-slb-vserver)#`	SLB Virtual Server Configuration Mode	From global configuration mode, use the **ip slb vserver** command.	`Router(config)# ip slb vserver vserver1` `Router(config-slb-vserver)#`

`(config-spe)`	SPE Configuration Mode	From global configuration mode, use the **spe** command.	`Router(config)# spe 1/0 1/23`
			`Router(config-spe)# firmware location flash:mcom-modem-code.5.2.30.bin`
			`Router(config-spe)#`

`(config-std-nacl)#`	Standard Named Access List (NACL) Configuration Mode	From global configuration mode, use the **ip access-list** or **ipx access-list** command.	`Router(config)# ip access-list standard Internetfilter`
			`Router(config-std-nacl)# permit 192.5.34.0 0.0.0.255`
			`Router(config-std-nacl)# deny 128.88.0.0 0.0.255.255`
			`Router(config-std-nacl)# exit`
			`Router(config)#`

`(config-subif)#`	Subinterface Configuration Mode	From interface configuration mode, specify a subinterface with an **interface** command.	`Router(config-if)# interface serial 2.1` `Router(config-subif)#`
`(config-time-range)#`	Time Range Configuration Mode	From global configuration mode, use the **time-range** *time-range-name* command.	`Router(config)# time-range no-http` `Router(config-time-range)#`

Table 26 Configuration Mode Summaries (continued)

Prompt	Configuration Mode Name	Access Method	Example
`(config-translate)#`	Translation-Rule Configuration Mode	From global configuration mode, use the **translation-rule** command.	`Router(config)# translation-rule 10` `Router(config-translate)#`
`(config-vc-class)#`	ATM VC Class Configuration Mode	From interface configuration mode or subinterface configuration mode, use the **vc-class atm** command.	`Router(config-if)# vc-class atm pvc1` `Router(config-vc-class)#`

`(config-vc-group)#`	ATM-FR VC Group Configuration Mode	From global configuration mode, use the **vc-group** command.	`router(config)# vc-group friends`
			`router(config-vc-group)# serial0 16 16`
			`router(config-vc-group)# serial0 17 17`

`(config-voiceport)#`	Voice-Port Configuration Mode	From global configuration mode, use the **voice port** *slot/*[*sub-unit/*]*port* command.	`Router(config)# voice port 1/1/2` `Router(config-voiceport)#`
`(config-vpdn)#`	VPDN Group Configuration Mode	From global configuration mode, use the **vpdn-group** *number* command.	`Router(config)# vpdn-group 1` `Router(config-vpdn)#`
`(config-vpdn-acc-in)#`	VPDN Accept-dialin Configuration Mode	From VPDN group mode, use the **accept-dialin** command.	`Router(config-vpdn)# accept-dialin` `Router(config-vpdn-acc-in)#`
`(config-vpdn-acc-out)#`	VPDN Accept-dialout Configuration Mode	From VPDN group mode, use the **accept-dialout** command.	`Router(config-vpdn)# accept-dialout` `Router(config-vpdn-acc-ou)#`
`(config-vpdn-profile)#`	(Resource-Pool) VPDN Profile Configuration Mode	From global configuration mode, use the **resource-pool profile vpdn** command.	`Router(config)# resource-pool profile vpdn profile2` `Router(config-vpdn-pro)#`
`(config-vpdn-req-in)#`	VPDN Request-dialin Configuration Mode	From VPDN group mode, use the **request-dialin** command.	`Router(config-vpdn)# request-dialin` `Router(config-vpdn-req-in)#`

Table 26 Configuration Mode Summaries (continued)

Prompt	Configuration Mode Name	Access Method	Example
`(config-vpdn-req-ou)#`	VPDN Request-dialout Configuration Mode	From VPDN group mode, use the **request-dialout** command.	`Router(config-vpdn)# request-dialout` `Router(config-vpdn-req-ou)#`
`(config-vpdn-templ)`	VPDN Template Configuration Modee	From global configuration mode, use the **vpdn-template** command.	`R(config)# vpdn-template` `R(config-vpdn-templ)#?`

`(config-vrf)#`	IP VPN Routing/Forwarding (VRF) Instance Configuration Mode
	(a.k.a. VRF Configuration Mode)
		From global configuration mode or router configuration mode, use the **ip vrf** command.	`Router(config)#ip vrf name`
			`Rotuer(config-vrf)#?`
			`IP VPN Routing/Forwarding instance configuration commands:`
			`  bgp  Commands pertaining to BGP`
			`. . .`
			`  export  VRF export`
			`  import  VRF import`
			`  maximum  Set a limit`
			`. . .`
			`  rd  Specify Route Distinguisher`
			`. . .`

`(config-x25)#`	X.25 Profile Configuration Mode	From global configuration mode, use the **x25 profile** command.	`Router(config)# x25 profile NetworkNodeA dce` `Router(config-x25)# x25 htc 128`

`(conf-voi-serv)#`	Voice Service Configuration Mode	From global mode, use the **voice service** command.	`Router(config)# voice service voatm`
			`Router(conf-voi-serv)# ?`
			`voice service configuration commands:`
			`  h323  Global H.323 commands`
			`  modem  Global modem commands`
			`  session  Voice session Protocol`

`(conf-voi-serv-s)#`	Voice Service Session Configuration Mode	From voice service configuration mode, use the **session protocol aal2** command.	`Router(config)# voice service voatm`
			`Router(conf-voi-serv)# session protocol aal2`
			`Router(conf-voi-serv-s)#`

`(lane-config-datab)#`	LANE Database Configuration Mode	From global configuration mode, use the **lane database** command.	`Router(config)# lane database red` `Router(lane-config-datab)#`
`(mpoa-client-config)#`	MPOA Client (MPC) configuration mode	From global configuration mode, use the **mpoa client config name** command.	`Router(config)# mpoa client config name ip_mpc` `Router(mpoa-client-config)#`

Table 26 Configuration Mode Summaries (continued)

Prompt	Configuration Mode Name	Access Method	Example
`(mpoa-server-config)#`	MPOA Server (MPS) configuration	From global configuration mode, use the **mpoa server config name** command.	`Router(config)# mpoa server config name ip_mps` `Router(mpoa-server-config)#`

`(tn3270-dlur)#`	TN3270 DLUR Configuration Mode	From TN3270 server configuration mode, use the **dlur** command.	`Router(config)# interface Channel3/2`
			`Router(config-if)# tn3270-server`
			`Router(cfg-tn3270)# dlur fq-cpname` *fq-dlusname*
			`Router(tn3270-dlur)#`

`(tn3270-dlur-lsap)#`	TN3270 DLUR Linked SAP Configuration Mode	From TN32370 DLUR configuration mode, use the **lsap** command.	`Router(config)# interface Channel3/2`
			`Router(config-if)# tn3270-server`
			`Router(cfg-tn3270)# dlur NETA.SHEK NETA.MVSD`
			`Router(tn3270-dlur)# lsap token-adapter 15 04`
			`Router(tn3270-dlur-lsap)#`

`(tn3270-dlur-pu)#`	TN3270 DLUR PU Configuration Mode	From DLUR configuration mode, use the **pu** (DLUR) command.	`Router(config)# interface Channel3/2`
			`Router(config-if)# tn3270-server`
			`Router(cfg-tn3270)# dlur NETA.SHEK NETA.MVSD`
			`Router(tn3270-dlur)# pu P0 05D99001 192.195.80.40`
			`Router(tn3270-dlur-pu)#`

`(tn3270-lpoint)#`	TN3270 Listen-Point Configuration Mode	From TN3270 server configuration mode, use the **listen-point** command.	`Router(config)# interface Channel3/2`
			`Router(config-if)# tn3270-server`
			`Router(cfg-tn3270)# listen-point 172.18.4.19 tcp-port 2023`
			`Router(tn3270-lpoint)#`

`(tn3270-lpoint-pu)#`	TN3270 Listen-Point PU Configuration Mode	From TN3270 listen-point configuration mode, use the **pu (listen-point)** command.	`Router(tn3270-lpoint)# pu PU1 94223456 tok 1 08`
			`Router(tn3270-lpoint-pu)#`
			or
			`Router(tn3270-lpoint)# pu P0 05D99001 dlur`
			`Router(tn3270-lpoint-pu)#`

`(tn3270-pu)#`	TN3270 PU Configuration Mode	From TN3270 server configuration mode, use the **pu (tn3270)** command.	`Router(config)# interface Channel3/2`
			`Router(config-if)# tn3270-server`
			`Router(cfg-tn3270)# pu PU1 05d00001 10.0.0.1 token-adapter 1 8 rmac 4000.0000.0001 rsap 4`
			`Router(tn3270-pu)#`

`(tn3270-resp-time)#`	TN3270 Response-Time Configuration Mode	From TN3270 server configuration mode, use the **response-time group** command.	`Router(config)# interface Channel3/2`
			`Router(config-if)# tn3270-server`
			`Router(cfg-tn3270)# response-time group MYSUBNET bucket boundaries 15 25 60 120 multiplier 35`
			`Router(tn3270-resp-time)#`

Table 26 Configuration Mode Summaries (continued)

Prompt	Configuration Mode Name	Access Method	Example

`(tn3270-sec-profile)#`	TN3270 Security Profile Configuration Mode	From TN3270 security configuration mode, use the **profile** command.	`Router(config)# interface Channel3/2`
			`Router(config-if)# tn3270-server`
			`Router(cfg-tn3270)# security`
			`Router(tn3270-security)# profile CERT40 SSL`
			`Router(tn3270-sec-profile)# servercert slot0:verisign187.pem`

`(tn3270-security)#`	TN3270 Security Configuration Mode	From TN3270 server configuration mode, use the **security** command.	`Router(config)# interface Channel3/2`
			`Router(config-if)# tn3270-server`
			`Router(cfg-tn3270)# security`

Configuring Line Cards on the Cisco 7500 Series

This appendix describes software configuration commands needed to configure line cards with certain Cisco IOS software features for Cisco 7500 series routers.

Note On the Cisco 7507 and Cisco 7513 routers, you can install two Route Switch Processor (RSP) cards in a single router to improve system availability. This feature was introduced in Cisco IOS Release 11.1(4) as the "High System Availability (HSA)" feature. Because High Availability (HA) has since come to apply to a variety of Cisco IOS hardware and software features that allow for 99.9999% uptime for Cisco devices, this feature is now referred to as the "Dual RSP" feature.

Note Boot ROM revision 11.1(2) or higher is required for HSA to work with an RSP2 line card.

The boot ROM is on a SIMM on the RSP2 and cannot be upgraded. You can identify the boot ROM version on your RSP2 by issuing the **show version | begin ROM** command in privileged EXEC mode.

Performing a Single Line Card Reload

The **service single-slot-reload-enable** global configuration command allows you to enable the Single Line Card Reload feature, a High Availability (HA) feature for Cisco 7500 series routers. When this feature is enabled, if a single line card crashes, only the line card that failed is reloaded. The physical lines and the routing protocols on the other line cards remain active (note that some packets may be dropped while the card reloads, but only packets that depend on the crashed card will be affected).

A single line card reload is substantially faster than the Cbus Complex process used in some early Cisco IOS releases.

The Cisco 7500 Single Line Card Reload feature works on all RSP images.

Note The Single Line Card Reload feature is disabled by default. Enabling this feature is highly recommended.

Configuring Dual RSPs on Cisco 7500 Series Routers

To configure Dual RSP operation, you must have a Cisco 7507 or Cisco 7513 router containing two RSP processor cards. For Dual RSP compatibility, download a Cisco IOS software subset image that has a "v" in it. For example, rsp-jv-mz, rsp-ajv-mz, and rsp-pv-mz are all Dual RSP-compatible Cisco IOS subset images.

Two RSP cards in a router provide the most basic level of increased system availability through a "cold restart" feature. A "cold restart" means that when one RSP card fails, the other RSP card reboots the router. In this way, your router is never in a failed state for very long, thereby increasing system availability.

When one RSP card takes over operation from another, system operation is interrupted. This change is similar to issuing the **reload** EXEC command. The following events occur when one RSP card fails and the other takes over:

- The router stops passing traffic.
- Route information is lost.
- All connections are lost.
- The backup or "slave" RSP card becomes the active or "master" RSP card that reboots and runs the router. Thus, the slave has its own image and configuration file so that it can act as a single processor.

Note Having Dual RSPs does not impact performance in terms of packets per second or overall bandwidth. The Dual RSP feature does not provide fault-tolerance or redundancy.

Understanding Master and Slave Operation

A router configured for Dual RSP operation has one RSP card that is the master and one that is the slave. The master RSP card functions as if it were a single processor, controlling all functions of the router. The slave RSP card does nothing but actively monitor the master for failure.

A system crash can cause the master RSP to fail or go into a nonfunctional state. When the slave RSP detects a nonfunctional master, the slave resets itself and takes part in *master-slave arbitration*. Master-slave arbitration is a ROM monitor process that determines which RSP card is the master and which is the slave upon startup (or reboot).

If a system crash causes the master RSP to fail, the slave RSP becomes the new master RSP and uses its own system image and configuration file to reboot the router. The failed RSP card now becomes the slave. The failure state of the slave (formerly the master) can be accessed from the console via the **show stacks** EXEC command.

With Dual RSP operation, the following items are important to note:

- An RSP card that acts as the slave runs a different software version than it does when it acts as the master. The slave mode software is a subset of the master mode software.
- The two RSP cards need not run the same master software image and configuration file. When the slave reboots the system and becomes the new master, it uses its own system image and configuration file to reboot the router.
- When enabled, automatic synchronization mode automatically ensures that the master and slave RSP card have the same configuration file.

- Both hardware and software failures can cause the master RSP to enter a nonfunctional state, but the system does not indicate the type of failure.
- The console is always connected to master. A Y cable is shipped with your Cisco 7507 or Cisco 7513 router. The "top" of the Y cable plugs into the console port on each RSP card, and the "bottom" of the Y cable plugs into a terminal or terminal server. The master RSP card has ownership of the Y cable in that the slave Universal Asynchronous Receiver Transmitter (UART) drivers are disabled. Thus, no matter which RSP card is configured as the master, your view of the internetwork environment is always from the master's perspective. Refer to your product's hardware installation and maintenance publication for information on properly installing the Y cable.

Understanding Dual RSP Implementation Methods

There are two common ways to use the Dual RSP feature, as follows:

- Simple hardware backup. Use this method to protect against an RSP card failure. With this method, you configure both RSP cards with the same software image and configuration information. Also, you configure the router to automatically synchronize configuration information on both cards when changes occur.
- Software error protection. Use this method to protect against critical Cisco IOS software errors in a particular release. With this method, you configure the RSP cards with different software images, but with the same configuration information. If you are using new or experimental Cisco IOS software, consider using the software error protection method.

You can also use Dual RSPs for advanced implementations. For example, you can configure the RSP cards with the following implementations:

- Same software images but different configuration files
- Different software images and different configuration files
- Widely varied configuration files (for example, various features or interfaces can be turned off and on per card)

Note Although other uses are possible, the configuration information in this guide describes commands for only the two common methods—simple hardware backup and software error protection.

Dual RSP Configuration Task List

To configure Dual RSP operation, perform the tasks described in the following sections. The first two and last two tasks are required for both implementations. The third and fourth tasks relates to simple hardware backup. The fifth task relates to software error protection only.

- Specifying the Default Slave RSP (both implementations)
- Ensuring That Both RSP Cards Contain the Same Configuration File (both implementations)
- Ensuring That Both RSP Cards Contain the Same System Image (simple hardware backup only)
- Ensuring That Both RSP Cards Contain the Same Microcode Image (simple hardware backup only)
- Specifying Different Startup Images for the Master and Slave RSPs (software error protection only)
- Setting Environment Variables on the Master and Slave RSP (both implementations)
- Monitoring and Maintaining Dual RSP Operation (both implementations)

Specifying the Default Slave RSP

To specify the default slave RSP card, use the following command in global configuration mode:

Command	Purpose
`Router(config)# slave default-slot processor-slot-number`	Specifies the slave RSP card.

After specifying the default slave card, save the running configuration to the startup configuration using the **copy running-config startup-config** or **copy system:running-config nvram:startup-config** EXEC command. When the system is rebooted, the RSP specification will take effect (if both RSP cards are operational): The specified default slave becomes the slave RSP card and the other RSP card takes over as the master RSP card.

The router uses the default slave information when booting as follows:

- If a system boot is due to powering up the router or using the **reload** EXEC command, then the specified default slave will be the slave RSP.
- If a system boot is due to a system crash or hardware failure, then the system ignores the default slave designation and makes the crashed or faulty RSP the slave RSP.

If you do not specifically define the default slave RSP, the RSP card located in the higher number processor slot is the default slave. On the Cisco 7507 router, processor slot 3 contains the default slave RSP. On the Cisco 7513 router, processor slot 7 contains the default slave RSP.

The following example sets the default slave RSP to processor slot 2 on a Cisco 7507 router:

```
Router# configure terminal
Router (config)# slave default-slot 2
Router (config)# end
Router# copy system:running-config nvram:startup-config
```

Ensuring That Both RSP Cards Contain the Same Configuration File

With both the simple hardware backup and software error protection implementation methods, you always want your master and slave configuration files to match. To ensure that they match, turn on automatic synchronization. In automatic synchronization mode, the master copies its startup configuration to the slave's startup configuration when you issue a **copy** EXEC command that specifies the master's startup configuration (**nvram:startup-config**) as the target.

Automatic synchronization mode is on by default; in the event that you need to reenable the automatic synchronization, use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router# slave auto-sync config`	Reenables automatic synchronization mode.
Step 3	`Router# end`	Exits configuration mode.

Step 4	`Router(config)# copy system:running-config nvram:startup-config`
	or
	`Router(config)# copy running-config startup-config`
		Saves this information to the system startup configuration and copies the configuration to the slave's startup configuration.

Ensuring That Both RSP Cards Contain the Same System Image

For simple hardware backup, ensure that both RSP cards have the same system image.

To ensure that both RSP cards have the same system image, use the following commands in EXEC mode:

	Command	Purpose
Step 1	`Router# show bootvar`	Displays the contents of the BOOT environment variable to learn the current booting parameters for the master and slave RSP.
Step 2	`Router# dir {bootflash: | slot0: | slot1:}`	Verifies the location and version of the master RSP software image.
Step 3	`Router# dir {slavebootflash: | slaveslot0: | slaveslot1:}`	Determines if the slave RSP contains the same software image in the same location.
Step 4	`Router# copy {bootflash:[filename] | slot0:[filename] | slot1:[filename]}{slavebootflash:[filename] | slaveslot0:[filename] | slaveslot1:[filename]}`	If the slave RSP does not contain the same system image in the same location, copies the master's system image to the appropriate slave location. Note that you may also need to use the **delete** or **squeeze** EXEC command in conjunction with the **copy** command to accomplish this step.

The following example shows the process of ensuring that both RSP cards have the same system image. Note that because no environment variables are set, the default environment variables are in effect for both the master and slave RSP. Therefore, the router will boot the image in slot 0.

```
Router# show bootvar

BOOT variable =
CONFIG_FILE variable =
Current CONFIG_FILE variable =
BOOTLDR variable does not exist

Configuration register is 0x0

current slave is in slot 7
BOOT variable =
CONFIG_FILE variable =
BOOTLDR variable does not exist

Configuration register is 0x0

Router# dir slot0:
-#- -length- -----date/time------ name
1    3482498  May 4 1993 21:38:04 rsp-k-mz11.2

7993896 bytes available (1496 bytes used)

Router# dir slaveslot0:
-#- -length- -----date/time------ name
1    3482498  May 4 1993 21:38:04 rsp-k-mz11.1

7993896 bytes available (1496 bytes used)

Router# delete slaveslot0:rsp-k-mz11.1
Router# copy slot0:rsp-k-mz11.2 slaveslot0:rsp-k-mz11.2
```

Ensuring That Both RSP Cards Contain the Same Microcode Image

To ensure that interface processors will load the same microcode, regardless of which RSP is used, use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# show controller cbus`	Determines the microcode images used on the interface processors. If all interface processors are running from the bundled system microcode, no further action is required.
Step 2	`Router# dir {bootflash: | slot0: | slot1:}`	If any interface processors are running from the Flash file system, verifies the location and version of the master RSP's supplementary microcode.
Step 3	`Router# dir {slavebootflash: | slaveslot0: | slaveslot1:}`	Determines if the slave RSP contains the same microcode image in the same location.
Step 4	`Router# copy {bootflash:[filename] | slot0:[filename] | slot1:[filename]} {slavebootflash:[filename] | slaveslot0:[filename] | slaveslot1:[filename]}`	If the slave RSP does not contain the same microcode image in the same location, copies the master's microcode image to the appropriate slave location. Note that you also may need to use the **delete** or **squeeze** command in conjunction with the **copy** command to accomplish this step.

The following example ensures that both RSP cards have the same microcode image. Notice that slots 0, 1, 4, 9, and 10 load microcode from the bundled software, as noted by the statement "software loaded from system." Slot 11, the Fast Serial Interface Processor (FSIP), does not use the microcode bundled with the system. Instead, it loads the microcode from slot0:pond/bath/rsp_fsip20-1. Thus, you must ensure that the slave RSP has a copy of the same FSIP microcode in the same location.

```
Router# show controller cbus

MEMD at 40000000, 2097152 bytes (unused 416, recarves 3, lost 0)
  RawQ 48000100, ReturnQ 48000108, EventQ 48000110
  BufhdrQ 48000128 (2948 items), LovltrQ 48000140 (5 items, 1632 bytes)
  IpcbufQ 48000148 (16 items, 4096 bytes)
  3571 buffer headers (48002000 - 4800FF20)
  pool0: 28 buffers, 256 bytes, queue 48000130
  pool1: 237 buffers, 1536 bytes, queue 48000138
  pool2: 333 buffers, 4544 bytes, queue 48000150
  pool3: 4 buffers, 4576 bytes, queue 48000158
  slot0: EIP, hw 1.5, sw 20.00, ccb 5800FF30, cmdq 48000080, vps 4096
    software loaded from system
    Ethernet0/0, addr 0000.0ca3.cc00 (bia 0000.0ca3.cc00)
      gfreeq 48000138, lfreeq 48000160 (1536 bytes), throttled 0
      rxlo 4, rxhi 42, rxcurr 0, maxrxcurr 2
      txq 48000168, txacc 48000082 (value 27), txlimit 27
                  .........
slot1: FIP, hw 2.9, sw 20.02, ccb 5800FF40, cmdq 48000088, vps 4096
    software loaded from system
    Fddi1/0, addr 0000.0ca3.cc20 (bia 0000.0ca3.cc20)
      gfreeq 48000150, lfreeq 480001C0 (4544 bytes), throttled 0
      rxlo 4, rxhi 165, rxcurr 0, maxrxcurr 0
      txq 480001C8, txacc 480000B2 (value 0), txlimit 95
 slot4: AIP, hw 1.3, sw 20.02, ccb 5800FF70, cmdq 480000A0, vps 8192
    software loaded from system
```

```
   ATM4/0, applique is SONET (155Mbps)
     gfreeq 48000150, lfreeq 480001D0 (4544 bytes), throttled 0
     rxlo 4, rxhi 165, rxcurr 0, maxrxcurr 0
     txq 480001D8, txacc 480000BA (value 0), txlimit 95
slot9: MIP, hw 1.0, sw 20.02, ccb 5800FFC0, cmdq 480000C8, vps 8192
   software loaded from system
   T1 9/0, applique is Channelized T1
     gfreeq 48000138, lfreeq 480001E0 (1536 bytes), throttled 0
     rxlo 4, rxhi 42, rxcurr 0, maxrxcurr 0
     txq 480001E8, txacc 480000C2 (value 27), txlimit 27
               .......

slot10: TRIP, hw 1.1, sw 20.00, ccb 5800FFD0, cmdq 480000D0, vps 4096
   software loaded from system
   TokenRing10/0, addr 0000.0ca3.cd40 (bia 0000.0ca3.cd40)
     gfreeq 48000150, lfreeq 48000200 (4544 bytes), throttled 0
     rxlo 4, rxhi 165, rxcurr 1, maxrxcurr 1
     txq 48000208, txacc 480000D2 (value 95), txlimit 95
              .........

slot11: FSIP, hw 1.1, sw 20.01, ccb 5800FFE0, cmdq 480000D8, vps 8192
   software loaded from flash slot0:pond/bath/rsp_fsip20-1
   Serial11/0, applique is Universal (cable unattached)
     gfreeq 48000138, lfreeq 48000240 (1536 bytes), throttled 0
     rxlo 4, rxhi 42, rxcurr 0, maxrxcurr 0
     txq 48000248, txacc 480000F2 (value 5), txlimit 27
             ...........

Router# dir slot0:pond/bath/rsp_fsip20-1
-#- -length- -----date/time------ name
3   10242    Jan 01 1995 03:46:31 pond/bath/rsp_fsip20-1

Router# dir slaveslot0:pond/bath/rsp_fsip20-1
No such file

4079832 bytes available (3915560 bytes used)

Router# copy slot0:pond/bath/rsp_fsip20-1 slaveslot0:
4079704 bytes available on device slaveslot0, proceed? [confirm]

Router# dir slaveslot0:pond/bath/rsp_fsip20-1
-#- -length- -----date/time------ name
3   10242    Mar 01 1993 02:35:04 pond/bath/rsp_fsip20-1

4069460 bytes available (3925932 bytes used)
```

Specifying Different Startup Images for the Master and Slave RSPs

For software error protection, the RSP cards should have different system images.

When the factory sends you a new Cisco 7507 or Cisco 7513 router with two RSPs, you receive the same system image on both RSP cards. For the software error protection method, you need two different software images on the RSP cards. Thus, you copy a desired image to the master RSP card and modify the **boot system** global configuration commands to reflect booting two different system images. Each RSP card uses its own image to boot the router when it becomes the master.

To specify different startup images for the master and slave RSP, use the following commands beginning in EXEC mode:

	Command	Purpose
Step 1	`Router# dir {bootflash: | slot0: | slot1:}`	Verifies the location and version of the master RSP software image.
Step 2	`Router# dir {slavebootflash: | slaveslot0: | slaveslot1:}`	Determines if the slave RSP contains the same software image in the same location.
Step 3	`Router# copy source-url {bootflash: | slot0: | slot1:}`	Copies a different system image to the master RSP.
Step 4	`Router# configure terminal`	Enters configuration mode from the terminal.

Step 5	`Router(config)# boot system flash bootflash:[filename]`
	`Router(config)# boot system flash slot0:[filename]`
	`Router(config)# boot system flash slot1:[filename]`
		From global configuration mode, configures the master RSP to boot the new image from the appropriate location.

Step 6	`Router(config)# boot system flash`
	`Router(config)# bootflash:[filename]`
	`Router(config)# boot system flash slot0:[filename]`
	`Router(config)# boot system flash slot1:[filename]`
		Also, add a **boot system** command that specifies the slave's boot image and location. This is the boot image that the slave uses when it becomes the master RSP and boots the system. Note that because the slave will boot this image when the slave is actually the new master RSP, the command syntax does not use a "**slave**" prefix.

Step 7	`Router(config)# boot system {rcp | tftp | ftp} [filename] [ip-address]`	(Optional) Configures the master RSP to boot from a network server.
Step 8	`Router(config)# config-register value`[1]	Sets the configuration register to enable the system to load the system image from a network server or from Flash.
Step 9	`Router(config)# end`	Exits configuration mode.

Step 10	`Router# copy system:running-config nvram:startup-config`
	OR
	`Router# copy running-config startup-config`
		Saves the configuration file to the master's startup configuration. Because automatic synchronization is turned on, this step saves the **boot system** commands to the master and slave startup configuration.

Step 11	`Router# reload`	Resets the router with the new configuration information.

1. Refer to the "Modifying the Configuration Register Boot Field" section on page 225 for more information on systems that can use this command to modify the software configuration register.

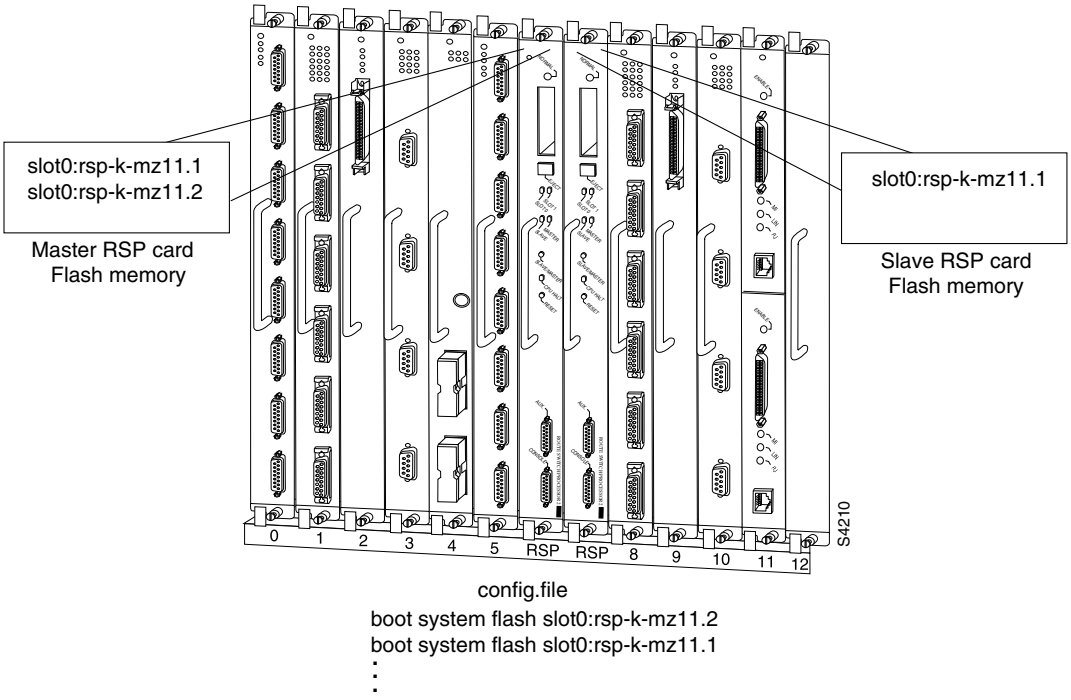
Upgrading to a New Software Version Example

In this example, assume the following:

- The master RSP is in processor slot 6 and the slave RSP is in processor slot 7 of a Cisco 7513 router.
- The system has the same image rsp-k-mz11.1 in PCMCIA slot 0 of both the master and slave RSP card.
- You want to upgrade to Cisco IOS Release 12.0, but you want to guard against software failures. So, you configure Dual RSP operation for software error protection.

Figure 31 illustrates the software error protection configuration for this example. The configuration commands for this configuration follow the figure.

Figure 31 Software Error Protection: Upgrading to a New Software Version



Because you always view the environment from the master RSP perspective, in the following command you view the master's slot 0 to verify the location and version of the master's software image:

```
Router# dir slot0:
-#- -length- -----date/time------ name
1    3482496  May 4 1993 21:38:04 rsp-k-mz11.1

7993896 bytes available (1496 bytes used)
```

Now view the slave's software image location and version:

```
Router# dir slaveslot0:
-#- -length- -----date/time------ name
1    3482496  May 4 1993 21:38:04 rsp-k-mz11.1

7993896 bytes available (1496 bytes used)
```

Because you want to run the Release 12.0 system image on one RSP card and the Release 11.1 system image on the other RSP card, copy the Release 12.0 system image to the master's slot 0:

```
Router# copy tftp: slot0:rsp-k-mz12.0
```

Enter global configuration mode and configure the system to boot first from a Release 12.0 system image and then from a Release 11.1 system image:

```
Router# configure terminal
Router (config)# boot system flash slot0:rsp-k-mz12.0
Router (config)# boot system flash slot0:rsp-k-mz11.1
```

With this configuration, when the slot 6 RSP card is master, it looks first in its PCMCIA slot 0 for the system image file rsp-k-mz11.2 to boot. Finding this file, the router boots from that system image. When the slot 7 RSP card is master, it also looks first in its slot 0 for the system image file rsp-k-mz12.0 to boot. Because that image does not exist in that location, the slot 7 RSP card looks for the system image

file rsp-k-mz11.1 in slot 0 to boot. Finding this file in its PCMCIA slot 0, the router boots from that system image. In this way, each RSP card can reboot the system using its own system image when it becomes the master RSP card.

Configure the system further with a fault-tolerant booting strategy:

```
Router (config)# boot system tftp rsp-k-mz11.1 192.168.1.25
```

Set the configuration register to enable loading of the system image from a network server or from Flash and save the changes to the master and slave startup configuration file:

```
Router (config)# config-register 0x010F
Router (config)# end
Router# copy system:running-config nvram:startup-config
```

Reload the system so that the master RSP uses the new Release 12.0 system image:

```
Router# reload
```

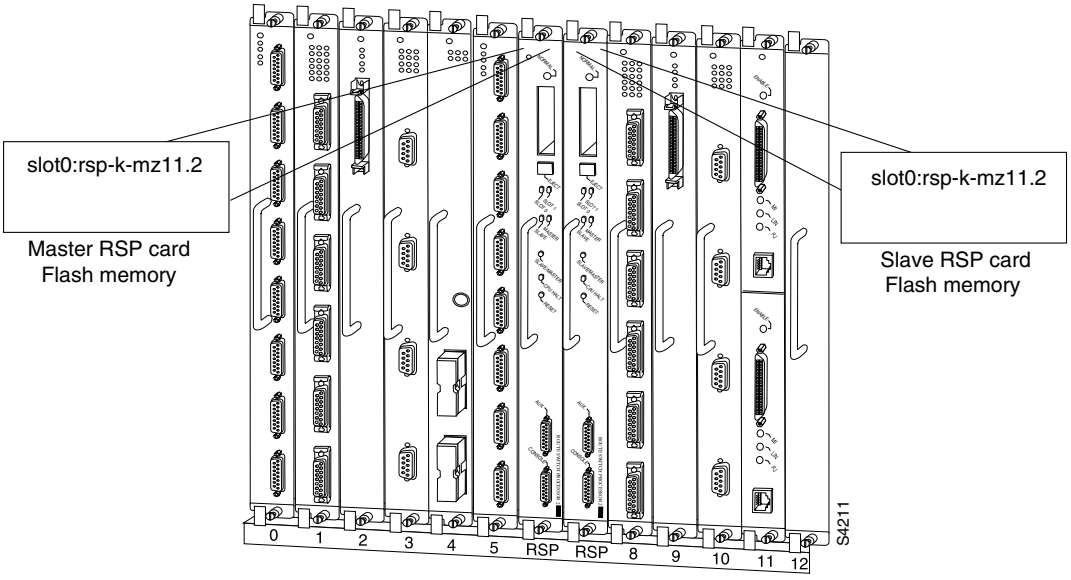
Dual RSP: Backing Up with an Older Software Version Example

In the following example, assume the following:

- The master RSP is in processor slot 6 and the slave RSP is in processor slot 7 of a Cisco 7513 router.
- The system has the same image rsp-k-mz11.2 in PCMCIA slot 0 of both the master and slave RSP card.
- You want to use to Cisco IOS Release 11.1 as backup to guard against software failures. So, you configure Dual RSP operation for software error protection.

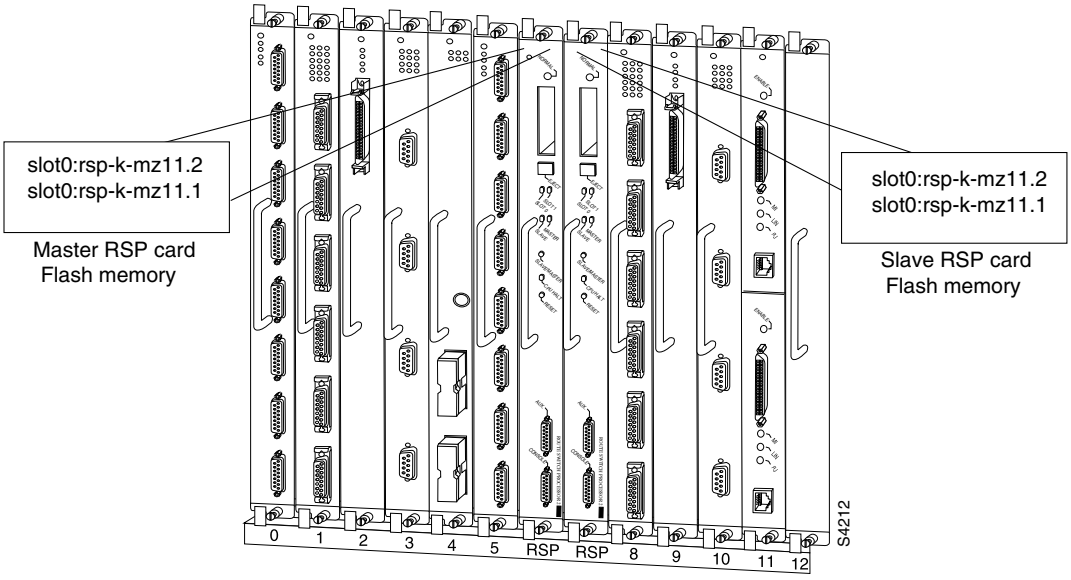
In this scenario, you begin with the configuration shown in Figure 32.

Figure 32 Software Error Protection: Backing Up with an Older Software Version, Part I



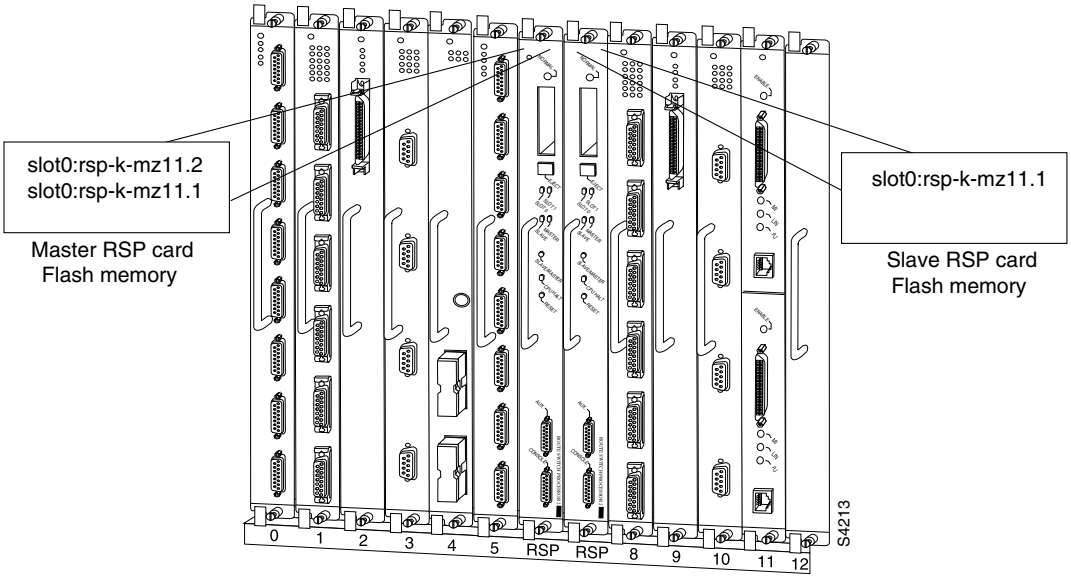
First, copy the rsp-k-mz11.1 image to the master and slave RSP card, as shown in Figure 33.

Figure 33 Software Error Protection: Backing Up with an Older Software Version, Part 2



Next, you delete the rsp-k-mz11.2 image from the slave RSP card. The final configuration is shown in Figure 34.

Figure 34 Software Error Protection: Backing Up with an Older Software Version, Part 3



The following commands configure software error protection for this example scenario.

View the master and slave slot 0 to verify the location and version of their software images:

```
Router# dir slot0:
-#- -length- -----date/time------ name
1    3482498  May 4 1993 21:38:04 rsp-k-mz11.2
```

```
7993896 bytes available (1496 bytes used)

Router# dir slaveslot0:
-#- -length- -----date/time------ name
1    3482498  May 4 1993 21:38:04 rsp-k-mz11.2

7993896 bytes available (1496 bytes used)
```

Copy the Release 11.1 system image to the master and slave slot 0:

```
Router# copy tftp: slot0:rsp-k-mz11.1
Router# copy tftp: slaveslot0:rsp-k-mz11.1
```

Delete the rsp-k-mz11.2 image from the slave RSP card:

```
Router# delete slaveslot0:rsp-k-mz11.2
```

Configure the system to boot first from a Release 11.2 system image and then from a Release 11.1 system image:

```
Router# configure terminal
Router (config)# boot system flash slot0:rsp-k-mz11.2
Router (config)# boot system flash slot0:rsp-k-mz11.1
```

Configure the system further with a fault-tolerant booting strategy:

```
Router (config)# boot system tftp rsp-k-mz11.1 192.168.1.25
```

Set the configuration register to enable loading of the system image from a network server or from Flash and save the changes to the master and slave startup configuration file:

```
Router (config)# config-register 0x010F
Router (config)# end
Router# copy system:running-config nvram:startup-config
```

Note You do not need to reload the router in this example, because the router is currently running the Release 11.2 image.

Setting Environment Variables on the Master and Slave RSP

You can set environment variables on both RSP cards in a Cisco 7507 and Cisco 7513 router.

Note When you configure Dual RSP operation, we recommend that you use the default environment variables. If you change the variables, we recommend setting the same device for equivalent environment variables on each RSP card. For example, if you set one RSP card's CONFIG_FILE environment variable device to NVRAM, set the other RSP card's CONFIG_FILE environment variable device to NVRAM as well.

You set environment variables on the master RSP just as you would if it were the only RSP card in the system. Refer to the following sections for more information on these steps:

- "Specifying the Startup System Image in the Configuration File" section on page 193 (in the "Loading and Maintaining System Images and Microcode" chapter).
- "Controlling Environment Variables" section on page 229.

- "Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems" section on page 167 (in the "Modifying, Downloading, and Maintaining Configuration Files" chapter).

You can set the same environment variables on the slave RSP card, manually or automatically. The following sections describe these two methods:

- Automatically Setting Environment Variables on the Slave RSP
- Manually Setting Environment Variables on the Slave RSP

Automatically Setting Environment Variables on the Slave RSP

With automatic synchronization turned on, the system automatically saves the same environment variables to the slave's startup configuration when you set the master's environment variables and save them.

Note Automatic synchronization mode is on by default. To turn off automatic synchronization, use the **no slave auto-sync config** global configuration command.

To set environment variables on the slave RSP when automatic synchronization is on, set the environment variables as described in the "Rebooting" chapter of this book. You can verify the boot variable using the **show bootvar** EXEC mode command.

Manually Setting Environment Variables on the Slave RSP

If you disable automatic synchronization of configuration files, you must manually synchronize the slave's configuration file to the master's configuration file to store environment variables on the slave RSP.

Once you set the master's environment variables, you can manually set the same environment variables on the slave RSP card using the **slave sync config** EXEC command.

To manually set environment variables on the slave RSP, perform the following procedure:

Step 1 Set the environment variables for the master RSP card as described in the "Rebooting" chapter of this book.

Step 2 Save the configuration using the **copy system:running-config nvram:startup-config** EXEC command.

Step 3 Save the same environment variable configuration to the slave RSP using the **slave sync config** privileged EXEC command. Issuing this command will synchronize the configuration files.

Step 4 Verify the environment variable settings using the **show bootvar** EXEC command.

Monitoring and Maintaining Dual RSP Operation

To monitor and maintain Dual RSP operation, complete the following tasks in the following sections:

- Overriding the Slave Image Bundled with the Master Image
- Manually Synchronizing Configuration Files

- Troubleshooting and Reloading a Failed RSP Card
- Disabling Access to the Slave Console
- Displaying Information About Master and Slave RSP Cards

Overriding the Slave Image Bundled with the Master Image

To override the slave image that is bundled with the master image, use the following command in global configuration mode:

Command	Purpose
`Router(config)# slave image {system | file-url}`	Specifies which image the slave runs.

Manually Synchronizing Configuration Files

To manually synchronize configuration files and ROM monitor environment variables on the master and slave RSP card, use the following command in privileged EXEC mode:

Command	Purpose
`Router(config)# slave sync config`	Manually synchronizes the master and slave configuration files.

Caution When you install a second RSP card for the first time, you *must* immediately configure it using the **slave sync config** command. This ensures that the new slave is configured consistently with the master. Failure to do so can result in an unconfigured slave RSP card taking over mastership of the router when the master fails, potentially rendering the network inoperable.

The **slave sync config** command is also a useful tool for more advanced implementation methods not discussed in this chapter.

Troubleshooting and Reloading a Failed RSP Card

When a new master RSP card takes over mastership of the router, it automatically reboots the failed RSP card as the slave RSP card. You can access the state of the failed RSP card in the form of a stack trace from the master console using the **show stacks** EXEC command.

The **debug oir** command is used to debug the online insertion and removal (OIR) feature (which is also known as hot-swapping or power-on servicing). The **debug oir** command often is useful in debugging problems related to OIR, including single line card reloading.

You can also manually reload a failed, inactive RSP card from the master console. This task returns the card to the active slave state. If the master RSP fails, the slave will be able to become the master. To manually reload the inactive RSP card, use the following command in global configuration mode:

Command	Purpose
`Router(config)# slave reload`	Reloads the inactive slave RSP card.

Disabling Access to the Slave Console

The slave console does not have enable password protection. Thus, a user connected to the slave console port can enter privileged EXEC mode and view or erase the configuration of the router. Use the **no slave terminal** global configuration command to disable slave console access and prevent security problems. When the slave console is disabled, users cannot enter commands.

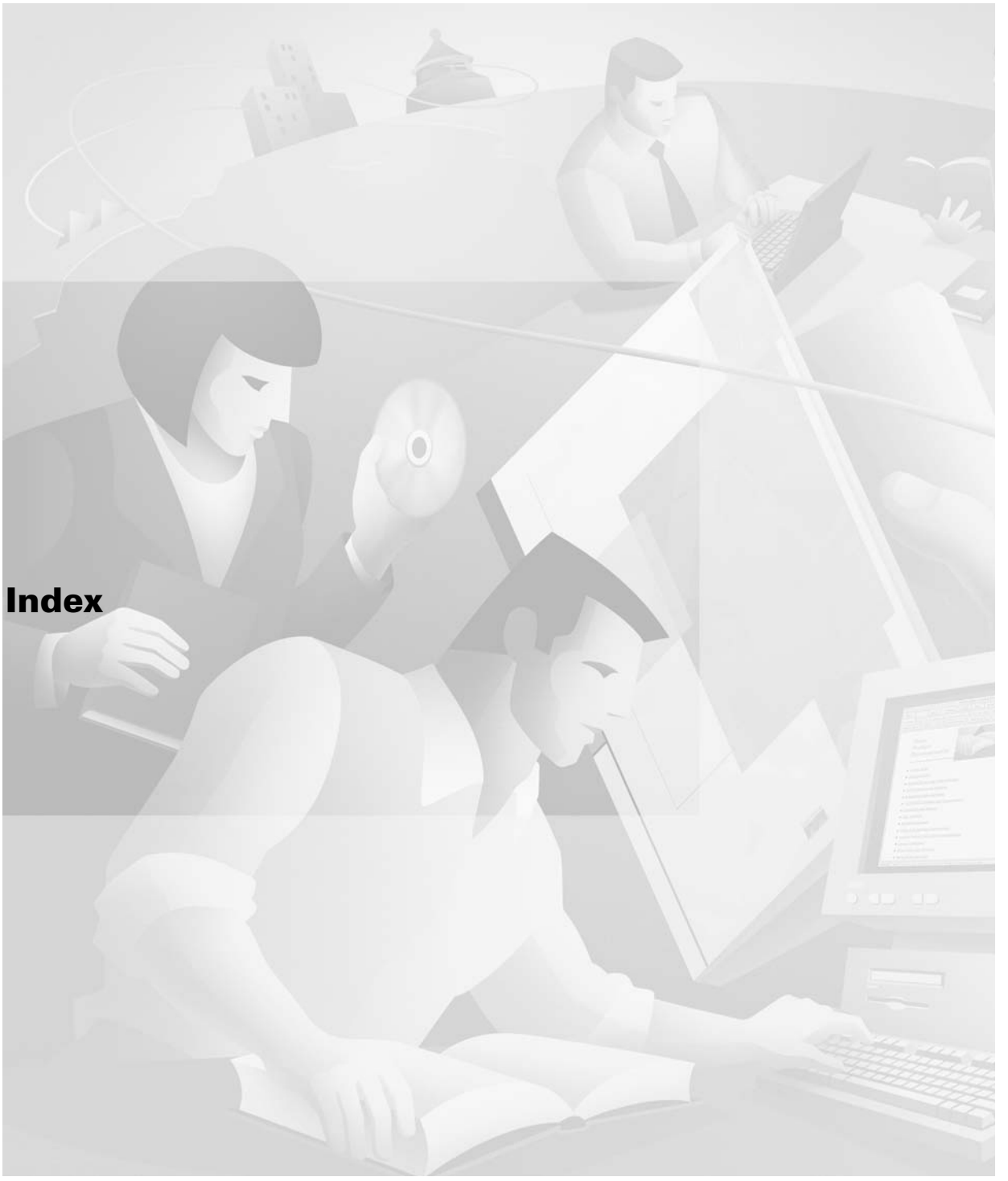
If slave console access is disabled, the following message appears periodically on the slave console:

```
%%Slave terminal access is disabled. Use "slave terminal" command in master RSP
configuration mode to enable it.
```

Displaying Information About Master and Slave RSP Cards

To display information about both the master and slave RSP cards, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show bootvar`	Displays the environment variable settings and configuration register settings for both the master and slave RSP cards.
`Router# show file systems`	Displays a list of Flash devices currently supported on the router.
`Router# show version`	Displays the software version running on the master and slave RSP card.
`Router# show stacks`	Displays the stack trace and version information of the master and slave RSP cards.



Index

BC	Cisco IOS Bridging and IBM Networking Configuration Guide
DC	Cisco IOS Dial Technologies Configuration Guide
FC	Cisco IOS Configuration Fundamentals Configuration Guide
IC	Cisco IOS Interface Configuration Guide
IPC	Cisco IOS IP Routing Configuration Guide
MWC	Cisco IOS Mobile Wireless Configuration Guide
P2C	Cisco IOS AppleTalk and Novell IPX Configuration Guide
P3C	Cisco IOS Apollo Domain, Banyan VINES, DECnet, ISO CLNS, and XNS Configuration Guide
QC	Cisco IOS Quality of Service Solutions Configuration Guide
SC	Cisco IOS Security Configuration Guide
TC	Cisco IOS Terminal Services Configuration Guide
VC	Cisco IOS Voice, Video, and Fax Configuration Guide
WC	Cisco IOS Wide-Area Networking Configuration Guide
XC	Cisco IOS Switching Services Configuration Guide

Symbols

Numerics

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y