# Router Products Command Reference

The products and specifications, configurations, and other technical information regarding the products contained in this manual are subject to change without notice. All statements, technical information, and recommendations contained in this manual are believed to be accurate and reliable but are presented without warranty of any kind, express or implied, and users must take full responsibility for their application of any products specified in this manual.

This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual for this device, may cause interference to radio communications. This equipment has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case users at their own expense will be required to take whatever measures may be required to correct the interference.



Digital Equipment Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

# TABLE OF CONTENTS

# PART 2

## System and Interface Configuration and Management

### Chapter 3

## PART 3
## Wide Area Networking

**Chapter 7**

# PART 5
## Bridging

### Chapter 22

## PART 6
## IBM Networking

# LIST OF TABLES

# About This Manual

This section discusses the objectives, audience, organization, and conventions of the *Router Products Command Reference* publication.

## Document Objectives

This publication provides an in-depth description of the commands necessary for configuring and maintaining your router. It describes tasks only in the context of using a particular command; it does not describe how the tasks interrelate or provide comprehensive configuration examples.You can use this publication as a standalone reference manual or in conjunction with the *Router Products Configuration Guide*. Not all of the **debug** commands are included in this publication, but you can find all of them in the *Debug Command Reference* publication.

## Audience

This publication is intended as a standalone document for experienced network administrators who will be configuring and maintaining routers and would like to reference commands. For less-experienced users who need to understand the tasks as well as the commands, it is intended as a companion guide to the *Router Products Configuration Guide*.

## Document Organization

This publication is divided into six main parts. Each part comprises chapters describing related tasks or functions. The organization of parts and chapters in this publication matches the organization of parts and chapters in the *Router Products Configuration Guide*, except that this document contains appendixes. The parts in this publication are as follows:

- Part 1, "Product Introduction," contains an overview of the router and command descriptions for the system user interface and command parser.
- Part 2, "System and Interface Configuration and Management," describes the commands pertaining to booting, terminal sessions and modem lines, system management, and system interfaces.
- Part 3, "Wide-Area Networking," describes the tasks pertaining to ATM, DDR, Frame Relay, HDLC, ISDN, PPP, SLIP, SMDS, and X.25. The chapters are arranged in alphabetical order for ease of use.

- Part 4, “Routing Protocols,” contains chapters that describe the commands used to configure each supported network protocol. These protocols include Apollo Domain, AppleTalk, Banyan VINES, DECnet, IP, ISO CLNS, Novell IPX, and XNS (including Ungermann-Bass and 3Com variations). This part also contains a chapter that discusses commands for IP routing protocols, which include IGRP, BGP, RIP, OSPF, IS-IS, and ISO-IGRP. The chapters are arranged in alphabetical order for ease of use.
- Part 5, “Bridging,” contains chapters that describe the commands used to configure transparent bridging, source-route bridging, source-route transparent (SRT) bridging, and source-route translational bridging (SR/TLB) on our routers/bridges.
- Part 6, “IBM Networking,” contains chapters that describe the commands used to configure the SDLC transport and serial tunneling mechanisms in an IBM local-area network. Also included are the commands for configuring the local acknowledgment feature, managing your source-route bridges with LAN Network Manager, and configuring SDLLC and QLLC conversion, our IBM network protocol translation features. Part 6 also contains chapters that describe commands used to configure SNA Downstream Physical Unit (DSPU) support and SNA Frame Relay Access Support. The IBM Channel Attach chapter documents the Channel Interface Processor (CIP) commands.

The appendixes contain a list of references and recommended reading, Ethernet type codes, regular expressions, a table of the ASCII character set, switching information, and a description of IOS 10.3 features supported by specific router platforms.

# Document Conventions

Software and hardware documentation uses the following conventions:

- The symbol ^ represents the Control key.

  For example, the key combinations ^D and Ctrl-D mean hold down the Control key while you press the D key. Keys are indicated in capitals, but are not case sensitive.
- A string is defined as a nonquoted set of characters.

  For example, when setting an SNMP community string to “public,” do not use quotes around the string, or the string will include the quotation marks.

Command descriptions use these conventions:

- Vertical bars ( | ) separate alternative, mutually exclusive, elements.
- Square brackets ([ ]) indicate optional elements.
- Braces ({ }) indicate a required choice.
- Braces within square brackets ([{ }]) indicate a required choice within an optional element.
- **Boldface** indicates commands and keywords that are entered literally as shown.
- *Italics* indicate arguments for which you supply values; in contexts that do not allow italics, arguments are enclosed in angle brackets (< >).

Examples use these conventions:

- Examples that contain system prompts denote interactive sessions, indicating that the user enters commands at the prompt. The system prompt indicates the current command mode. For example, the prompt `router(config)#` indicates global configuration mode.
- Terminal sessions and information the system displays are in `screen` font.
- Information you enter is in **`boldface screen`** font.

- Nonprinting characters, such as passwords, are in angle brackets (< >).
- Default responses to system prompts are in square brackets ([ ]).
- Exclamation points (!) at the beginning of a line indicate a comment line. They are also displayed by the router for certain processes.

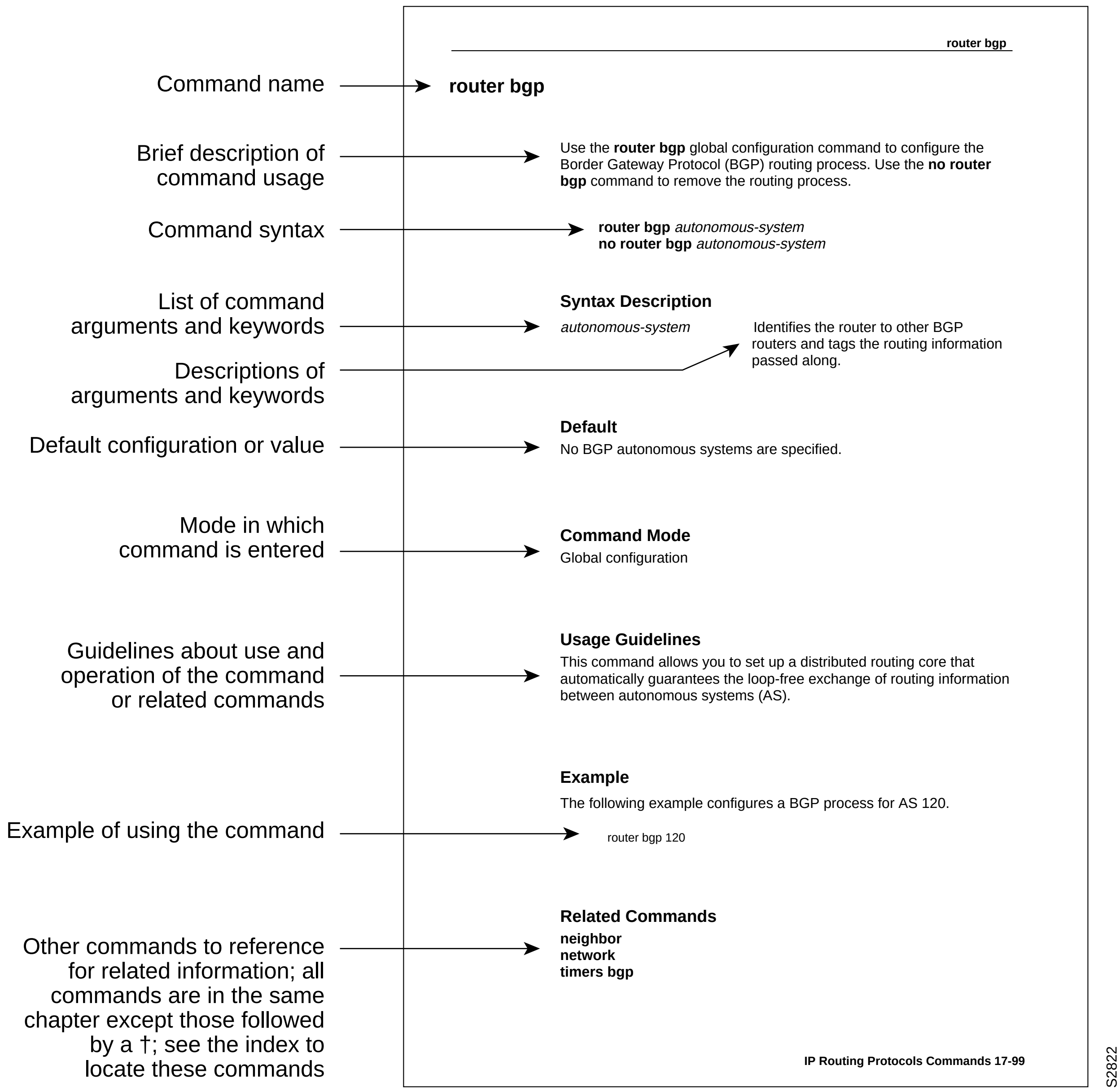
**Caution** Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

---

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

---

The following illustration explains the fields on a typical command reference page:


Command name
Brief description of command usage
Command syntax
List of command arguments and keywords
Descriptions of arguments and keywords
Default configuration or value
Mode in which command is entered
Guidelines about use and operation of the command or related commands
Example of using the command
Other commands to reference for related information; all commands are in the same chapter except those followed by a †; see the index to locate these commands
router bgp
router bgp
Use the router bgp global configuration command to configure the Border Gateway Protocol (BGP) routing process. Use the no router bgp command to remove the routing process.
router bgp autonomous-system
no router bgp autonomous-system
Syntax Description
autonomous-system
Identifies the router to other BGP routers and tags the routing information passed along.
Default
No BGP autonomous systems are specified.
Command Mode
Global configuration
Usage Guidelines
This command allows you to set up a distributed routing core that automatically guarantees the loop-free exchange of routing information between autonomous systems (AS).
Example
The following example configures a BGP process for AS 120.
router bgp 120
Related Commands
neighbor
network
timers bgp
IP Routing Protocols Commands 17-99
S2822

PART 1

# Product Introduction

# Router Product Overview

Computer networks that run different protocols on a variety of local-area network (LAN) media over a variety of wide-area network (WAN) technologies must be able to communicate. The Cisco Systems Internetwork Operating System (IOS) software provides this capability. The IOS software runs on internetworking platforms purchased directly from Cisco Systems and from many Cisco partners.

This chapter describes the capabilities of the IOS as implemented on router platforms. It contains the following sections:

- IOS Software Benefits
- Supported Network Protocols
- Supported IP Routing Protocols
- Supported Media
- Supported Platforms
- Configuring the Router

We provide various documents about your router. Refer to the *Documentation Roadmap* for information about the interrelationship among the various documents. For the latest information about the software, including new features added since the documentation was printed and additional caveats about using the software, refer to the release note that accompanies the software.

## IOS Software Benefits

The IOS software supports users and applications throughout the enterprise and provides security and data integrity for the internetwork. The IOS software cost-effectively manages resources by controlling and unifying complex, distributed network intelligence. It also functions as a flexible vehicle for adding new services, features, and applications to the internetwork.

The IOS software provides four types of internetwork benefits, which are described in the following sections:

- Reliable, Adaptive Routing
- WAN Optimization
- Management and Security
- Scalability

## Reliable, Adaptive Routing

The IOS software is reliable and adaptive because it identifies the best paths and routes traffic around network failures. It also reduces costs by efficiently using network bandwidth and resources while eliminating needless management of static routes.

Policy-based IOS features such as route filtering and routing information translatability save network resources by preventing data from being unnecessarily broadcast to nodes that do not require it. Priority output queuing and custom queuing grant priority to important sessions when network bandwidth is saturated. Load balancing makes use of all available paths across the internetwork, preserving valuable bandwidth and improving performance. The IOS software also provides the most effective and efficient scaling available for network applications that require transparent or source-route bridging algorithms.

Increasingly, internetworks are incorporating new technologies such as Asynchronous Transfer Mode (ATM) and LAN switching. Through CiscoFusion, Cisco's scalable architecture for switched internetworks, the IOS software provides the framework for a new technology called multilayer switching, which fuses the ease of switching solutions with the power of routed solutions.

By distributing routing intelligence and switching functions to create "virtual LANs," CiscoFusion's multilayer switching increases bandwidth while simplifying moves, additions, and changes across the enterprise. This extends the power and flexibility of the IOS beyond internetwork routers to include the ATM and LAN switches that are increasingly being deployed throughout today's internetworks.

## WAN Optimization

Because most network costs are expended on WAN switching and usage functions, an effective internetwork must optimize all WAN-related operations. Optimization increases network throughput while reducing delay time. It also minimizes costs by eliminating unnecessary traffic and intelligently selecting the most economical WAN links available.

The IOS software seamlessly accommodates circuit-switched services such as Integrated Services Digital Network (ISDN), switched T1, and dial-up telephone lines. IOS software innovations such as dial-on-demand access and dial backup capabilities provide cost-effective alternatives to point-to-point switched leased lines. Support for advanced, packet-switched services such as X.25, Frame Relay, Switched Multimegabit Data Service (SMDS), and ATM extends the internetwork across the broad range of WAN interface alternatives now available.

## Management and Security

The IOS software provides an array of network management and security capabilities designed to meet the needs of today's large, complex internetworks. Integrated management simplifies administrative procedures and shortens the time required to diagnose and fix problems. Automated operations reduce hands-on tasks and make it possible to manage large, geographically dispersed internetworks with a small staff of experts located at a central site.

The IOS software provides several important management features that are built into every Cisco router. These include configuration services that lower the cost of installing, upgrading, and reconfiguring routers, as well as comprehensive monitoring and diagnostic services. In addition, the IOS software provides valuable information and services to router management applications developed by Cisco and its partners. The Cisco applications, known collectively as CiscoWorks, offer administrators a wide-ranging suite of operational, design, and management capabilities that increase productivity and reduce costs.

The IOS management services are matched by their security capabilities. The IOS software includes a diverse tool kit for partitioning resources and prohibiting access to sensitive or confidential information or processes. Multidimensional filters prevent users from knowing that other users or resources are even on the network. Encrypted passwords, dial-in authentication, multilevel configuration permissions, and accounting and logging features provide protection from—and information about—unauthorized access attempts.

### Scalability

Scalability provide the flexibility required to address all of the key issues facing internetworks as organizations evolve. The IOS software's scalable routing protocols help avoid needless congestion, overcome inherent protocol limitations, and bypass many of the obstacles that can arise because of the scope and geographical dispersion of an internetwork.

The IOS software also helps to cut costs by reducing network bandwidth and processing overhead, off-loading servers and conserving resources, and easing system configuration tasks. Advanced IOS features such as filtering, protocol termination and translation, smart broadcasts, and helper address services combine to create a flexible, scalable infrastructure that can keep pace with evolving network requirements.

## Supported Network Protocols

IOS software supports many networking protocols, as well as their associated routing protocols. These protocols are based on both open standards and proprietary protocols from a variety of vendors. The IOS software also supports a wide set of bridging and IBM connectivity solutions.

The IOS Software can receive and forward packets concurrently from any combination of the following:

- WAN protocols
  - Asynchronous Transfer Mode (ATM)
  - Frame Relay
  - High-Level Data Link Control (HDLC)
  - Integrated Services Digital Networks (ISDN)
  - Point-to-Point Protocol (PPP)
  - Serial Line Internet Protocol (SLIP)—for asynchronous lines
  - Switched Multimegabit Data Service (SMDS)
  - X.25 and its derivatives, including Link Access Procedure, Balanced (LAPB) and Defense Data Network (DDN) X.25
- Network protocols
  - Apollo Domain
  - AppleTalk (Phase 1 and Phase 2)
  - Banyan VINES
  - DECnet Phase IV, Phase IV Prime, and Phase V
  - Internet Protocol (IP)
  - ISO Connectionless Network Services (CLNS) and Connection Mode Network Services (CMNS)

- Novell IPX
- Xerox Network Systems (XNS) and two variations developed by Ungermann-Bass and 3Com

- Bridging types
  - Transparent bridging and source-route transparent (SRT) bridging
  - Source-route bridging (SRB) and remote source-route bridging (RSRB)
  - Source-route translational bridging (SR/TLB)
- Support for IBM networking
  - Serial tunnel (STUN)
  - Logical Link Control, type 2 (LLC2) and Synchronous Data Link Control (SDLC)
  - SDLLC—A software feature that translates between LLC2 and Synchronous Data Link Control (SDLC)
  - Qualified Logical Link Control (QLLC) conversion
  - IBM Channel Attach

These protocols, bridging, and IBM networking topics are described in separate chapters of the *Router Products Configuration Guide*. For background information, refer to the *Internetworking Technology Overview* publication.

# Supported IP Routing Protocols

The IOS software supports the following IP routing protocols:

- Interior Gateway Protocols
  - Internet Gateway Routing Protocol (IGRP)
  - Enhanced IGRP
  - Open Shortest Path First (OSPF)
  - Routing Information Protocol (RIP)
  - Intermediate System-to-Intermediate System (IS-IS)
- Exterior Gateway Protocols
  - Border Gateway Protocol (BGP)
  - Exterior Gateway Protocol (EGP)
- Router Discovery Protocols
  - ICMP Router Discovery Protocol (IRDP)
  - Hot Standby Router Protocol (HSRP)

The “Configuring IP Routing Protocols” chapter in the *Router Products Configuration Guide* describes these protocols in detail.

# Supported Media

Our routers support the following industry-standard networking media:

- Asynchronous serial
- Channelized T1
- Ethernet—IEEE 802.3 and Type II
- Fiber Distributed Data Interface (FDDI)—single and dual mode
- High-Speed Serial Interface (HSSI)—supports T1, T3, E3, and SONET rates
- ISDN Basic Rate Interface (BRI) and Multiport BRI (MBRI)
- ISDN Primary Rate Interface (PRI)
- Synchronous serial—V.35, RS-232, RS-449, RS-530, X.21, and G.703
- Token Ring—IEEE 802.5

These media are described briefly in the "Configuring Interfaces" chapter of the *Router Products Configuration Guide*. For additional information, refer to the *Internetworking Technology Overview* publication.

# Supported Platforms

The IOS software runs on a variety of Cisco internetworking devices and partners' platforms. For details on the supported platforms, refer to the *Cisco Systems Products Catalogue*.

# Configuring the Router

The following sections describe alternative mechanisms for configuring a router:

- Using Cisco Configuration Builder
- Using the Command Interpreter

## Using Cisco Configuration Builder

Cisco's Configuration Builder lets you create configuration files for multiple routers without knowing the router command-line language or syntax. It is a Microsoft Windows-based application that runs on an IBM PC or compatible computer.

To use Configuration Builder, refer to the *Cisco Configuration Builder Getting Started Guide*.

If you do not have the platform to run Configuration Builder, configure your router using the command interpreter, as described in the next section.

## Using the Command Interpreter

You can build most straightforward router configurations and create a configuration file using the **setup** facility. This facility is described in the *Router Products Getting Started Guide*.

In order to configure your router, you must decide the following:

- What network protocols you are supporting (for example, AppleTalk, IP, Novell IPX, and so on)
- Your addressing plan for each network protocol

- What WAN protocols you will run on each interface (for example, Frame Relay, HDLC, SMDS, X.25, and so on)
- What routing protocol you will use for each network protocol

The *Router Products Getting Started Guide* contains worksheets to help you plan your router configuration.

To enhance the configuration, perform the protocol-specific tasks described in the appropriate chapters of the *Router Products Configuration Guide.*

The router software provides a user interface called a command interpreter, or EXEC, that lets you configure and manage the router. This user interface also provides context-sensitive help. The command interpreter has several command modes, each of which provides a group of related commands that you can use to configure the router and display its status. Some commands are available to all users; others can be executed only after the user enters an enabling password. Context-sensitive help gives information about command syntax. The command interpreter and its help feature are described in the "Understanding the User Interface" chapter of the *Router Products Configuration Guide*.

You use the command interpreter (also known as the command-line parser) to configure interfaces, terminal sessions, and asynchronous communications lines. Interfaces are connections to network media, such as Ethernet, Token Ring, and serial media. You configure them to run different routing protocols and other networking protocols. You configure terminal sessions and modems connected to the router so that other network users can log in to the router. Configuring terminal sessions and asynchronous communications lines is discussed in the "Configuring Terminal Lines and Modem Support" chapter of the *Router Products Configuration Guide*. Configuring interfaces is described in the "Configuring Interfaces" chapter of the *Router Products Configuration Guide*; the routing, bridging, and IBM protocols you can configure on these interfaces are described in the protocol-specific chapters of the *Router Products Configuration Guide*.

You also can configure and manage the router itself, performing such tasks as naming the router, setting the router's time, configuring SNMP, and setting security. These tasks are described in the "Managing the System" chapter of the *Router Products Configuration Guide*.

CHAPTER 2

# User Interface Commands

This chapter describes the commands used to enter and exit the various Internetwork Operating System (IOS) configuration command modes. It provides a description of the **help** command and help features, lists the command editing keys and functions, and details the command history feature.

You can abbreviate the syntax of IOS configuration commands. The router recognizes a command when you enter enough characters of the command to uniquely identifyit.

For user interface task information and examples, see the “Understanding the User Interface” chapter of the *Router Products Configuration Guide*.

# disable

To exit privileged EXEC mode and return to user EXEC mode, enter the **disable** EXEC command.

**disable** [*level*]

## Syntax Description

| | |
|---|---|
| *level* | Privilege level to exit to. |

## Command Mode

EXEC

## Example

In the following example, entering the **disable** command causes the system to exit privileged EXEC mode and return to user EXEC mode as indicated by the angle bracket (>):

```
Router# disable
Router>
```

## Related Command

**enable**

# editing

To enable enhanced editing mode for a particular line, use the **editing** line configuration command. To disable the enhanced editing mode, use the **no** form of this command.

**editing**
**no editing**

## Syntax Description

This command has no arguments or keywords.

## Default

Enabled

## Command Mode

Line configuration

## Usage Guidelines

Table 2-1 provides a description of the keys used to enter and edit commands. Ctrl indicates the Control key. It must be pressed simultaneously with its associated letter key. Esc indicates the Escape key. It must be pressed first, followed by its associated letter key. Keys are *not* case sensitive.

**Table 2-1 Editing Keys and Functions for Software Release 9.21 and Later**

| Keys | Function |
|---|---|
| Tab | Completes a partial command name entry. When you enter a unique set of characters and press the Tab key, the system completes the command name. If you enter a set of characters that could indicate more than one command, the system beeps to indicate an error. Enter a question mark (?) immediately following the partial command (no space). The system provides a list of commands that begin with that string. |
| Delete or Backspace | Erases the character to the left of the cursor. |
| Return | At the command line, pressing the Return key performs the function of processing a command. At the "---More---" prompt on a terminal screen, pressing the Return key scrolls down a line. |
| Space Bar | Allows you to see more output on the terminal screen. Press the space bar when you see the line "---More---" on the screen to display the next screen. |
| Left Arrow[1] | 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 key repeatedly to scroll back toward the system prompt and verify the beginning of the command entry. |
| Right Arrow[1] | Moves the cursor one character to the right. |
| Up Arrow[1] or Ctrl-P | Recalls commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands. |
| Down Arrow[1] or Ctrl-N | Return to more recent commands in the history buffer after recalling commands with the Up Arrow or Ctrl-P. Repeat the key sequence to recall successively more recent commands. |
| Ctrl-A | Moves the cursor to the beginning of the line. |

| Keys | Function |
|---|---|
| Ctrl-B | Moves the cursor back one character. |
| Ctrl-D | Deletes the character at the cursor. |
| Ctrl-E | Moves the cursor to the end of the command line. |
| Ctrl-F | Moves the cursor forward one character. |
| Ctrl-K | Deletes all characters from the cursor to the end of the command line. |
| Ctrl-L and Ctrl-R | Redisplays the system prompt and command line. |
| Ctrl-T | Transposes the character to the left of the cursor with the character located at the cursor. |
| Ctrl-U and Ctrl-X | Deletes all characters from the cursor back to the beginning of the command line. |
| Ctrl-V and Esc Q | Inserts a code to indicate to the system that the keystroke immediately following should be treated as a command entry, *not* as an editing key. |
| Ctrl-W | Deletes the word to the left of the cursor. |
| Ctrl-Y | Recalls the most recent entry in the delete buffer. The delete buffer contains the last ten items you have deleted or cut. Ctrl-Y can be used in conjunction with Esc Y. |
| Ctrl-Z | Ends configuration mode and returns you to the EXEC prompt. |
| Esc B | Moves the cursor back one word. |
| Esc C | Capitalizes the word at the cursor. |
| Esc D | Deletes from the cursor to the end of the word. |
| Esc F | Moves the cursor forward one word. |
| Esc L | Changes the word at the cursor to lowercase. |
| Esc U | Capitalizes from the cursor to the end of the word. |
| Esc Y | Recalls the next buffer entry. The buffer contains the last ten items you have deleted. Press Ctrl-Y first to recall the most recent entry. Then press Esc Y up to nine times to recall the remaining entries in the buffer. If you bypass an entry, continue to press Esc Y to cycle back to it. |

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

Table 2-2 lists the editing keys and functions of the earlier software release.

**Table 2-2 Editing Keys and Functions for Software Release 9.1 and Earlier**

| Key | Function |
|---|---|
| Delete or Backspace | Erases the character to the left of the cursor. |
| Ctrl-W | Erases a word. |
| Ctrl-U | Erases a line. |
| Ctrl-R | Redisplays a line. |
| Ctrl-Z | Ends configuration mode and returns to the EXEC prompt. |
| Return | Executes single-line commands. |

## Example

In the following example, enhanced editing mode is disabled on virtual terminal line 3:

```
line vty 3
no editing
```

## Related Command

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**terminal editing**††

# enable

To enter privileged EXEC mode, use the **enable** EXEC command.

**enable** [*level*]

## Syntax Description

| | |
|---|---|
| *level* | (Optional) Privilege level to log into on the router. |

## Command Mode

EXEC

## Usage Guidelines

Because many of the privileged commands set operating parameters, privileged access should be password-protected to prevent unauthorized use. If the system administrator has set a password with the **enable password** global configuration command, you are prompted to enter it before being allowed access to privileged EXEC mode. The password is case sensitive.

## Example

In the following example, the user enters the **enable** command and is prompted to enter a password. The password is not displayed on the screen. After entering the password, the system enters privileged command mode as indicated by the pound sign (#).

```
Router> enable
Password:
Router#
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**disable**
**enable password** †

# end

To exit configuration mode, use the **end** global configuration command.

**end**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Global configuration

## Usage Guidelines

You can also press Ctrl-Z to exit configuration mode.

## Example

In the following example, the router name is changed to *george* using the **hostname** global configuration command. Entering the **end** command causes the system to exit configuration mode and return to EXEC mode.

```
Router(config)# hostname alibaba
george(config)# end
george#
```

# exit

To exit any command mode or close an active terminal session and terminate the EXEC, use the **exit** command at the system prompt.

**exit**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Available in all command modes

## Usage Guidelines

When you enter the **exit** command at the EXEC levels, the EXEC mode is ended. Use the **exit** command at the configuration level to return to privileged EXEC mode. Use the **exit** command in interface, line, router, ipx-router, and route-map command modes to return to global configuration mode. Use the **exit** command in subinterface configuration mode to return to interface configuration mode. You can also press Ctrl-Z from any configuration mode to return to privileged EXEC mode.

## Examples

In the following example, the user exits subinterface configuration mode to return to interface configuration mode:

```
Router(config-subif)# exit
Router(config-if)#
```

The following example shows how to exit an active session.

```
Router> exit
```

## Related Commands

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**disconnect**††
**logout**††

# full-help

To get help for the full set of user-level commands, use the **full-help** command.

**full-help**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Available in all command modes.

## Usage Guidelines

The **full-help** command enables (or disables) an unprivileged user to see all of the help messages available. It is used with the **show ?** command.

## Example

The following example is output for **show ?** with **full-help** disabled:

```
Router> show ?
clock Display the system clock
history Display the session command history
hosts IP domain-name, lookup style, nameservers, and host table
sessionsInformation about Telnet connections
terminal Display terminal configuration parameters
users Display information about terminal lines
version System hardware and software status
```

## Related Commands

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**help**
**terminal full-help**††

# help

To display a brief description of the help system, enter the **help** command.

**help**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Available in all command modes

## Usage Guidelines

The **help** command provides a brief description of the context-sensitive help system.

- To list all commands available for a particular command mode, enter a question mark (?) at the **system prompt.**
- To obtain a list of commands that begin with a particular character string, enter the abbreviated command entry immediately followed by a question mark (?). This form of help is called word help, because it lists only the keywords or arguments that begin with the abbreviation you entered.
- To list a command's associated keywords or arguments, enter **a question mar**k (?) in pla**ce of a keyword or argument on the command line.** This form of help is called command syntax help, because it lists the keywords or arguments that apply based on the command, keywords, and arguments you have already entered.

## Examples

Enter the **help** command for a brief 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 backup 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?'.)
```

The following example shows how to use word help to display all the privileged EXEC commands that begin with the letters "co":

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

The following example shows how to use command syntax help to display the next argument of a partially complete **access-list** command. One option is to add a wildcard mask. The <cr> symbol indicates that the other option is to press Return to execute the command.

```
Router(config)# access-list 99 deny 131.108.134.234 ?
  A.B.C.D  Mask of bits to ignore
```

```
<cr>
```

## Related Commands

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**full-help**
**terminal full-help** ††

# history

To enable the command history function, or to change the command history buffer size for a particular line, use the **history** line configuration command. To disable the command history feature, use the **no** form of this command.

**history** [**size** *number-of-lines*]
**no history** [**size** *number-of-lines*]

## Syntax Description

| | |
|---|---|
| **size** *number-of-lines* | (Optional) Specifies the number of command lines that the system will record in its history buffer. The range is 0 to 256. |

## Default

10 lines

## Command Mode

Line configuration

## Usage Guidelines

The **history** command without the **size** keyword and the *number-of-lines* argument enables the history function with the last buffer size specified or with the default of 10 lines, if there was not a prior setting.

The **no history** command without the **size** keyword and the *number-of lines* argument disables the history feature but remembers the buffer size if it was something other than the default. The **no history size** command resets the buffer size to 10.

The command history feature provides a record of EXEC commands you have entered. This feature is particularly useful for recalling long or complex commands or entries, including access lists.

Table 2-3 lists the keys and functions you can use to recall commands from the command history buffer.

**Table 2-3 History Keys**

| Key | Function |
|---|---|
| Ctrl-P or Up Arrow[1] | Recalls commands in the history buffer in a backward sequence, beginning with the most recent command. Repeat the key sequence to recall successively older commands. |
| Ctrl-N or Down Arrow[1] | Returns to more recent commands in the history buffer after recalling commands with Ctrl-P or the Up Arrow. Repeat the key sequence to recall successively more recent commands. |

1. The arrow keys function only with ANSI-compatible terminals such as VT100s.

## Example

In the following example, virtual terminal line 4 is configured with a history buffer size of 35 lines:

```
line vty 4
history size 35
```

## Related Commands

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**show history**
**terminal history size**††

# show history

To list the commands you have entered in the current EXEC session, use the **show history** EXEC command.

**show history**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

The command history feature provides a record of EXEC commands you have entered. The number of commands the history buffer will record is determined by the **history size** line configuration command or the **terminal history size** EXEC command.

Table 2-4 lists the keys and functions you can use to recall commands from the command history buffer.

**Table 2-4 History Keys**

| Key | Function |
|---|---|
| Ctrl-P or Up Arrow | Recalls commands in the history buffer in a backward sequence, beginning with the most recent command. Repeat the key sequence to recall successively older commands. |
| Ctrl-N or Down Arrow | Returns to more recent commands in the history buffer after recalling commands with Ctrl-P or the Up Arrow. Repeat the key sequence to recall successively more recent commands. |

## Sample Display

The following is sample output from the **show history** command, which lists the commands the user has entered in EXEC mode for this session:

```
Router# show history
help
  where
  show hosts
  show history
Router#
```

## Related Commands

Two daggers (††) indicates that the command is documented in the *Cisco Access Connection Guide*.

**history size**
**terminal history size** ††

PART 2

# System and Interface Configuration and Management

CHAPTER 3

# System Image, Microcode Image, and Configuration File Load Commands

This chapter provides detailed descriptions of the commands used to load and copy system images, microcode images, and configuration files. Microcode images contain microcode to be downloaded to various hardware devices. System images contain the system software. Configuration files contain commands entered to customize the function of the router.

For router configuration information and examples, refer to the “Loading System Images, Microcode Images, and Configuration Files” chapter in the *Router Products Configuration Guide*.

# async-bootp

Use the **async-bootp** global configuration command to enable support for extended BOOTP requests as defined in RFC 1084 when the router is configured for SLIP. Use the **no async-bootp** global configuration command to restore the default.

**async-bootp** *tag* [*:hostname*] *data*
**no async-bootp**

## Syntax Description

| | |
|---|---|
| *tag* | Item being requested; expressed as filename, integer, or IP dotted-decimal address. See Table 3-1 for possible values. |
| *:hostname* | (Optional) This entry applies only to the host specified. The argument *:hostname* accepts both an IP address and a logical host name. |
| *data* | List of IP addresses entered in dotted-decimal notation or as logical host names, a number, or a quoted string. |

**Table 3-1 Async-BOOTP Tag Keywords**

| Keyword | Description |
|---|---|
| **bootfile** | Specifies use of a server boot file from which to download the boot program. Use the optional *:hostname* and *data* arguments to specify the filename. |
| **subnet-mask** *mask* | Dotted-decimal address specifying the network and local subnetwork mask (as defined by RFC 950). |
| **time-offset** *offset* | Signed 32-bit integer specifying the time offset of the local subnetwork in seconds from Universal Coordinated Time (UTC). |
| **gateway** *address* | Dotted-decimal address specifying the IP addresses of gateways for this subnetwork. A preferred gateway should be listed first. |
| **time-server** *address* | Dotted-decimal address specifying the IP address of time servers (as defined by RFC 868). |
| **IEN116-server** *address* | Dotted-decimal address specifying the IP address of name servers (as defined by IEN 116). |
| **DNS-server** *address* | Dotted-decimal address specifying the IP address of Domain Name Servers (as defined by RFC 1034). |
| **log-server** *address* | Dotted-decimal address specifying the IP address of an MIT-LCS UDP log server. |
| **quote-server** *address* | Dotted-decimal address specifying the IP address of Quote of the Day servers (as defined in RFC 865). |
| **lpr-server** *address* | Dotted-decimal address specifying the IP address of Berkeley UNIX Version 4 BSD servers. |
| **impress-server** *address* | Dotted-decimal address specifying the IP address of Impress network image servers. |
| **rlp-server** *address* | Dotted-decimal address specifying the IP address of Resource Location Protocol (RLP) servers (as defined in RFC 887). |

| Keyword | Description |
|---|---|
| **hostname** *name* | The name of the client, which may or may not be domain qualified, depending upon the site. |
| **bootfile-size** *value* | A two-octet value specifying the number of 512-octet (byte) blocks in the default boot file. |

## Default

If no extended BOOTP commands are entered, the router software generates a gateway and subnet mask appropriate for the local network.

## Command Mode

Global configuration

## Usage Guidelines

Use the EXEC command **show async-bootp** to list the configured parameters. Use the **no async-bootp** command to clear the list.

## Examples

The following example illustrates how to specify different boot files: one for a PC, and one for a Macintosh. With this configuration, a BOOTP request from the host on 128.128.1.1 results in a reply listing the boot filename as *pcboot*. A BOOTP request from the host named *mac* results in a reply listing the boot filename as *macboot*.

```
async-bootp bootfile :128.128.1.1 “pcboot”
async-bootp bootfile :mac "macboot"
```

The following example specifies a subnet mask of 255.255.0.0:

```
async-bootp subnet-mask 255.255.0.0
```

The following example specifies a negative time offset of the local subnetwork of -3600 seconds:

```
async-bootp time-offset -3600
```

The following example specifies the IP address of a time server:

```
async-bootp time-server 128.128.1.1
```

## Related Command

**show async-bootp**

# boot

To boot the router manually, use the **boot** ROM monitor command.

**boot**
**boot** *filename* [*ip-address*]
**boot flash** [*filename*]
**boot flash** [*device*:]*partition-number*:[*filename*]

## Syntax Description

| | |
|---|---|
| *filename* | Name of the system image you want to netboot. The filename is case sensitive. |
| *ip-address* | (Optional) IP address of the TFTP server on which the system image resides. If omitted, this value defaults to the IP broadcast address of 255.255.255.255. |
| **flash** *filename* | (Optional) Boots the router from Flash memory with the optional filename of the image you want loaded. The filename is case sensitive. Without *filename*, the first valid file in Flash memory is loaded. |
| *device*: | (Optional) Valid value is **flash**. |
| *partition-number*: | Boots the router from Flash memory with the optional filename of the image you want loaded from the specified Flash partition. If you do not specify a filename, the first valid file in the specified partition of Flash memory is loaded. |
| *filename* | (Optional) Boots the router from Flash memory with the filename of the image you want loaded from the specified Flash partition, if a partition is specified. If a partition is not specified, the system boots with the filename from the first partition. The filename is case sensitive. If you do not specify a filename, the first valid file in the specified partition of Flash memory is loaded. |

## Default

If you enter the **boot** command and press Return, the router boots from ROM by default.

If you enter the **boot flash** command without a *filename*, the first valid file in Flash memory is loaded.

For other defaults, see the Syntax Description section.

## Command Mode

ROM monitor

## Usage Guidelines

Use this command only when your router cannot find the configuration information needed in nonvolatile random-access memory (NVRAM). To get to the ROM monitor prompt (>), enter the **reload** EXEC command, and then press the Break key during the first 60 seconds of startup, or change the boot bits in the configuration register to zero, for manual booting, and then issue the **reload** command.

Refer to the *Cisco 7000 Hardware Installation and Maintenance* publication for the correct jumper settings for the Cisco 7000 series.

## Examples

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

```
> boot
F3:
(ROM Monitor copyrights)
```

In the following example, the file *routertest* is netbooted from IP address 131.108.15.112:

```
> boot routertest 131.108.15.112
F3:
(ROM Monitor copyrights)
```

The following example shows the **boot flash** command without the *filename* argument.The first valid file in Flash memory is loaded.

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

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

(ROM Monitor copyrights)
```

In the following example, the **boot flash** command is used with the filename *gs7-k*. That is the file that will be loaded.

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

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

(ROM Monitor copyrights)
```

In the following example, the **boot flash flash** command boots the relocatable image file *igs-bpx-l* from partition 2 in Flash memory.

```
> boot flash flash:2:igs-bpx-l
F3: 3562264+98228+303632 at 0x30000B4

(ROM Monitor copyrights)
```

# boot bootstrap

To configure the filename that is used to boot a secondary bootstrap image, use the **boot bootstrap** global configuration command. Use the **no boot bootstrap** command to disable booting from a secondary bootstrap image.

**boot bootstrap flash** [*filename*]
**no boot bootstrap flash** [*filename*]

**boot bootstrap mop** *filename [mac-address] [interface]*
**no boot bootstrap mop** *filename [mac-address] [interface]*

**boot bootstrap** [**tftp**] *filename* [*ip-address*]
**no boot bootstrap** [**tftp**] *filename* [*ip-address*]

## Syntax Description

| | |
|---|---|
| **flash** | Indicates that the router will be booted from Flash memory. |
| **mop** | Indicates that the router will be netbooted from a system image stored on a DEC MOP server. |
| **tftp** | (Optional) Indicates that the router will be netbooted from a system image stored on a TFTP server. |
| *filename* | (Optional with **flash**.) Name of the system image from which you want to netboot. If you omit the filename when booting from Flash, the router uses the first system image stored in Flash memory. |
| *ip-address* | (Optional) IP address of the TFTP server on which the system image resides. If omitted, this value defaults to the IP broadcast address of 255.255.255.255. |
| *mac-address* | (Optional) MAC address of the MOP server on which the file resides. If the MAC address argument is not included, a broadcast message is sent to all MOP boot servers. The first MOP server to indicate that it has the file will be the server from which the router gets the boot image. |
| *interface* | (Optional) Interface out which the router should send MOP requests to reach the MOP server. The interface options are **async**, **dialer**, **Ethernet**, **loopback**, **null**, **serial**, and **tunnel**. If the interface argument is not specified, a request will be sent on all interfaces that have MOP enabled, and the interface from which the first response is received will be used to load the software. |

## Default

No secondary bootstrap

## Command Mode

Global configuration

## Usage Guidelines

The **boot bootstrap** command, in conjunction with setting bit 9 on the configuration register of an AGS, CGS, or MGS router, causes the router to load a secondary bootstrap image over the network. The secondary bootstrap image then loads the specified system image file. The name of the secondary bootstrap file is boot-csc3 or boot-csc4, depending on the router model. See the appropriate hardware installation guide for details on the configuration register and secondary bootstrap filename.

Use this command when you have attempted to load a system image but have run out of memory even after compressing the system image. Secondary bootstrap allows you to load a larger system image through a smaller secondary image.

## Example

In the following example, the system image file *sysimage-2* will be loaded by using a secondary bootstrap image:

```
boot bootstrap sysimage-2
```

# boot buffersize

To modify the buffer size used to load configuration files, use the **boot buffersize** global configuration command. Use the **no boot buffersize** command to return to the default setting.

**boot buffersize** *bytes*
**no boot buffersize**

## Syntax Description

| | |
|---|---|
| *bytes* | Specifies the size of the buffer to be used. There is no minimum or maximum size that can be specified. |

## Default

Buffer size of the NVRAM

## Command Mode

Global configuration

## Usage Guidelines

Normally, the router uses a buffer the size of the system NVRAM to hold configuration commands read from the network. You can increase this size if you have a very complex configuration.

## Example

The following example sets the buffer size to 64000:

```
configure terminal
boot buffersize 64000
```

# boot host

To change the default name of the host configuration filename from which you want to load configuration commands, use the **boot host** global configuration command. Use the **no boot host** command to restore the host configuration filename to the default.

**boot host mop** *filename* [*mac-address*] [*interface*]
**no boot host mop** *filename* [*mac-address*] [*interface*]

**boot host** [**tftp** | **rcp**] *filename* [*ip-address*]
**no boot host** [**tftp** | **rcp**] *filename* [*ip-address*]

## Syntax Description

| | |
|---|---|
| **mop** | Indicates that the router will be configured from a configuration file stored on a DEC MOP server. |
| **tftp** | (Optional) Indicates that the router will be configured from a configuration file stored on a TFTP server. |
| **rcp** | (Optional) Indicates that the router will be configured from a configuration file stored on an rcp server. |
| *filename* | Name of the file from which you want to load configuration commands. |
| *ip-address* | (Optional) IP address of the TFTP server on which the file resides. If omitted, this value defaults to the IP broadcast address of 255.255.255.255. |
| *mac-address* | (Optional) MAC address of the MOP server on which the file resides. If the MAC address argument is not included, a broadcast message is sent to all MOP boot servers. The first MOP server to indicate that it has the file will be the server from which the router gets the boot image. |
| *interface* | (Optional) Interface out which the router should send MOP requests to reach the MOP server. The interface options are **async**, **dialer**, **ethernet**, **serial**, and **tunnel**. If the interface argument is not specified, a request will be sent on all interfaces that have MOP enabled, and the interface from which the first response is received will be used to load the software. |

## Default

The router uses its host name to form a host configuration filename. To form this name, the router converts its name to all lowercase letters, removes all domain information, and appends *-confg*.

## Command Mode

Global configuration

## Usage Guidelines

Use the **service config** command to enable the loading of the specified configuration file at reboot time. Without this command, the router ignores the **boot host** command and uses the configuration information in NVRAM. If the configuration information in NVRAM is invalid or missing, the **service config** command is enabled automatically.

The network server will attempt to load two configuration files from remote hosts. The first is the network configuration file containing commands that apply to all network servers on a network. The second is the host configuration file containing commands that apply to one network server in particular.

## Example

The following example sets the host filename to *wilma-confg* at address 192.31.7.19:

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

## Related Commands

**boot network**
**service config**

# boot network

To change the default name of the network configuration file from which you want to load configuration commands, use the **boot network** global configuration command. Use the **no boot network** command to restore the network configuration filename to the default.

**boot network mop** *filename* [*mac-address*] [*interface*]
**no boot network mop** *filename* [*mac-address*] [*interface*]

**boot network** [**tftp** | **rcp**] *filename* [*ip-address*]
**no boot network** [**tftp** | **rcp**] *filename* [*ip-address*]

## Syntax Description

| | |
|---|---|
| **mop** | Configures the router to download the configuration file from a network server using the Digital Maintenance Operation Protocol (MOP) protocol. |
| **tftp** | (Optional) Configures the router to download the configuration file from a network server using TFTP. If omitted and **rcp** is not specified, defaults to **tftp**. |
| **rcp** | (Optional) Configures the router to download the configuration file from a network server using rcp. If omitted, defaults to **tftp**. |
| *filename* | Name of the file from which you want to load configuration commands. The default filename is *network-config*. |
| *ip-address* | (Optional) If **rcp** or **tftp** is specified, the IP address of the network server on which the compressed image file resides. If the IP address is omitted, this value defaults to the IP broadcast address of 255.255.255.255. |
| *mac-address* | (Optional) If **MOP** is specified, the MAC address of the network server on which the file resides. If the MAC address argument is not included, a broadcast message is sent to all MOP boot servers. The first server to indicate that it has the file will be the server from which the router gets the boot image. |
| *interface* | (Optional) If MOP is specified, interface out which the router should send MOP requests to reach the server. The interface options are **async**, **dialer**, **ethernet**, **serial**, and **tunnel**. If the interface argument is not specified, a request will be sent on all interfaces that have MOP enabled, and the interface from which the first response is received will be used to load the software. |

## Default

The default filename is *network-config*. The default transfer protocol type is TFTP, if neither **tftp** nor **rcp** is specified.

## Command Mode

Global configuration

## Usage Guidelines

When booting from a network server, routers ignore routing information, static IP routes, and bridging information. As a result, intermediate routers are responsible for handling rcp or TFTP requests correctly. Before booting from a network server, verify that a server is available by using the **ping** command.

Use the **service config** command to enable the loading of the specified configuration file at reboot time. Without this command, the router ignores the **boot network** command and uses the configuration information in NVRAM. If the configuration information in NVRAM is invalid or missing, the **service config** command is enabled automatically.

The network server will attempt to load two configuration files from remote hosts. The first is the network configuration file containing commands that apply to all network servers on a network. Use the **boot network** command to identify the network configuration file.

The rcp software requires that a client send the remote username on each rcp request to the network server. When the **boot network rcp** command is executed, the router software sends the router host name as the both the remote and local usernames. The rcp implementation searches for the configuration files to be used relative to the account directory of the remote username on the network server, if the server has a directory structure, for example, as do UNIX systems.

If you copy the system image to a personal computer used as a file server, the remote host computer must support the remote shell (rsh) protocol.

## Examples

The following example changes the network configuration filename to *bridge_9.1* and uses the default broadcast address:

```
boot network bridge_9.1
service config
```

The following example changes the network configuration filename to *bridge_9.1*, specifies that rcp is to be used as the transport mechanism, and gives 131.108.1.111 as the IP address of the server on which the network configuration file resides:

```
boot network rcp bridge_9.1 131.108.1.111
service config
```

## Related Commands

**boot host**
**service config**

# boot system

To change the filename of the system image that is loaded onto the router at reboot time, use the **boot system** global configuration command. Use the **no boot system** command to remove the name.

**boot system flash** [*device***:**][*partition-number***:**][*filename*]
**no boot system flash** [*filename*]

**boot system mop** *filename* [*mac-address*] [*interface*]
**no boot system mop** *filename* [*mac-address*] [*interface*]

**boot system rom**
**no boot system rom**

**boot system** [**tftp** | **rcp**] *filename* [*ip-address*]
**no boot system** [**tftp** | **rcp**] *filename* [*ip-address*]

**no boot system**

## Syntax Description

| | |
|---|---|
| **flash** | Boots the router from Flash memory. |
| **mop** | Boots the router from a system image stored on a Digital MOP server. |
| **rom** | Boots the router from ROM. |
| **tftp** | (Optional) Boots the router from a system image stored on a TFTP server. |
| **rcp** | (Optional) Boots the router from a system image stored on a network server using rcp. If you omit this keyword, the transport mechanism defaults to **tftp**. |
| *filename* | (Optional with **flash**.) Name of the system image from which you want to netboot. It is case sensitive. |
| *mac-address* | (Optional) Media Access Control (MAC) address of the MOP server containing the specified system image file. If you do not include the MAC address argument, the system sends a broadcast message to all MOP boot servers. The first MOP server to indicate that it has the specified file will be the server from which the router gets the boot image. |
| *interface* | (Optional) Interface out which the router should send MOP requests to reach the MOP server. The interface options are **async**, **dialer**, **ethernet**, **serial**, and **tunnel**. If the interface argument is not specified, a request will be sent on all interfaces that have MOP enabled, and the interface from which the first response is received will be used to load the software. |
| *ip-address* | (Optional) IP address of the TFTP server on which the image file resides. If omitted, this value defaults to the IP broadcast address of 255.255.255.255. |

| | |
|---|---|
| *device*: | (Optional) Valid value is **flash**. |
| *partition-number*: | (Optional) Boots the router from Flash memory with the optional filename of the image you want loaded from the specified Flash partition. If you do not specify a filename, the first valid file in the specified partition of Flash memory is loaded. |

## Default

If you do not specify a system image file with the **boot system** command, the router uses the configuration register settings to determine the default system image filename for netbooting. The router forms the default boot filename by starting with the word *cisco* and then appending the octal equivalent of the boot field number in the configuration register, followed by a hyphen, and the processor type name (cisco*nn-cpu*). See the appropriate hardware installation guide for details on the configuration register and default filename. See also the command **config-register**. See also the Syntax Description section preceding this section.

## Command Mode

Global configuration

## Usage Guidelines

In order for this command to work, the **config-register** command must be set properly.

Enter several **boot system** commands to provide a fail-safe method for booting your router. Use the **boot system rom** command to specify use of the ROM system image as a backup to other **boot** commands in the configuration. The **boot system** commands are stored and executed on the order in which they are entered. If you enter multiple boot commands of the same type—for example, if you enter two commands that instruct the router to boot from different network servers—then the router tries them in the order they are entered.

Each time you write a new software image to Flash memory, you must delete the existing filename in the configuration file with the **no boot system flash** *filename* command. Then add a new line in the configuration file with the **boot system flash** *filename* command.

---

**Note** The **no boot system** global 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 command specified by these arguments.

---

You can netboot from a compressed image. When a server netboots software, the image being booted and the running image must both fit into memory. Use compressed images to ensure that there is enough available memory to boot the router. You can produce a compressed software image on any UNIX platform using the compress command. Refer to your UNIX platform's documentation for the exact usage of the **compress** command. (You can also uncompress data with the UNIX **uncompress** command.)

The rcp protocol requires that a client send the remote username on an rcp request to a server. When the **boot system rcp** command is executed, by default the router software sends the router host name as the both the remote and local usernames. The rcp software searches for the system image to be booted from the remote server relative to the directory of the remote username, if the server has a directory structure, for example, as do UNIX systems.

## Examples

The following example illustrates a list specifying two possible internetwork locations for a system image, with the ROM software being used as a backup:

```
boot system cs3-rx.90-1 192.31.7.24
boot system cs3-rx.83-2 192.31.7.19
boot system rom
```

The following example boots the system boot relocatable image file *igs-bpx-l* from partition 2 of the Flash device:

```
boot system flash flash:2:igs-bpx-l
```

## Related Commands

**config-register**
**copy flash rcp**
**copy flash tftp**
**copy rcp flash**
**copy tftp flash**
**ip rcmd remote-username**

# config-register

To change the router configuration register settings, use the **config-register** global configuration command.

**config-register** *value*

## Syntax Description

| | |
|---|---|
| *value* | Hexadecimal or decimal value that represents the 16-bit configuration register value you want to use the next time the router is restarted. The value range is from 0x0 to 0xFFFF (0 to 65535 in decimal). |

## Default

For the router models without Flash memory, the default is 0x101, which causes the router to boot from ROM and the Break key to be ignored. For router models with Flash memory, the default is 0x10F, which causes the router to boot from Flash memory and the Break key to be ignored.

## Command Mode

Global configuration

## Usage Guidelines

This command applies only to the Cisco 2000, Cisco 3000, Cisco 4000 series, or to the Cisco 7000 series. All other models use a hardware configuration register.

The lowest four bits of the configuration register (bits 3, 2, 1, and 0) form the boot field. The boot field determines if the router boots manually, from ROM, or from Flash or the network. Bit 8 controls the console Break key; when set to 1, it causes the Break key to be ignored. The remaining bits control other features of the router and are typically set to 0.

To change the boot field value and leave all other bits set to their default values, follow these guidelines:

- If you set the configuration register value to 0x100, you must boot the operating system manually with the **boot** command.
- If you set the configuration register value to 0x101, the router boots using the default ROM software.
- If you set the configuration register to any value from 0x102 to 0x10F, the router uses the boot field value to form a default boot filename for netbooting.

For more information about the configuration register bit settings and default filenames, see the appropriate router hardware installation guide.

## Example

In the following example, the configuration register is set to boot the system image from Flash memory:

```
config-register 0x010F
```

## Related Commands

**boot system**
**o**
**show version**

# configure

To enter global configuration mode, use the **configure** privileged EXEC command. You must be in global configuration mode to enter global configuration commands.

**configure** {**terminal** | **memory** | **network**}

## Syntax Description

| | |
|---|---|
| **terminal** | Executes configuration commands from the terminal. |
| **memory** | Executes the configuration commands stored in NVRAM. |
| **network** | Retrieves the configuration commands stored in a file on a server. |

## Default

None

## Command Mode

Privileged EXEC

## Usage Guidelines

If you do not specify **terminal**, **memory**, or **network**, the router prompts you for the source of configuration commands. After you enter the **configure** command, the system prompt changes from <router-name># to <router-name>(config)#, indicating that you are in global configuration mode. To leave global configuration mode and return to the privileged EXEC prompt, press **Ctrl-Z**.

---

**Note** The commands **configure net network** and **configure net host** no longer clear line parameters.

---

## Examples

In the following example, the router is configured from the terminal:

```
Router# configure

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

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

```
Router1# configure network

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

## Related Commands

**show configuration**
**write memory**
**write terminal**

# configure overwrite-network

To load a configuration file directly into NVRAM, use the **configure overwrite-network** privileged EXEC command.

**configure overwrite-network**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Privileged EXEC

## Usage Guidelines

Use caution when entering the filename, because this command is not run through the parser. Also be careful not to load a file that is larger than NVRAM.

This command is useful if you are running an older version of software and are going to upgrade to a new Cisco Internetwork Operating System (Cisco IOS) release. For example, if you have Software Release 9.1 ROMs, you can save time by loading a Cisco IOS Release 10.2 configuration file before you get the Release 10.2 software. That way, you will be ready to reboot when you receive the Release 10.2 software image.

This command also allows you to replace an entire old configuration, and ensure that none of the old configuration will remain.

## Example

The following example directly loads the host configuration file *doc-ags+1-confg* from a remote host into NVRAM:

```
doc-ags+1# configure overwrite-network
Host or network configuration file [host]?
Address of remote host [255.255.255.255]?
Name of configuration file [doc-ags+1-confg]?
Configure using doc-ags+1-confg from 255.255.255.255? [confirm]
Loading doc-ags+1-confg...
```

# continue

To return to the EXEC mode from ROM monitor mode, use the **continue** ROM monitor command.

**continue**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

ROM monitor

## Usage Guidelines

Use this command when you are in ROM monitor mode, and you want to return to EXEC mode to use the system image instead of reloading.

## Example

In the following example, the **continue** command takes you from ROM monitor to EXEC mode:

```
> continue
Router#
```

# copy bootflash rcp

To use rcp to copy a bootstrap image from Flash memory on a Cisco 4500 router to a network server, use the **copy bootflash rcp** EXEC command.

**copy bootflash rcp**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

This command is supported on the Cisco 4500 router only. The copy of the bootstrap image can serve as a backup copy and also can be used to verify that the copy in Flash memory is the same as the original file on disk.

The rcp protocol requires that a client send the remote username of an rcp request to the server. When you issue the **copy bootflash rcp** command, by default the router software sends the username associated with the current TTY, if that name is valid. For example, if the user is connected to the router through Telnet and the user was authenticated through the **username** command, then the router software sends that username as the remote username.

---

**Note** For Cisco, TTYs are commonly used in communication servers. The concept of TTYs originated with UNIX. For UNIX systems, each physical device is represented in the file system. Terminals are called TTY devices (which stands for teletype, the original UNIX terminal).

---

If the TTY username is invalid, the router software uses the router host name as the both the remote and local usernames. To specify a different remote username to be sent to the rcp server, use the **ip rcmd remote-username** command.The rcp software copies the bootstrap image to an appropriate remote server. For example, if the server has a directory structure as do UNIX systems, the bootstrap image is copied to the remote server relative to the directory of the remote username.

**Caution** The remote username must be associated with an account on the destination server. If you do not use the **ip rcmd remote-username** command to specify the name of a remote user associated with an account on the server, then the remote username associated with the current TTY process must be associated with an account on the server. If there is no username for the current TTY process, then the router host name must be associated with an account on the server. If the network administrator of the destination server did not establish accounts for the remote username used, this command will not execute successfully.

If you copy the bootstrap image to a personal computer used as a file server, the remote host computer must support rcp.

## Example

The following example shows how to use this command on a Cisco 4500 router:

```
Router(config)# ip rcmd remote-username netadmin1
Router# copy bootflash rcp

System flash directory, partition 2:
File  Length   Name/status
  1   984      junk
[1048 bytes used, 8387560 available, 8388608 total]
Address or name of remote host [223.255.254.254]?
Source file name? junk
Destination file name [junk]? junk
Verifying checksum for 'junk' (file # 1)...  OK
Copy 'junk' from Flash to server
  as 'junk'? [yes/no]y
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Upload to server done
Flash copy took 0:00:00 [hh:mm:ss]
```

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

## Related Commands

**copy rcp bootflash**
**ip rcmd remote-username**

# copy bootflash tftp

On the Cisco 4500, to copy a boot image from Flash memory to a TFTP server, use the **copy bootflash tftp** EXEC command.

**copy bootflash tftp**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

You might want to copy the boot image in order to save a back-up copy of it or to verify that the copy in Flash is the same as on the original file.

## Example

The following example illustrates how to use this command:

```
Router# copy bootflash tftp

System flash directory, partition 2:
File  Length   Name/status
  1   984      junk
[1048 bytes used, 8387560 available, 8388608 total]
Address or name of remote host [223.255.254.254]?
Source file name? junk
Destination file name [junk]? junk
Verifying checksum for 'junk' (file # 1)...  OK
Copy 'junk' from Flash to server
  as 'junk'? [yes/no]y
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Upload to server done
Flash copy took 0:00:00 [hh:mm:ss]
```

## Related Commands

**copy mop bootflash**
**copy tftp bootflash**
**copy verify bootflash**
**erase bootflash**
**show bootflash**

# copy flash rcp

To copy a system image from Flash memory to a network server using rcp, use the **copy flash rcp** EXEC command.

**copy flash rcp**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

You can use the copy of the system image as a backup copy. You can also use it to verify that the copy in Flash memory is the same as on the original file on disk.

The rcp software requires that a client send the remote username on each rcp request to the server. When you issue the **copy flash rcp** command, by default the router software sends the remote username associated with the current TTY, if that name is valid. For example, if the user is connected to the router through Telnet and the user was authenticated through the **username** command, then the router software sends that username as the remote username.

If the TTY username is invalid, the router software uses the router host name as the both the remote and local usernames.

---

**Note** For Cisco, TTYs are commonly used in communications servers. The concept of TTYs originated with UNIX. For UNIX systems, each physical device is represented in the file system. Terminals are called TTY devices (which stands for teletype, the original UNIX terminal).

---

To specify a different remote username to be sent to the server, use the **ip rcmd remote-username** command. The rcp software copies the system image to the remote server relative to the directory of the remote username, if the server has a directory structure, for example, as do UNIX systems.

**Caution** The remote username must be associated with an account on the destination server. If you do not use the **ip rcmd remote-username** command to specify the name of a remote user associated with an account on the server, then the remote username associated with the current TTY process must be associated with an account on the server. If there is no username for the current TTY process, then the router host name must be associated with an account on the server. If the network administrator of the destination server did not establish accounts for the remote username used, this command will not execute successfully if a default remote username is used.

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

## Examples

The following example shows how to use this command on a Cisco 4500 router:

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

System flash directory, partition 2:
File  Length   Name/status
  1   984      junk
[1048 bytes used, 8387560 available, 8388608 total]
Address or name of remote host [223.255.254.254]?
Source file name? junk
Destination file name [junk]? junk
Verifying checksum for 'junk' (file # 1)...  OK
Copy 'junk' from Flash to server
  as 'junk'? [yes/no]y
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Upload to server done
Flash copy took 0:00:00 [hh:mm:ss]
```

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

The following example illustrates how to use this command:

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

The following example illustrates how to use this command when copying from a particular partition of Flash memory:

```
Router# copy flash rcp
System flash partition information:
Partition   Size     Used    Free    Bank-Size   State       Copy-Mode
    1       4096K    2048K   2048K   2048K       Read Only   RXBOOT-FLH
    2       4096K    2048K   2048K   2048K       Read/Write  Direct

[ Type ?<number> for partition directory; ? for full directory; q to abort]
Which partition? [default = 1]
```

The system will prompt if there are two or more partitions. If the partition entered is not valid, the process terminates. You have the option to enter a partition number, **?** for directory display of all partitions, or **?***number* for directory display of a particular partition. The default is the first partition.

```
System flash directory, partition 2:
File  Length   Name/status
  1   3459720  master/igs-bfpx.100-4.3
[3459784 bytes used, 734520 available, 4194304 total]
Address or name of remote host [ABC.CISCO.COM]?
Source file name?
```

The file will be copied from the partition given by the user earlier:

```
Destination file name [default = source name]?
Verifying checksum for 'master/igs-bfpx.100-4.3' (file # 1)...  OK
Copy 'master/igs-bfpx.100-4.3' from Flash to server
as 'master/igs-bfpx.100-4.3'? [yes/no] yes
```

## Related Commands

**boot system flash**
**copy rcp flash**
**ip rcmd remote-username**

# copy flash tftp

To copy a system image from Flash memory to a TFTP server, use the **copy flash tftp** EXEC command.

**copy flash tftp**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

You can use the copy of the system image as a backup copy. You can also use it to verify that the copy in Flash memory is the same as on the original file on disk.

## Examples

The following example illustrates how to use this command:

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

The following example illustrates how to use this command when copying from a particular partition of Flash memory:

```
Router# copy flash tftp
System flash partition information:
Partition   Size     Used    Free    Bank-Size   State       Copy-Mode
    1       4096K    2048K   2048K   2048K       Read Only   RXBOOT-FLH
    2       4096K    2048K   2048K   2048K       Read/Write  Direct

[ Type ?<number> for partition directory; ? for full directory; q to abort]
Which partition? [default = 1]
```

The system will prompt if there are two or more partitions. If the partition entered is not valid, the process terminates. You can enter a partition number, **?** for directory display of all partitions, or **?***number* for directory display of a particular partition. The default is the first partition.

```
System flash directory, partition 2:
File  Length   Name/status
  1   3459720  master/igs-bfpx.100-4.3
[3459784 bytes used, 734520 available, 4194304 total]
Address or name of remote host [ABC.CISCO.COM]?
Source file name?
```

The file will be copied from the partition given by the user earlier:

```
Destination file name [default = source name]?
Verifying checksum for 'master/igs-bfpx.100-4.3' (file # 1)...  OK
Copy 'master/igs-bfpx.100-4.3' from Flash to server
as 'master/igs-bfpx.100-4.3'? [yes/no] yes
```

## Related Commands

**boot system flash**
**copy tftp flash**

# copy mop bootflash

To copy a boot image from a MOP server to Flash memory on the Cisco 4500, use the **copy mop bootflash** EXEC command.

**copy mop bootflash**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

The router prompts for the name of the image file. It provides an option to erase the existing boot image in Flash before writing the new image into Flash. If no free space is available, or if files have never been written to Flash memory, you must erase Flash memory before copying the MOP image.

You do not need to specify the address of a MOP server. The router automatically solicits a MOP boot server for the specified file by sending a multicast file-request message.

The copying process takes several minutes; the actual time differs from network to network.

Before booting from Flash memory, verify that the checksum of the image in Flash memory matches the checksum listed in the README file that was distributed with the boot software image. The checksum of the boot image in Flash memory is displayed when the **copy mop bootflash** command completes. The README file was copied to the MOP server automatically when you installed the boot software image.

**Caution** If the checksum values do not match, do not reboot the router. Instead, reissue the **copy mop bootflash** command and compare the checksums again. If the checksum is repeatedly wrong, copy the original boot software image back into Flash memory *before* you reboot the router from Flash memory.

## Example

The following example shows how to use this command to copy the boot image *c4500-k*:

```
Router# copy mop bootflash

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

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

Copy 'junk' from server
  as 'junk' into Flash WITH erase? [yes/no]yes
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
```

```
Loading junk from 1234.5678.9abc via Ethernet0: !
[OK - 984/8388608 bytes]

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

## Related Commands

**copy bootflash tftp**
**copy tftp bootflash**
**copy verify bootflash**
**erase bootflash**
**show bootflash**

# copy mop flash

To copy a system image using MOP into Flash memory, use the **copy mop flash** EXEC command.

**copy mop flash**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

MOP must be enabled on the relevant interfaces before you can use this command.

The router prompts for the MOP filename. It provides an option to erase existing Flash memory before writing onto it. The entire copying process takes several minutes and will differ from network to network.

Before booting from Flash memory, verify that the checksum of the image in Flash memory matches the checksum listed in the README file that was distributed with the system software image. The checksum of the image in Flash memory is displayed at the bottom of the screen when you issue the **copy mop flash** command.

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

## Examples

The following example shows a sample output of the **copy mop flash** command. In this example, the system image *c4500-k*, which already exists in Flash memory, is copied to Flash memory, and there is enough memory to copy the file without erasing any existing files.

```
Router# copy mop flash

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

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

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

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

The following example shows sample output of copying a system image into a partition of Flash memory. The system will prompt only if there are two or more read/write partitions or one read-only and one read/write partition and dual Flash bank support in boot ROMs. If the partition entered is not valid, the process terminates. You can enter a partition number, **?** for directory display of all partitions, or **?***number* for directory display of a particular partition. The default is the first read/write partition.

```
Router# copy mop flash
System flash partition information:
Partition   Size     Used    Free    Bank-Size   State       Copy-Mode
    1       4096K    2048K   2048K   2048K       Read Only   RXBOOT-FLH
    2       4096K    2048K   2048K   2048K       Read/Write  Direct

[ Type ?<no> for partition directory; ? for full directory; q to abort]
Which partition? [default = 2]
```

If the partition is read-only and has dual Flash bank support in boot ROMs, the session continues as follows:

```
                               ****  NOTICE  ****
Flash load helper v1.0
This process will accept the copy options and then terminate
the current system image to use the ROM based image for the copy.
Routing 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.
                               ---- ******** ----
Proceed? [confirm]
System flash directory, partition 1:
File  Length   Name/status
  1   3459720  master/igs-bfpx.100-4.3
[3459784 bytes used, 734520 available, 4194304 total]
Source file name? master/igs-bfpx-100.4.3
Destination file name [default = source name]?
```

The file will be copied into the partition given by the user earlier:

```
Loading master/igs-bfpx.100-4.3 from 131.108.1.111: !
Erase flash device before writing? [confirm]
Flash contains files. Are you sure? [confirm]
Copy 'master/igs-bfpx.100-4.3' from MOP server
as 'master/igs-bfpx.100-4.3' into Flash WITH erase? [yes/no] yes
```

If the partition is read-write, the session continues as follows:

```
System flash directory, partition 2:
File  Length   Name/status
  1   3459720  master/igs-bfpx.100-4.3
[3459784 bytes used, 734520 available, 4194304 total]
Source file name? master/igs-bfpx.100-4.3
Destination file name [default = source name]?
```

The file will be copied into the partition given by the user earlier:

```
Loading master/igs-bfpx.100-4.3 from 131.108.1.111: !
Erase flash device before writing? [confirm]
Flash contains files. Are you sure? [confirm]
Copy 'master/igs-bfpx.100-4.3' from MOP server
as 'master/igs-bfpx.100-4.3' into Flash WITH erase? [yes/no] yes
```

## Related Commands

**boot system flash**
**copy flash tftp**
**copy verify**

# copy rcp bootflash

To copy a bootstrap image from a network server to Flash memory on a Cisco 4500 router using rcp, use the **copy rcp bootflash** EXEC command.

**copy rcp bootflash**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

Use this command to copy a bootstrap image to Flash memory on a Cisco 4500 router. The router prompts for the name or address of the server and the name of the file to be copied. It provides an option to erase existing Flash memory before writing onto it, and allows you to confirm the erasure. The entire copying process takes several minutes and will differ from network to network.

Before loading the router from Flash memory, verify that the checksum of the bootstrap image in Flash memory matches the checksum listed in the README file that was distributed with the system software image.

The checksum of the bootstrap image in Flash memory is displayed at the bottom of the screen when you issue the **copy rcp bootflash** command. The README file was copied to the server automatically when you installed the system software.

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

The rcp protocol requires that a client send the remote username of an rcp request to the server. When you issue the **copy rcp bootflash** command, by default the router software sends the username associated with the current TTY, if that name is valid. For example, if the user is connected to the router through Telnet and the user was authenticated through the **username** command, then the router software sends that username as the remote username.

If the TTY username is invalid, the router software uses the router host name as the both the remote and local usernames.

**Note** For Cisco, TTYs are commonly used in communication servers. The concept of TTYs originated with UNIX. For UNIX systems, each physical device is represented in the file system. Terminals are called TTY devices (which stands for teletype, the original UNIX terminal).

To specify a different remote username to be sent to the rcp server, use the **ip rcmd remote-username** command. The rcp software searches for the bootstrap image to copy from the remote server relative to the directory of the remote username, if the server has a directory structure, for example, as do UNIX systems.

**Caution** The remote username must be associated with an account on the destination server. If you do not use the **ip rcmd remote-username** command to specify the name of a remote user associated with an account on the server, then the router host name must be associated with an account on the server. If the network administrator of the destination server did not establish accounts for the remote username used, this command will not execute successfully if a default remote username is used.

If you copy the bootstrap image from a personal computer used as a file server, the computer must support the rsh protocol.

## Example

The following example shows how to use this command on a Cisco 4500 router:

```
Router# configure terminal
Router(config)# ip rcmd remote-username netadmin1
Ctrl-Z
Router# copy rcp bootflash

Boot flash directory:
File  Length   Name/status
  1   2622607  c4500-xboot
[2622672 bytes used, 1571632 available, 4194304 total]

Address or name of remote host [255.255.255.255]? 223.255.254.254
Source file name? c4500-xboot.101
Destination file name [c4500-xboot.101]?
Accessing file 'c4500-xboot.101' on 223.255.254.254...
Loading c4500-xboot.101 from 223.255.254.254 (via Ethernet0): ! [OK]

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

Copy 'c4500-xboot.101' from TFTP server into
     bootflash as 'c4500-xboot.101' WITH erase? [yes/no] yes
Erasing device... eeeeeeeeeeeeeeee ...erased
Loading c4500-xboot.101 from 223.255.254.254 (via Ethernet0): !!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 2622607/4194304 bytes]

Verifying checksum...  OK (0xE408)
Flash copy took 0:00:10 [hh:mm:ss]
```

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

## Related Commands

**boot system flash**
**copy flash rcp**
**ip rcmd remote-username**

# copy rcp flash

To copy a system image from a network server into Flash memory using rcp, use the **copy rcp flash** EXEC command.

**copy rcp flash**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

The router prompts for the address of the rcp server and rcp filename. It provides an option to erase existing Flash memory before writing onto it. The entire copying process takes several minutes and will differ from network to network.

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

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

The rcp protocol requires that a client send the remote username of an rcp request to the server. When you issue the **copy rcp flash** command, by default the router software sends the username associated with the current TTY, if that name is valid. For example, if the user is connected to the router through Telnet and the user was authenticated through the **username** command, then the router software sends that username as the remote username.

**Note** For Cisco, TTYs are commonly used in communications servers. The concept of TTYs originated with UNIX. For UNIX systems, each physical device is represented in the file system. Terminals are called TTY devices (which stands for teletype, the original UNIX terminal).

If the TTY username is invalid, the router software uses the router host name as the both the remote and local usernames. To specify a different remote username to be sent to the rcp server, use the **ip rcmd remote-username** command. The rcp software copies the system image from the remote server relative to the directory of the remote username, if the server has a directory structure, for example, as do UNIX systems.

**Caution** The remote username must be associated with an account on the destination server. If you do not use the **ip rcmd remote-username** command to specify the name of a remote user associated with an account on the server, then the remote username associated with the current TTY process must be associated with an account on the server. If there is no username for the current TTY process, then the router host name must be associated with an account on the server. If the network administrator of the destination server did not establish accounts for the remote username used, this command will not execute successfully if a default remote username is used.

If you copy the system image from a personal computer used as a file server, the remote host computer must support rcp.

## Examples

The following example shows how to use this command on a Cisco 4500 system. The interface might differ slightly on other systems. This example copies a system image named *IJ09140z* from the netadmin1 directory on the remote server named *SERVER1.CISCO.COM* with an IP address of 131.108.101.101 to the router's Flash memory. To ensure that enough Flash memory is available to accommodate the system image to be copied, the router software allows you to erase the contents of Flash memory first.

```
Router1# configure terminal

Router1(config)# rcmd remote-username netadmin1
Ctrl-Z
Router# copy rcp flash

System flash directory, partition 2:
File  Length   Name/status
  1   984      junk [deleted]
  2   984      junk
[2096 bytes used, 8386512 available, 8388608 total]
Address or name of remote host [255.255.255.255]? 223.255.254.254
Source file name? junk
Destination file name [junk]?
Accessing file 'junk' on 223.255.254.254...
Loading dirt/ssangiah/junk .from 223.255.254.254 (via Ethernet0): ! [OK]

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

Copy 'junk' from server
  as 'junk' into Flash WITH erase? [yes/no]yes
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
Loading junk from 223.255.254.254 (via Ethernet0): !
[OK - 984/8388608 bytes]

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

The following example shows sample output when copying a system image into a partition of Flash memory. The system prompts only if there are two or more read/write partitions or one read-only and one read/write partition and dual Flash bank support in boot ROMs. If the partition entered is not valid, the process terminates. You can enter a partition number, **?** for directory display of all partitions, or **?***number* for directory display of a particular partition. The default is the first read/write partition.

```
Router# copy rcp flash

System flash partition information:
```

```
Partition   Size     Used    Free    Bank-Size   State        Copy-Mode
    1       4096K    2048K   2048K   2048K       Read Only    RXBOOT-FLH
    2       4096K    2048K   2048K   2048K       Read/Write   Direct

[ Type ?<no> for partition directory; ? for full directory; q to abort]
Which partition? [default = 2]
```

If the partition is read-only and has dual Flash bank support in boot ROM, the session continues as follows:

```
                               ****  NOTICE  ****
Flash load helper v1.0
This process will accept the copy options and then terminate
the current system image to use the ROM based image for the copy.
Routing 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.
                               ---- ******** ----
Proceed? [confirm]
System flash directory, partition 1:
File  Length   Name/status
  1   3459720  master/igs-bfpx.100-4.3
[3459784 bytes used, 734520 available, 4194304 total]
Address or name of remote host [255.255.255.255]? 131.108.1.1
Source file name? master/igs-bfpx-100.4.3
Destination file name [default = source name]?
```

The file will be copied into the partition given by the user earlier:

```
Loading master/igs-bfpx.100-4.3 from 131.108.1.111: !
Erase flash device before writing? [confirm]
Flash contains files. Are you sure? [confirm]
Copy 'master/igs-bfpx.100-4.3' from TFTP server
as 'master/igs-bfpx.100-4.3' into Flash WITH erase? [yes/no] yes
```

If the partition is read-write, the session continues as follows:

```
System flash directory, partition 2:
File  Length   Name/status
  1   3459720  master/igs-bfpx.100-4.3
[3459784 bytes used, 734520 available, 4194304 total]
Address or name of remote host [255.255.255.255]? 131.108.1.1
Source file name? master/igs-bfpx.100-4.3
Destination file name [default = source name]?
```

The file will be copied into the partition given by the user earlier:

```
Accessing file 'master/igs-bfpx.100-4.3' on ABC.CISCO.COM...
Loading master/igs-bfpx.100-4.3 from 131.108.1.111: !
Erase flash device before writing? [confirm]
Flash contains files. Are you sure? [confirm]
Copy 'master/igs-bfpx.100-4.3' from TFTP server
as 'master/igs-bfpx.100-4.3' into Flash WITH erase? [yes/no] yes
```

## Related Commands

**boot system flash**
**copy flash rcp**
**ip rcmd remote-username**
**copy verify**

# copy rcp running-config

To use rcp to copy a configuration file from a network server to the router, then run that configuration, use the **copy rcp running-config** EXEC command.

**copy rcp running-config**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

Use this command to copy either a host configuration file or a network configuration file from a remote server to the router using rcp, load the configuration file into RAM, and run it on the router.

The router software allows you to specify the type of configuration file to be copied. Accept the default value of *host* to copy and load a host configuration file containing commands that apply to one network server in particular. Enter a value for *network* to copy and load a network configuration file containing commands that apply to all network servers on a network.

The rcp protocol requires that a client send the remote username of an rcp request to the server. When you issue the **copy rcp running-config** command, by default the router software sends the username associated with the current TTY, if that name is valid. For example, if the user is connected to the router through Telnet and the user was authenticated through the **username** command, then the router software sends that username as the remote username.

If the TTY username is invalid, the router software uses the router host name as the both the remote and local usernames. To specify a different remote username to be sent to the rcp server, use the **ip rcmd remote-username** command. The rcp protocol copies the configuration file from the remote server relative to the directory of the remote username, if the server has a directory structure, for example, as do UNIX systems.

The remote username must be associated with an account on the destination server. If you do not use the **ip rcmd remote-username** command to specify the name of a remote user associated with an account on the server, then the remote username associated with the current TTY process must be associated with an account on the server.

If there is no username for the current TTY process, then the router host name must be associated with an account on the server. If the network administrator of the destination server did not establish accounts for the remote username used, this command will not execute successfully if a default remote username is used.

---

**Note** For Cisco, TTYs are commonly used in communication servers. The concept of TTYs originated with UNIX. For UNIX systems, each physical device is represented in the file system. Terminals are called TTY devices (which stands for teletype, the original UNIX terminal).

---

**Caution** If you copy the configuration file from a personal computer used as a file server, the computer must support the rsh protocol.

## Example

The following example shows how to use this command on a Cisco 4500 system. The interface might differ slightly on other systems. This example specifies a remote username of *netadmin1*. Then it copies and runs a host configuration file name *host1-confg* from the *netadmin1* directory on the remote server with an IP address of 131.108.101.101.

```
Router# configure terminal
Router(config)# ip rcmd remote-username netadmin1
Ctrl-Z
Router# copy rcp running-config

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

## Related Commands

**copy running-config rcp**
**ip rcmd remote-username**

# copy rcp startup-config

To copy a configuration file from a network server to the router's NVRAM using rcp, use the **copy rcp startup-config** EXEC command.

**copy rcp startup-config**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

Use this command to retrieve the commands stored in a configuration file on a server and write them to a file of the same name stored in NVRAM on the router.

The router software allows you to specify the type of configuration file to be copied. Accept the default value of *host* to copy and store a host configuration file containing commands that apply to one network server in particular. Enter *network* to copy and store a network configuration file containing commands that apply to all network servers on a network.

The rcp protocol requires that a client send the remote username of an rcp request to the server. When you issue the **copy rcp startup-config** command, by default the router software sends the username associated with the current TTY, if that name is valid. For example, if the user is connected to the router through Telnet and the user was authenticated through the **username** command, then the router software sends that username as the remote username.

**Note** For Cisco, TTYs are commonly used in communication servers. The concept of TTYs originated with UNIX. For UNIX systems, each physical device is represented in the file system. Terminals are called TTY devices (which stands for teletype, the original UNIX terminal).

To specify a different remote username to be sent to the server, use the **ip rcmd remote-username** command. The rcp software copies the configuration file from the remote server relative to the directory of the remote username, if the server has a directory structure, for example, as do UNIX systems.

**Caution** The remote username must be associated with an account on the destination server. If you do not use the **ip rcmd remote-username** command to specify the name of a remote user associated with an account on the server, then the remote username associated with the current TTY process must be associated with an account on the server. If there is no username for the current TTY process, then the router host name must be associated with an account on the server. If the network administrator of the destination server did not establish accounts for the remote username used, this command will not execute successfully if a default remote username is used.

If you copy the configuration file from a personal computer used as a file server, the PC must support the remote shell (rsh) protocol.

## Example

The following example shows how to use this command on a Cisco 4000 system. The interface might differ slightly on other systems. This example specifies a remote username of *netadmin1*. Then it copies and stores a host configuration file *host2-confg* from the netadmin1 directory on the remote server with an IP address of 131.108.101.101.

```
Router# configure terminal
Router(config)# ip rcmd remote-username netadmin1
Ctrl-Z
Router# copy rcp startup-config

Host or network configuration file [host]?
Address of remote host [255.255.255.255]? 131.108.101.101
Name of configuration file[rtr2-confg]? host2-confg
Configure using rtr2-confg from 131.108.101.101?[confirm]
Connected to 131.108.101.101
Loading 1112 byte file rtr2-confg:![OK]
[OK]
Router#
%SYS-5-CONFIG_NV:Non-volatile store configured from rtr2-config by
rcp from 131.108.101.101
```

## Related Commands

**copy startup-config rcp**
**ip rcmd remote-username**

# copy running-config

To copy the running configuration file from the router to a network server using rcp or TFTP, use the **copy running-config** EXEC command.

**copy running-config** {**rcp** | **tftp**}

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

This command copies the current configuration file to a server on the network. The copy of the configuration file can serve as a backup copy. You are prompted for a destination host and filename.

The rcp protocol requires that a client send the remote username of an rcp request to the server. When you issue the **copy running-config-rcp** command, by default the router software sends the username associated with the current TTY, if that name is valid. For example, if the user is connected to the router through Telnet and the user was authenticated through the **username** command, then the router software sends that username as the remote username.

If the TTY username is invalid, the router software uses the router host name as the both the remote and local usernames.

---

**Note** For Cisco, TTYs are commonly used in communication servers. The concept of TTYs originated with UNIX. For UNIX systems, each physical device is represented in the file system. Terminals are called TTY devices (which stands for teletype, the original UNIX terminal).

---

To specify a different remote username to be sent to the server, use the **ip rcmd remote-username** command. The rcp software copies the running configuration file to the remote server relative to the directory of the remote username that you specify, if the server has a directory structure, for example, as do UNIX systems.

**Caution** The remote username must be associated with an account on the destination server. If you do not use the **ip rcmd remote-username** command to specify the name of a remote user associated with an account on the server, then the remote username associated with the current TTY process must be associated with an account on the server. If there is no username for the current TTY process, then the router host name must be associated with an account on the server. If the network administrator of the destination server did not establish accounts for the remote username used, this command will not execute successfully if a default remote username is used.

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

To run this command, the router must contain Flash memory.

## Example

The following example shows how to use this command on a Cisco 4500 system. The interface may differ slightly on other systems. This example specifies a remote username of *netadmin1*. Then it copies the running configuration file, named *Rtr2-confg* to the *netadmin1* directory on the remote host with an IP address of 131.108.101.101.

```
Router# configure terminal
Router(config)# ip rcmd remote-username netadmin1
Ctrl-Z
Router# copy running-config rcp
Remote host[]? 131.108.101.101

Name of configuration file to write [Rtr2-confg]?
Write file rtr2-confg on host 131.108.101.101?[confirm]
###![OK]
Connected to 131.108.101.101
```

## Related Commands

**copy rcp running-config**
**ip rcmd remote-username**

# copy startup-config

To copy a startup configuration file to a network server using rcp or TFTP, use the **copy startup-config** EXEC command.

**copy startup-config** {**rcp** | **tftp**}

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

Use this command to copy the contents of the configuration file in NVRAM to a network server.

The rcp protocol requires that a client send the remote username of an rcp request to the server. When you issue the **copy startup-config rcp** command, by default the router software sends the username associated with the current TTY, if that name is valid. For example, if the user is connected to the router through Telnet and the user was authenticated through the **username** command, then the router software sends that username as the remote username.

If the TTY username is invalid, the router software uses the router host name as the both the remote and local usernames.

---

**Note** For Cisco, TTYs are commonly used in communication servers. The concept of TTYs originated with UNIX. For UNIX systems, each physical device is represented in the file system. Terminals are called TTY devices (which stands for teletype, the original UNIX terminal).

---

To specify a different remote username to be sent to the server, use the **ip rcmd remote-username** command. The rcp software copies the system image to the remote server relative to the directory of the remote username, if the server has a directory structure, for example, as do UNIX systems.

**Caution** The remote username must be associated with an account on the destination server. If you do not use the **ip rcmd remote-username** command to specify the name of a remote user associated with an account on the server, then the remote username associated with the current TTY process must be associated with an account on the server. If there is no username for the current TTY process, then the router host name must be associated with an account on the server. If the network administrator of the destination server did not establish accounts for the remote username used, this command will not execute successfully if a default remote username is used.

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

## Example

The following example shows how to use this command on a Cisco 4500 router. The interface might differ slightly on other systems.

```
Router# configure terminal
Router(config)# ip rcmd remote-username netadmin2
Ctrl-Z
Router# copy startup-config rcp
Remote host[]? 131.108.101.101

Name of configuration file to write [rtr2-confg]? <cr>
Write file rtr2-confg on host 131.108.101.101?[confirm] <cr>
![OK]
```

## Related Commands

**copy rcp startup-config**
**ip rcmd remote-username**

# copy tftp bootflash

On the Cisco 4500, to copy a boot image from a TFTP server to Flash memory on the Cisco 4500, use the **copy tftp bootflash** EXEC command.

**copy tftp bootflash**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

The router prompts for the address of the TFTP server and the name of the file. It provides an option to erase the existing boot image in Flash before writing the new image into Flash. The copying process takes several minutes; the actual time differs from network to network.

Before booting from Flash memory, verify that the checksum of the image in Flash memory matches the checksum listed in the README file that was distributed with the boot software image. The checksum of the boot image in Flash memory is displayed when the **copy tftp bootflash** command completes. The README file was copied to the TFTP server automatically when you installed the boot software image.

**Caution** If the checksum values do not match, do not reboot the router. Instead, reissue the **copy tftp bootflash** command and compare the checksums again. If the checksum is repeatedly wrong, copy the original boot software image back into Flash memory *before* you reboot the router from Flash memory.

## Example

The following example shows how to use this command:

```
Router# copy tftp bootflash

Boot flash directory:
File  Length   Name/status
  1   2622607  c4500-xboot
[2622672 bytes used, 1571632 available, 4194304 total]

Address or name of remote host [255.255.255.255]? 223.255.254.254
Source file name? c4500-xboot.101
Destination file name [c4500-xboot.101]?
Accessing file 'c4500-xboot.101' on 223.255.254.254...
Loading c4500-xboot.101 from 223.255.254.254 (via Ethernet0): ! [OK]

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

Copy 'c4500-xboot.101' from TFTP server into
     bootflash as 'c4500-xboot.101' WITH erase? [yes/no] yes
Erasing device... eeeeeeeeeeeeeeee ...erased
Loading c4500-xboot.101 from 223.255.254.254 (via Ethernet0): !!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 2622607/4194304 bytes]

Verifying checksum...  OK (0xE408)
Flash copy took 0:00:10 [hh:mm:ss]
```

## Related Commands

**copy bootflash tftp**
**copy mop bootflash**
**copy verify bootflash**
**erase bootflash**
**show bootflash**

# copy tftp flash

To copy a system image using TFTP into Flash memory, use the **copy tftp flash** EXEC command.

**copy tftp flash**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

The router prompts for the address of the TFTP server and TFTP filename. It provides an option to erase existing Flash memory before writing onto it. The entire copying process takes several minutes and will differ from network to network.

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

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

## Examples

The following example shows sample output of copying a system image named *IJ09140Z* into Flash memory:

```
Router# copy tftp flash

System flash directory, partition 2:
File  Length   Name/status
  1   984      ij09140z [deleted]
  2   984      ij09140z
[2096 bytes used, 8386512 available, 8388608 total]
Address or name of remote host [255.255.255.255]? 223.255.254.254
Source file name? ij09140z
Destination file name [ij09140z]?
Accessing file 'ij09140z' on 223.255.254.254...
Loading dirt/ssangiah/ij09140z .from 223.255.254.254 (via Ethernet0): ! [OK]

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

Copy 'ij09140z' from server
  as 'ij09140z' into Flash WITH erase? [yes/no] yes
```

```
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
Loading ij09140z from 223.255.254.254 (via Ethernet0): !
[OK - 984/8388608 bytes]

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

The exclamation points (!) indicate the copy process. The series of Vs in the sample output indicates that a checksum verification of the image is occurring after the image is written to Flash memory.

The following example shows sample output when copying a system image into a partition of Flash memory. The system will prompt only if there are two or more read/write partitions or one read-only and one read/write partition and dual Flash bank support in boot ROMs. If the partition entered is not valid, the process terminates. You can to enter a partition number, **?** for directory display of all partitions, or **?***number* for directory display of a particular partition. The default is the first read/write partition.

```
Router# copy tftp flash

System flash partition information:
Partition   Size     Used    Free    Bank-Size   State       Copy-Mode
    1       4096K    2048K   2048K   2048K       Read Only   RXBOOT-FLH
    2       4096K    2048K   2048K   2048K       Read/Write  Direct

[ Type ?<no> for partition directory; ? for full directory; q to abort]
Which partition? [default = 2]
```

If the partition is read-only and has dual Flash bank support in boot ROM, the session continues as follows:

```
                               ****  NOTICE  ****
Flash load helper v1.0
This process will accept the copy options and then terminate
the current system image to use the ROM based image for the copy.
Routing 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.
                               ---- ******** ----
Proceed? [confirm]
System flash directory, partition 1:
File  Length   Name/status
  1   3459720  master/igs-bfpx.100-4.3
[3459784 bytes used, 734520 available, 4194304 total]
Address or name of remote host [255.255.255.255]? 131.108.1.1
Source file name? master/igs-bfpx-100.4.3
Destination file name [default = source name]?
```

The file will be copied into the partition given by the user earlier:

```
Loading master/igs-bfpx.100-4.3 from 131.108.1.111: !
Erase flash device before writing? [confirm]
Flash contains files. Are you sure? [confirm]
Copy 'master/igs-bfpx.100-4.3' from TFTP server
as 'master/igs-bfpx.100-4.3' into Flash WITH erase? [yes/no] yes
```

If the partition is read-write, the session continues as follows:

```
System flash directory, partition 2:
File  Length   Name/status
  1   3459720  master/igs-bfpx.100-4.3
[3459784 bytes used, 734520 available, 4194304 total]
Address or name of remote host [255.255.255.255]? 131.108.1.1
Source file name? master/igs-bfpx.100-4.3
Destination file name [default = source name]?
```

The file will be copied into the partition given by the user earlier:

```
Accessing file 'master/igs-bfpx.100-4.3' on ABC.CISCO.COM...
Loading master/igs-bfpx.100-4.3 from 131.108.1.111: !
Erase flash device before writing? [confirm]
Flash contains files. Are you sure? [confirm]
Copy 'master/igs-bfpx.100-4.3' from TFTP server
as 'master/igs-bfpx.100-4.3' into Flash WITH erase? [yes/no] yes
```

## Related Commands

**boot system flash**
**copy flash tftp**
**copy verify**

# copy verify

To verify the checksum of a system image in Flash memory, use the **copy verify** EXEC command.

**copy verify**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

The **copy verify** command works on Cisco 3000, Cisco 4000 series, and Cisco 7000 series routers only.

Each system software or microcode image that is distributed on disk uses a single checksum for the entire image. This checksum is displayed only when the image is copied into Flash memory; it is *not* displayed when the image file is copied from one disk to another.

The README file (which is included with the image on the disk) lists the name, file size, and checksum of the image. Review the contents of the README file before loading or duplicating the new image so that you can verify the checksum when you copy it into the Flash memory or onto a TFTP server.

To display the contents of Flash memory, use the **show flash** or **show flash all** command. The Flash content listing does not include the checksum of individual files. To recompute and verify the image checksum after the image is copied into Flash memory, use the **copy verify** command. When you enter the command, the screen prompts you for the filename to verify. By default, it prompts for the last file in Flash (most recent). Press Return to recompute the default file checksum or enter the filename of a different file at the prompt.

## Example

The following example illustrates how to use this command:

```
Router# copy verify

Name of file to verify [gsxx]?
Verifying via checksum...
vvvvvvvvvvvvvvvvvvvvvvvvvvvvv

Flash verification successful. Length = 1923712, checksum = 0xA0C1
Router#
```

## Related Command

**copy tftp flash**

# copy verify bootflash

To verify the checksum of a boot image in Flash memory on the Cisco 4500, use the **copy verify bootflash** EXEC command.

**copy verify bootflash**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

You can use this command only on routers that have two banks of Flash: one bank for the boot image and the second bank for the system image.

Each boot software image that is distributed on disk uses a single checksum for the entire image. This checksum is displayed only when the image is copied into Flash memory; it is not displayed when the image file is copied from one disk to another.

The README file, which is included with the image on the disk, lists the name, file size, and checksum of the image. Review the contents of the README file before loading or duplicating the new image so that you can verify the checksum when you copy it into Flash memory or onto a server.

To display the contents of Flash memory, use the **show flash** command. The Flash contents listing does not include the checksum of individual files. To recompute and verify the image checksum after the image has been copied into Flash memory, use the **copy verify bootflash** command. When you enter the command, the system prompts you for the filename to verify. By default, it prompts for the last file (most recent) in Flash. Press Return to recompute the default file checksum, or enter the name of a different file at the prompt.

## Example

The following example illustrates how to use this command:

```
Router# copy verify bootflash

Boot flash directory:
File  name/status
  1   c4500-xboot
[1387336 bytes used, 2806968 bytes available]

Name of file to verify? c4500-xboot
Verifying checksum for 'c4500-xboot' (file # 1)...  [OK]
```

## Related Commands

**copy bootflash tftp**
**copy mop bootflash**
**copy tftp bootflash**
**erase bootflash**
**show bootflash**

# erase bootflash

To erase the boot image in Flash memory on the Cisco 4500, use the **erase bootflash** EXEC command.

**erase bootflash**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

You can use this command only on routers that have two banks of Flash memory: one bank for the boot image and the second bank for the system image.

## Example

The following example erases the boot image in Flash memory:

```
erase bootflash
```

## Related Commands

**copy bootflash tftp**
**copy mop bootflash**
**copy tftp bootflash**
**copy verify bootflash**
**show bootflash**

# erase flash

To erase Flash memory, use the **erase flash** EXEC command.

**erase flash**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

This command performs the same action as the **copy erase flash** command.

## Example

The following example illustrates how to use this command. Note that this example reflects the dual Flash bank feature available only on low-end systems (the AccessPro PC card, Cisco 2500 series, Cisco 3000 series, and Cisco 4000 series).

```
Router# erase flash

System flash partition information:
Partition   Size     Used    Free    Bank-Size   State       Copy-Mode
    1       4096K    2048K   2048K   2048K       Read Only   RXBOOT-FLH
    2       4096K    2048K   2048K   2048K       Read/Write  Direct

[ Type ?<no> for partition directory; ? for full directory; q to abort]
Which partition? [default = 2]
```

The system will prompt only if there are two or more read/write partitions. If the partition entered is not valid or is the read-only partition, the process terminates. You can enter a partition number, **?** for directory display of all partitions, or **?***number* for directory display of a particular partition. The default is the first read/write partition.

```
System flash directory, partition 2:
File  Length   Name/status
  1   3459720  master/igs-bfpx.100-4.3
[3459784 bytes used, 734520 available, 4194304 total]

Erase flash device, partition 2? [confirm] <Return>
```

# ip rarp-server

Use the **ip rarp-server** interface configuration command to allow the router to act as a Reverse Address Resolution Protocol (RARP) server. Use the **no ip rarp-server** command to restore the interface to the default of no RARP server support.

**ip rarp-server** *ip-address*
**no ip rarp-server** *ip-address*

## Syntax Description

| | |
|---|---|
| *ip-address* | IP address that is to be provided in the source protocol address field of the RARP response packet. Normally, this is set to whatever address you configure as the primary address for the interface. |

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

This feature makes diskless booting of clients possible between network subnets where the client and server are on separate subnets.

RARP server support is configurable on a per interface basis, so that the router does not interfere with RARP traffic on subnets that do not need RARP assistance from the router.

The router answers incoming RARP requests only if both of the following two conditions are met:

- The **ip rarp-server** command has been configured for the interface on which the request was received.
- There is a static entry found in the IP ARP table that maps the MAC address contained in the RARP request to an IP address.

Use the **show ip arp** EXEC command to display the contents of the IP ARP cache.

Sun Microsystems, Inc. makes use of RARP and UDP-based network services to facilitate network-based booting of SunOS on their workstations. By bridging RARP packets and using both the **ip helper-address** interface configuration command and the **ip forward-protocol** global configuration command, the router should be able to perform the necessary packet switching to enable booting of Sun workstations across subnets. Unfortunately, some Sun workstations assume that the sender of the RARP response, in this case the router, is the host the client can contact to TFTP load the bootstrap image. This causes the workstations to fail to boot.

By using the **ip rarp-server** feature, the router can be configured to answer these RARP requests, and the client machine should be able to reach its server by having its TFTP requests forwarded through the router that acts as the RARP server.

In the case of RARP responses to Sun workstations attempting to diskless boot, the IP address specified in the **ip rarp-server** interface configuration command should be the IP address of the TFTP server. In addition to configuring RARP service, the router must also be configured to forward UDP-based Sun portmapper requests to completely support diskless booting of Sun workstations. This can be accomplished using configuration commands of the form:

```
ip forward-protocol udp 111
interface interface name
ip helper-address target-address
```

RFC 903 documents the Reverse Address Resolution Protocol.

## Examples

The following partial example configures the router to act as a RARP server. The router is configured to use the primary address of the specified interface in its RARP responses.

```
arp 128.105.2.5 0800.2002.ff5b arpa
interface ethernet 0
ip address 128.105.3.100 255.255.255.0
ip rarp-server 128.105.3.100
```

In the following example, the router is configured to act as a RARP server, with TFTP and portmapper requests forwarded to the Sun server:

```
! Allow the router to forward broadcast portmapper requests
ip forward-protocol udp 111
! Provide the router with the IP address of the diskless sun
arp 128.105.2.5 0800.2002.ff5b arpa
interface ethernet 0
! Configure the router to act as a RARP server, using the Sun Server's IP
! address in the RARP response packet.
ip rarp-server 128.105.3.100
! Portmapper broadcasts from this interface are sent to the Sun Server.
ip helper-address 128.105.3.100
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**ip forward-protocol** †
**ip helper-address** †

# ip rcmd domain-lookup

Use the **ip rcmd domain-lookup** global configuration command to enable Domain Name System (DNS) security for rcp and rsh. To bypass DNS security for rcp and rsh, use the **no** form of this command.

**ip rcmd domain-lookup**
**no ip rcmd domain-lookup**

## Syntax Description

This command has no arguments or keywords.

## Default

Enabled

## Command Mode

Global configuration

## Usage Guidelines

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

This command will turn off DNS lookups for rsh and rcp only. The **no ip domain-lookup** command takes precedence over the **ip rcmd domain-lookup** command. If **ip domain-lookup** is disabled using the **no ip domain-lookup** command, DNS will be bypassed for rcp and rsh, even if **ip rcmd domain-lookup** is enabled.

---

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

---

## Example

In the following example, DNS security is enabled for rcp and rsh:

```
ip rcmd domain-lookup
```

## Related Command

A dagger (†) indicates that the command is documented in another chapter.

**ip domain-lookup** †

# ip rcmd rcp-enable

To configure the router to allow remote users to copy files to and from the router, use the **ip rcmd rcp-enable** global configuration command. Use the **no rcp-enable** command to disable a router that is enabled for rcp.

**ip rcmd rcp-enable**
**no ip rcmd rcp-enable**

## Syntax Description

This command has no arguments or keywords.

## Default

To ensure security, the router is not enabled for rcp by default.

## Command Mode

Global configuration

## Usage Guidelines

To allow a remote user to execute rcp commands on the router, you must also create an entry for the remote user in the local router's authentication database.

The **no ip rcmd rcp-enable command** does not prohibit a local user from using rcp to copy system images and configuration files to and from the router.

To protect against undesirable users copying the system image or configuration files without consent, the router is not enabled for rcp by default.

---

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

---

## Example

The following example shows how to enable the router for rcp:

```
rcp-enable
```

## Related Command

**ip rcmd remote-host**

# ip rcmd remote-host

To allow remote users to execute commands on the router using rsh or rcp, use the **ip rcmd remote-host** global configuration command to create an entry for the remote user in a local authentication database. Use the **no ip rcmd remote-host** command to remove an entry for a remote user from the local authentication database.

**ip rcmd remote-host** *local-username* {*ip-address* | *host*} *remote-username* [**enable**]
**no ip rcmd remote-host** *local-username* {*ip-address* | *host*} *remote-username* [**enable**]

## Syntax Description

| | |
|---|---|
| *local-username* | Name of the user on the local router. You can specify the router host name as the username. This name needs to be communicated to the network administrator or the user on the remote system. To be allowed to remotely execute commands on the router, the remote user must specify this value correctly. |
| *ip-address* | IP address of the remote host from which the local router will accept remotely executed commands. Either the IP address or the host name is required. |
| *host* | Name of the remote host from which the local router will accept remotely executed commands. Either the host name or the IP address is required. |
| *remote-username* | Name of the user on the remote host from which the router will accept remotely executed commands. |
| **enable** | (Optional) Enables the remote user to execute privileged EXEC commands using rsh. This keyword does not apply to rcp. |

## Command Mode

Global configuration

## Usage Guidelines

A TCP connection to a router is established using an IP address. Using the host name is valid only when you are initiating an rcp or rsh command from a local router. The host name is converted to an IP address using DNS or host-name aliasing.

To allow a remote user to execute rcp or rsh commands on a local router, you must create an entry for the remote user in the local router's authentication database. You must also enable the router to act as an rsh or rcp server.

To enable the router to act as an rsh server, issue the **ip rcmd rsh-enable** command. To enable the router to act as an rcp server, issue the **ip rcmd rcp-enable** command.The router cannot act as a server for either of these protocols unless you explicitly enable the capacity.

A local authentication database, which is similar in concept and use to a UNIX *.rhosts* file, is used to enforce security on the router through access control. Each entry that you configure in the authentication database identifies the local user, the remote host, and the remote user. To permit a remote user of rsh to execute commands in privileged EXEC mode, specify the **enable** keyword.

An entry that you configure in the router authentication database differs from an entry in a UNIX *.rhost* file in the following aspect. Because the *.rhosts* file on a UNIX system resides in the home directory of a local user account, an entry in a UNIX *.rhosts* file does not need to include the local username; the local username is determined from the user account. To provide equivalent support on a router configured, specify the local username along with the remote host and remote username in each authentication database entry that you configure.

For a remote user to be able to execute commands on the router in its capacity as a server, the local username, host address or name, and remote username sent with the remote client request must match values configured in an entry in the local authentication file.

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

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

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

---

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

---

## Example

The following example allows the remote user *netadmin3* on a remote host with the IP address 131.108.101.101 to execute commands on *router1* using the rsh protocol. For rsh, user *netadmin3* is allowed to execute commands in privileged EXEC mode.

```
ip rcmd remote-host router1 131.108.101.101 netadmin3 enable
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**ip rcmd rcp-enable**
**ip rcmd rsh-enable**
**no ip domain-lookup** †

# ip rcmd remote-username

To configure the remote username to be used when requesting a remote copy using rcp, use the **ip rcmd remote-username** global configuration command. To remove from the configuration the remote username, use the **no ip rcmd remote-username** command.

**ip rcmd remote-username** *username*
**no ip rcmd remote-username** *username*

**Caution** The remote username must be associated with an account on the destination server.

## Syntax Description

| | |
|---|---|
| *username* | Name of the remote user on the server. This name is used for rcp copy requests. All files and images to be copied are searched for or written relative to the directory of the remote user's account, if the server has a directory structure, for example, as do UNIX systems. |

## Default

If you do not issue this command, the router software sends the remote username associated with the current TTY process, if that name is valid, for rcp copy commands. For example, if the user is connected to the router through Telnet and the user was authenticated through the **username** command, then the router software sends that username as the remote username.

## Command Mode

Global configuration

## Usage Guidelines

The rcp protocol requires that a client send the remote username on an rcp request to the server. Use this command to specify the remote username to be sent to the server for an rcp copy request. All files and images to be copied are searched for or written relative to the directory of the remote user's account, if the server has a directory structure, for example, as do UNIX systems.

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

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

**Note** For Cisco, TTYs are commonly used in communication servers. The concept of TTYs originated with UNIX. For UNIX systems, each physical device is represented in the file system. Terminals are called TTY devices (which stands for teletype, the original UNIX terminal).

## Example

The following example shows how to use this command:

```
configure terminal
ip rcmd remote-username netadmin1
Ctrl-Z
```

## Related Commands

**boot network rcp**
**boot system rcp**
**copy bootflash rcp**
**copy flash rcp**
**copy rcp bootflash**
**copy rcp flash**
**copy rcp running-config**
**copy rcp startup-config**
**copy running-config rcp**
**copy startup-config rcp**

# ip rcmd rsh-enable

To configure the router to allow remote users to execute commands on the router using rsh, use the **ip rcmd rsh-enable** global configuration command. Use the **no ip rcmd rsh-enable** command to disable a router that is enabled for rsh.

**ip rcmd rsh-enable**
**no ip rcmd rsh-enable**

## Syntax Description

This command has no arguments or keywords.

## Default

To ensure security, the router is not enabled for rsh by default.

## Command Mode

Global configuration

## Usage Guidelines

Use this command to enable the router to receive rsh requests from remote users. In addition to issuing this command, to allow a remote user to execute rsh commands on the router, you must create an entry for the remote user in the local router's authentication database.

The **no ip rcmd rsh-enable** command does not prohibit a local user of the router from executing a command on other routers and UNIX hosts on the network using rsh.

---

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

---

## Example

The following example shows how to enable the router as an rsh server:

```
ip rcmd rsh-enable
```

## Related Command

**ip rcmd remote-host**

# microcode

To specify the location of the microcode you want to download from Flash memory into the writable control store (WCS) on a Cisco 7000 series, use the **microcode** interface configuration command.

**microcode** *interface* [**flash** | **rom** | **system**] [**flash** *filename*]
**no microcode** *interface* [**flash** | **rom**] [**flash** *filename*]

## Syntax Description

| | |
|---|---|
| *interface* | One of the following interface processor names: **aip, fip**, **fsip, hip, mip, trip**, **eip**, or **sp**. |
| **flash** | (Optional) If the **flash** keyword is specified, a *filename* argument is required, unless you are using the **no microcode** *interface* **flash** command. |
| **rom** | (Optional) If the **rom** keyword is specified, no further arguments are necessary. For example, the command **microcode fip rom** specifies that all FDDI Interface Processors (FIPs) should be loaded from their onboard ROM microcode. This onboard ROM microcode is not the same as the eight ROMs on the RP that contain the system image. |
| **system** | (Optional) If **system** is specified, the router loads the microcode from the microcode bundled into the system image you are running for that interface type. |
| *filename* | (Optional) Filename of the microcode in Flash memory that you want to download. This argument is only used with the **flash** keyword. If you use the **flash** keyword, the name of the microcode file in Flash is required unless the command is **no microcode** *interface* **flash**. (This command results in the same default condition as the command **microcode** *interface* **rom**, which indicates that the card should be loaded from its onboard ROM microcode.) |

## Default

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

## Command Mode

Interface configuration

## Examples

In the following example, all FIP cards will use their onboard ROM microcode:

```
microcode fip rom
```

In the following example, all FIP cards will be loaded with the microcode found in Flash memory file *fip.v141-7* when the system is booted, when a card is inserted or removed, or when the **microcode reload** interface configuration command is issued. The configuration is then written to NVRAM.

```
microcode fip flash fip.v141-7
^Z
> write memory
```

## Related Command

**microcode reload**

# microcode reload

To signal to the Cisco 7000 series that all microcode configuration commands have been entered and the processor cards should be reloaded, use the **microcode reload** interface configuration command.

**microcode reload**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Interface configuration

## Example

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:

```
microcode reload
```

## Related Command

**microcode**

# mop device-code

To identify the type of device sending MOP sysid messages and request program messages, use the **mop device-code** global configuration command. Use the **no mop device-code** command to set the identity to the default value.

**mop device-code** {**cisco | ds200**}
**no mop device-code** {**cisco | ds200**}

## Syntax Description

| | |
|---|---|
| **cisco** | Denotes a Cisco device code. |
| **ds200** | Denotes a DECserver 200 device code. |

## Default

Cisco device code

## Command Mode

Global configuration

## Usage Guidelines

The sysid messages and request program messages use the identity information indicated by this command.

## Example

The following example identifies a DECserver 200 device as sending MOP sysid and request program messages:

```
mop device-code ds200
```

## Related Command

A dagger (†) indicates that the command is documented in another chapter.

**mop sysid** †

# mop retransmit-timer

To configure the length of time the router waits before retransmitting boot requests to a MOP server, use the **mop retransmit-timer** global configuration command. Use the **no mop retransmit-timer** command to reinstate the default value.

**mop retransmit-timer** *seconds*
**no mop retransmit-timer**

## Syntax Description

| | |
|---|---|
| *seconds* | Sets the length of time, in seconds, that the router waits before retransmitting a message. The value is a number from 1 to 20. |

## Default

4 seconds

## Command Mode

Global configuration

## Usage Guidelines

By default, when the router transmits a request that requires a response from a MOP boot server and the server does not respond, the message will be retransmitted after 4 seconds. 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 might want to configure the router to wait longer than 4 seconds before retransmitting the message if you are using such a link.

## Example

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

```
mop retransmit-timer 10
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**mop device-code**
**mop retries**
**mop enabled**†

# mop retries

To configure the number of times a router will retransmit boot requests to a MOP server, use the **mop retries** global configuration command. Use the **no mop retries** command to reinstate the default value.

**mop retries** *count*
**no mop retries**

## Syntax Description

| | |
|---|---|
| *count* | Indicates the number of times a router will retransmit a MOP boot request. The value is a number from 3 to 24. |

## Default

8 times

## Command Mode

Global configuration

## Example

In the following example, the router will attempt to retransmit a message to an unresponsive host 11 times before declaring a failure:

```
mop retries 11
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**mop device-code**
**mop retransmit-timer**
**mop enabled** †

# o

To list the value of the boot field (bits 0-3) in the configuration register, use the ROM monitor **o** command. To reset the value of the boot field so that the router boots from ROM, use the ROM monitor **o/r** command.

**o**
**o/r**

## Syntax Description

This command has no arguments or keywords.

## Default

Refer to the appropriate hardware installation guide for default values.

## Command Mode

ROM monitor

## Usage Guidelines

To get to the ROM monitor prompt at a Cisco 2000, Cisco 3000, Cisco 4000, or Cisco 7000 series, use the **reload** EXEC command if the configuration register has a boot value of 0. (For systems with a software configuration register, a value can be included on the **o/r** command line.) Use the **i** command in conjunction with the **o/r** command to initialize the router. (The **i** command is documented in the *Hardware Installation and Maintenance* publication for your product.) The **o/r** command resets the configuration register to 0x141, which disables the Break key, ignores the NVRAM configuration, and boots the default system image from ROM.

## Examples

The following is an example of the **o** command:

```
> o

Bit#Configuration register option settings:
15Diagnostic mode disabled
14IP broadcasts do not have network numbers
13Do not boot default ROM software if network boot fails
12-11Console speed is 9600 baud
10IP broadcasts with ones
09Do not use secondary bootstrap
08Break enabled
07OEM disabled
06Ignore configuration disabled
03-00Boot to ROM monitor

>
```

The following is an example of the **o/r** and **i** commands used to reset and boot the default system image from ROM:

```
> o/r
> i
```

## Related Command

**config-register**

# partition flash

To separate Flash memory into two partitions, use the **partition flash** global configuration command. Use the **no** form of this command to undo partitioning, and restore Flash memory to one partition.

**partition flash** *partitions* [*size1 size2*]
**no partition flash**

## Syntax Description

| | |
|---|---|
| *partitions* | Number of partitions in Flash memory. Can be 1 or 2. |
| *size1* | (Optional) Size of the first partition in megabytes. |
| *size2* | (Optional) Size of the second partition in megabytes. |

## Default

Flash memory consists of one partition.

If this command is entered but partition size is not specified, two partitions of equal size will be created.

## Command Mode

Global configuration

## Usage Guidelines

To undo partitioning, use either the **partition flash 1** or **no partition flash** command. If one or more files exist in the second partition, you must manually erase the second partition with the **erase flash** command before reverting to a single partition.

When creating two partitions, you must not truncate a file or cause the spillover of a file into the second partition.

## Example

The following example creates two partitions of 4 MB each in Flash memory:

```
partition flash 2 4 4
```

# reload

To reload the operating system, use the **reload** EXEC command.

**reload**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

The **reload** command halts the system. If the system is set to restart on error, it reboots itself. Use the **reload** command after configuration information is entered into a file and saved into NVRAM.

You cannot reload from a virtual terminal if the system is not set up for automatic booting. This prevents the system from dropping to the ROM monitor and thereby taking the system out of the remote user's control.

If you modify your configuration file, you are prompted to save the configuration.

## Example

The following example illustrates how to enter the **reload** command at the privileged EXEC prompt:

```
Router# reload
```

## Related Command

**write memory**

# rsh

To execute a command remotely on a remote rsh host, use the **rsh** privileged EXEC command.

**rsh** {*ip-address* **|** *host*} [**/user** *username*] *remote-command*

## Syntax Description

| | |
|---|---|
| *ip-address* | IP address of the remote host on which to execute the rsh command. Either the IP address or the host name is required. |
| *host* | Name of the remote host on which to execute the command. Either the host name or the IP address is required. |
| **/user** *username* | (Optional) Remote username. |
| *remote-command* | Command to be executed remotely. This is a required parameter. |

## Default

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

**Note** For Cisco, TTYs are commonly used in communications servers. The concept of TTYs originated with UNIX. For UNIX systems, each physical device is represented in the file system. Terminals are called TTY devices (which stands for teletype, the original UNIX terminal).

## Command Mode

Privileged EXEC

## Usage Guidelines

Use the **rsh** command to execute commands remotely. The host on which you remotely execute the command must support the rsh protocol, and the *.rhosts* files on the rsh host must include an entry that permits you to remotely execute commands on that host.

For security reasons, the router software does not default to a remote login if no command is specified, as does UNIX. Instead, the router provides Telnet and connect services that you can use rather than rsh.

## Example

The following command specifies that user *sharon* attempts to remotely execute the UNIX ls command with the -a argument on the remote host *mysys.cisco.com* . The command output resulting from the remote execution follows the command example:

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

# service compress-config

To compress configuration files on the Cisco 7000 series, Cisco 4000 series, Cisco 3000, and AGS+ routers, which have NVRAM, use the **service compress-config** global configuration command. To disable compression, use the **no** form of this command.

**service compress-config**
**no service compress-config**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Global configuration

## Usage Guidelines

If the file compression completes successfully, the following message is displayed:

```
Compressing configuration from configuration-size to compressed-size
[OK]
```

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
```

If the file compression fails, the following message is displayed:

```
Error trying to compress nvram
```

One way to determine whether a configuration file will compress enough to fit into NVRAM is to use a text editor to enter the configuration, then use the UNIX **compress** command to check the compressed size. To get a closer approximation of the compression ratio, use the UNIX command **compress -b12**.

Once the configuration file has been compressed, the router functions normally. A **show configuration** command would uncompress the configuration before displaying it. At boot time, the system would recognize that the configuration file was compressed, uncompress it, and proceed normally.

To disable compression of the configuration file, enter configuration mode and specify the **no service compress-config** command. Then enter the **write memory** command. The router displays an OK message if it is able to successfully write the uncompressed configuration to NVRAM. Otherwise, the router displays an error message indicating that the configuration is too large to store. If the configuration file is larger than the physical NVRAM, the following message is displayed:

```
###Configuration too large to fit uncompressed in NVRAM Truncate configuration? [confirm]
```

To truncate and save the configuration, type **Y.** To not truncate and not save the configuration, type **N.**

## Example

In the following example, the configuration file is compressed:

```
service compress-config
```

## Related Command

**show configuration**

# service config

To enable autoloading of configuration files from a network server, use the **service config** global configuration command. Use the **no service config** command to restore the default.

**service config**
**no service config**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled, except on systems without NVRAM or with invalid or incomplete information in NVRAM. In these cases, autoloading of configuration files from a network server is enabled automatically.

## Command Mode

Global configuration

## Usage Guidelines

Usually, the **service config** command is used in conjunction with the **boot host** or **boot network** command. You must enter the **service config** command to enable the router to automatically configure the system from the file specified by the **boot host** or **boot network** command.

The **service config** command can also be used without the **boot host** or **boot network** command. If you do not specify host or network configuration filenames, the router uses the default configuration files. The default network configuration file is *network-confg*. The default host configuration file is host-confg, where host is the host name of the router. If the router cannot resolve its host name, the default host configuration file is *router-confg*.

## Example

In the following example, the router is configured to autoload the default host configuration file:

```
boot host
service config
```

## Related Commands

**boot host**
**boot network**

# show async-bootp

Use the **show async-bootp** privileged EXEC command to display the parameters that have been configured for SLIP extended BOOTP requests.

**show async-bootp**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

**Privileged EXEC**

## Sample Display

The following is a sample output of the **show async-bootp** command:

```
Router# show async-bootp

The following extended data will be sent in BOOTP responses:

bootfile (for address 128.128.1.1) "pcboot"
bootfile (for address 131.108.1.111) "dirtboot"
subnet-mask 255.255.0.0
time-offset -3600
time-server 128.128.1.1
```

Table 3-2 describes significant fields shown in the display.

**Table 3-2 Show Async-BOOTP Field Descriptions**

| Field | Description |
|---|---|
| bootfile... “pcboot” | Boot file for address 128.128.1.1 is named pcboot. |
| subnet-mask 255.255.0.0 | Subnet mask. |
| time-offset -3600 | Local time is one hour (3600 seconds) earlier than UTC time. |
| time-server 128.128.1.1 | Address of the time server for the network. |

## Related Command

**async-bootp**

# show bootflash

To verify boot Flash memory on the Cisco 4500, use the **show bootflash** EXEC command.

**show bootflash**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

You can use this command only on routers that have two banks of Flash: one bank for the boot image and the second bank for the system image.

The **show bootflash** command displays the type of boot Flash memory present, any files that may currently exist in boot Flash memory, and the amount of boot Flash memory used and remaining.

## Sample Display

The following is sample output from the **show bootflash** command:

```
Router# show bootflash
Boot flash directory:
File  name/status
  1   c4500-xboot
[1387336 bytes used, 2806968 bytes available]
```

Table 3-3 describes the fields shown in the output.

**Table 3-3 Show Bootflash Field Descriptions**

| Field | Description |
|---|---|
| Boot File | Number of the boot file. |
| flash directory: name/status | Name and status of the boot file. The status is displayed if appropriate and can be one of the following:<br>• [deleted]—File has been deleted.<br>• [invalid checksum]—File has an incorrect checksum. |

# show configuration

Use the **show configuration** EXEC command to display the contents of the NVRAM, if present and valid.

**show configuration**

NVRAM stores the configuration information in the network server in text form as configuration commands. The **show configuration** command shows the version number of the software used when you last executed the **write memory** command.

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show configuration** command:

```
Router# show configuration

Using 5057 out of 32768 bytes
!
enable-password xxxx
service pad
!
boot system dross-system 131.108.13.111
boot system dross-system 131.108.1.111
!
exception dump 131.108.13.111
!
no ip ipname-lookup
!
decnet routing 13.1
decnet node-type area
decnet max-address 1023
!
interface Ethernet 0
ip address 131.108.1.1 255.255.255.0
ip helper-address 131.120.1.0
ip accounting
ip gdp
decnet cost 3
!
ip domain-name CISCO.COM
ip name-server 255.255.255.255
!
end
```

The following is partial sample output from the **show configuration** command when the configuration file has been compressed:

```
Router# show configuration
Using 21542 out of 65536 bytes, uncompressed size = 142085 bytes
!
version 9.22
service compress-config
!
hostname kyoto
!
boot system flash gs7-k.sthormod_clean
boot system rom
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**configure**
**description** †
**service compress-config**
**write memory**
**write terminal**

# show flash

Use the **show flash** EXEC command to verify Flash memory. The **show flash** command displays the type of Flash memory present, any files that might currently exist in Flash memory, and the amounts of Flash memory used and remaining.

**show flash** [**all** | **chips** | **detailed** | **err** | **partition** *number* [**all** | **chips** | **detailed** | **err** ] | **summary**]

## Syntax Description

| | |
|---|---|
| **all** | (Optional) Shows complete information about Flash memory, including information about the individual ROM devices in Flash memory and the names and sizes of all system image files stored in Flash memory, including those that are invalidated. |
| **chips** | (Optional) Shows information per partition and per chip, including which bank the chip is in, its code, size, and name. |
| **detailed** | (Optional) Shows detailed file directory information per partition, including file length, address, name, Flash checksum, computer checksum, bytes used, bytes available, total bytes, and bytes of system Flash memory. |
| **err** | (Optional) Shows write or erase failures in the form of number of retries. |
| **partition** *number* | (Optional) Shows output for the specified partition number. If you specify the **partition** keyword, you must specify a partition number. You can use this keyword only when Flash memory has multiple partitions. |
| **summary** | (Optional) Shows summary information per partition, including the partition size, bank size, state, and method by which files can be copied into a particular partition. You can use this keyword only when Flash memory has multiple partitions. |

## Command Mode

EXEC

## Sample Displays

The following is sample output from the **show flash** command on the Cisco 3000 and Cisco 7000 series:

```
Router# show flash

4096K bytes of flash memory sized on embedded flash.

File     name/status
  0     ahp4/gs7-k
  1     micro/eip1-0
  2     micro/sp1-3
  3     micro/trip1-1
  4     micro/hip1-0
```

```
  5     micro/fip1-1
  6     fsipucode
  7     spucode
  8     tripucode
  9     fipucode
  10    eipucode
  11    hipucode
  12    sipucode
  13    sp_q160-1
  14    ahp4/sp160-3 [deleted]
  15    ahp4/sp160-3
[682680/4194304 bytes free/total]
```

Table 3-4 describes the **show flash** display fields for the Cisco 3000 and Cisco 7000 series.

**Table 3-4 Show Flash Field Descriptions**

| Field | Description |
|---|---|
| File | Number of file in Flash memory |
| name/status | Files that currently exist in Flash memory |
| bytes free | Amount of Flash memory remaining |
| [deleted] | Flag indicating that another file exists with the same name or that process has been aborted |

As the display shows, the Flash memory can store and display multiple, independent software images for booting itself or for TFTP server software for other products. This feature is useful for storing default system software. These images can be stored in compressed format (but cannot be compressed by the router).

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

The following is a sample output from the **show flash** command on a router that has Flash memory partitioned:

```
Router# show flash

System flash directory, partition 1:
  File  Length   Name/status
    1   3459720  master/igs-bfpx.100-4.3
  [3459784 bytes used, 734520 available, 4194304 total]
4096K bytes of processor board System flash (Read Only)

System flash directory, partition 2:
  File  Length   Name/status
    1   3459720  igs-kf
  [3459784 bytes used, 734520 available, 4194304 total]
  4096K bytes of processor board System flash (Read/Write)
```

The following is a sample output from the **show flash all** command on the Cisco 3000 and Cisco 7000. The format of the display is different on different router models. The format of your display might differ.

```
Router# show flash all

4096K bytes of flash memory sized on embedded flash.
   Chip    socket   code      bytes     name
     0       U63     89BD   0x040000   INTEL 28F020
```

```
      1       U62     89BD   0x040000   INTEL 28F020
      2       U61     89BD   0x040000   INTEL 28F020
      3       U60     89BD   0x040000   INTEL 28F020
      4       U48     89BD   0x040000   INTEL 28F020
      5       U47     89BD   0x040000   INTEL 28F020
      6       U46     89BD   0x040000   INTEL 28F020
      7       U45     89BD   0x040000   INTEL 28F020
      8       U30     89BD   0x040000   INTEL 28F020
      9       U29     89BD   0x040000   INTEL 28F020
     10       U28     89BD   0x040000   INTEL 28F020
     11       U27     89BD   0x040000   INTEL 28F020
     12       U17     89BD   0x040000   INTEL 28F020
     13       U16     89BD   0x040000   INTEL 28F020
     14       U15     89BD   0x040000   INTEL 28F020
     15       U14     89BD   0x040000   INTEL 28F020

Flash file directory:
File  name/status
        addr     length    fckcsum  ccksum
 0  gs7-k
      0x12000080  2601100   0x4015 0x4015
 1  micro/eip1-0
      0x1227B14C  53364     0x0    0x0
 2  micro/sp1-3
      0x12288200  55418     0x0    0x0
 3  micro/trip1-1
      0x12295ABC  105806    0x0    0x0
 4  micro/hip1-0
      0x122AF84C  35528     0x0    0x0
 5  micro/fip1-1
      0x122B8354  97070     0x0    0x0
 6  fsipucode
      0x122CFEC4  6590      0x0    0x0
 7  spucode
      0x122D18C4  55418     0x0    0x0
 8  tripucode
      0x122DF180  105806    0x0    0x0
 9  fipucode
      0x122F8F10  97070     0x0    0x0
 10  eipucode
      0x12310A80  53330     0x60A1 0x60A1
 11  hipucode
      0x1231DB14  35528     0x0    0x0
 12  sipucode
      0x1232661C  54040     0x0    0x0
 13  sp_q160-1
      0x1233974  42912      0x0    0x0
 14  ahp4/sp160-3 [deleted]
      0x1233E154  55730   0x0 0x0
 15  ahp4/sp160-3
      0x1234BB48  55808   0x0 0x0
[682680/4194304 bytes free/total]
```

Table 3-5 describes the **show flash all** display fields for the Cisco 3000 and Cisco 7000 series.

**Table 3-5 Show Flash All Field Descriptions**

| Field | Description |
|---|---|
| bytes of flash memory sized on embedded flash | Total amount of Flash memory present. |
| Chip | Identifies the ROM unit. |

| Field | Description |
|---|---|
| socket | Location of the ROM unit. |
| code | Vendor code identifying the vendor of the ROM unit. |
| bytes | Size of the ROM unit (in hex bytes). |
| name (in row beginning with Chip) | Vendor name and chip part number of the ROM unit. |
| security jumper, flash memory | Security jumper is/is not installed. Flash memory is programmable or read-only. If the security jumper is not installed, you will see the **show flash** display with a message indicating that the jumper is not installed. |
| File | Number of the system image file. If no filename is specified in the **boot system flash** command, the router boots the system image file with the lowest file number. |
| name/status | Filename and status of a system image file. The status [invalidated] appears when a file has been rewritten (recopied) into Flash memory. The first (now invalidated) copy of the file is still present within Flash memory, but it is rendered unusable in favor of the newest version. The [invalidated] status can also indicate an incomplete file that results from the user aborting the copy process, a network timeout, or a Flash memory overflow. |
| addr | Address of the file in Flash memory. |
| length | Size of the system image file (in bytes). |
| fcksum | Checksum recorded in Flash memory. |
| ccksum | Computer checksum. |
| [deleted] | Flag indicating that another file exists with the same name or that process has been aborted. |
| bytes free/total | Amount of Flash memory used/total amount of Flash memory. |

In the following example, the security jumper is not installed and you cannot write to Flash memory until the security jumper is installed:

```
Router# show flash all

4096K bytes of flash memory on embedded flash (in RP1).
  security jumper(12V) is not installed,
flash memory is read-only.

file     offset length   name
00xDCD0     1903892    gs7-k [deleted]
10x1DEA24   1903912    gs7-k
 [329908/4194304 bytes free]
```

The following is sample output for the **show flash all** command on a Cisco 3000 that has Flash memory partitioned:

```
Router# show flash all

System flash partition information:
Partition   Size     Used      Free    Bank-Size   State       Copy-Mode
    1       4096K    3459K      637K   4096K       Read Only   RXBOOT-FLH
    2       4096K    3224K      872K   4096K       Read/Write  Direct

System flash directory, partition 1:
File  Length   Name/status
        addr      fcksum  ccksum
  1   3459720  master/igs-bfpx.100-4.3
```

```
        0x40      0x3DE1  0x3DE1
[3459784 bytes used, 734520 available, 4194304 total]
4096K bytes of processor board System flash (Read ONLY)

   Chip    Bank     Code      Size      Name
    1      1        89A2      1024KB    INTEL 28F008SA
    2      1        89A2      1024KB    INTEL 28F008SA
    3      1        89A2      1024KB    INTEL 28F008SA
    4      1        89A2      1024KB    INTEL 28F008SA
Executing current image from System flash [partition 1]

 System flash directory, partition2:
File  Length   Name/status
        addr      fcksum  ccksum
  1   3224008  igs-kf.100
        0x40      0xEE91  0xEE91
[3224072 bytes used, 970232 available, 4194304 total]
4096K bytes of processor board System flash (Read/Write)

   Chip    Bank     Code      Size      Name
    1      2        89A2      1024KB    INTEL 28F008SA
    2      2        89A2      1024KB    INTEL 28F008SA
    3      2        89A2      1024KB    INTEL 28F008SA
    4      2        89A2      1024KB    INTEL 28F008SA
```

Table 3-6 describes the additional fields in the display.

**Table 3-6 Show Flash All Fields for Partitioned Flash Memory**

| Field | Description |
|---|---|
| Partition | Partition number in Flash memory. |
| Size | Size of partition in bytes. |
| Used | Number of bytes used in partition. |
| Free | Number of bytes free in partition. |
| Bank-Size | Size of bank in bytes. |
| State | State of the partition. It can be one of the following values:<br>• Read-Only indicates the partition that is being executed from.<br>• Read/Write is a partition that can be copied to. |
| Copy-Mode | Method by which the partition can be copied to:<br>• RXBOOT-FLH indicates copy via Flash load helper.<br>• Direct indicates user can copy directly into Flash memory.<br>• None indicates that it is not possible to copy into that partition. |
| Chip | Chip number. |
| Bank | Bank number. |
| Code | Code number. |
| Size | Size of chip. |
| Name | Name of chip. |

The following is sample output for the **show flash chips** command on a router that has Flash memory partitioned:

```
Router# show flash chips

System flash partition 1:
4096K bytes of processor board System flash (Read ONLY)

  Chip    Bank    Code      Size      Name
   1      1       89A2      1024KB    INTEL 28F008SA
   2      1       89A2      1024KB    INTEL 28F008SA
   3      1       89A2      1024KB    INTEL 28F008SA
   4      1       89A2      1024KB    INTEL 28F008SA
Executing current image from System flash [partition 1]

System flash partition 2:
4096K bytes of processor board System flash (Read/Write)

  Chip    Bank    Code      Size      Name
   1      2       89A2      1024KB    INTEL 28F008SA
   2      2       89A2      1024KB    INTEL 28F008SA
   3      2       89A2      1024KB    INTEL 28F008SA
   4      2       89A2      1024KB    INTEL 28F008SA
```

The following is sample output for the **show flash detailed** command on a router that has Flash memory partitioned:

```
Router# show flash detailed

System flash directory, partition 1:
File  Length   Name/status
        addr      fcksum  ccksum
  1   3224008  igs-kf.100
        0x40      0xEE91  0xEE91
[3224072 bytes used, 970232 available, 4194304 total]
4096K bytes of processor board System flash (Read/Write)

System flash directory, partition 2:
File  Length   Name/status
        addr      fcksum  ccksum
  1   3224008  igs-kf.100
        0x40      0xEE91  0xEE91
[3224072 bytes used, 970232 available, 4194304 total]
4096K bytes of processor board System flash (Read/Write)
```

The following is sample output for the **show flash err** command on a Cisco 3000 that has Flash memory partitioned:

```
Router# show flash err

System flash directory, partition 1:
File  Length   Name/status
  1   37376    master/igs-bfpx.100-4.3  [invalid checksum]
[37440 bytes used, 4156864 available, 4194304 total]
4096K bytes of processor board System flash (Read/Write)

   Chip    Bank     Code      Size      Name                erase  write
    1      1        89A2      1024KB    INTEL 28F008SA      0      0
    2      1        89A2      1024KB    INTEL 28F008SA      0      0
    3      1        89A2      1024KB    INTEL 28F008SA      0      0
    4      1        89A2      1024KB    INTEL 28F008SA      0      0
Executing current image from System flash [partition 1]

System flash directory, partition 2:
```

```
File  Length   Name/status
  1   37376    master/igs-bfpx.100-4.3  [invalid checksum]
[37440 bytes used, 4156864 available, 4194304 total]
4096K bytes of processor board System flash (Read/Write)

   Chip    Bank     Code      Size      Name               erase  write
    1      2        89A2      1024KB    INTEL 28F008SA      0      0
    2      2        89A2      1024KB    INTEL 28F008SA      0      0
    3      2        89A2      1024KB    INTEL 28F008SA      0      0
    4      2        89A2      1024KB    INTEL 28F008SA      0      0
```

The following is sample output for the **show flash summary** command on a router that has Flash memory partitioned. The partition that indicates a state of "Read Only" is the partition that is being executed from.

```
Router# show flash summary

System flash partition information:
Partition   Size     Used      Free    Bank-Size   State       Copy-Mode
    1       4096K    2048K     2048K   2048K       Read Only   RXBOOT-FLH
    2       4096K    2048K     2048K   2048K       Read/Write  Direct
```

The following are possible values for Copy-Mode:

- RXBOOT-MANUAL—Copy manually by reloading to the boot ROM image.
- RXBOOT-FLH—Copy via Flash load helper.
- Direct—Copy directly into Flash memory.
- None—Copy not allowed.

# show flh-log

To view the system console output generated during the Flash load helper operation, use the **show flh-log** privileged EXEC command.

**show flh-log**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Privileged EXEC

## Usage Guidelines

If you are a remote Telnet user performing the Flash upgrade without a console connection, this command 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.

## Sample Display

The following is sample output from the **show flh-log** command:

```
Router# show flh-log

%FLH: abc/igs-kf.914 from 131.108.1.111 to flash ...

System flash directory:
File  Length   Name/status
  1   2251320  abc/igs-kf.914

[2251384 bytes used, 1942920 available, 4194304 total]
Accessing file 'abc/igs-kf.914' on 131.108.1.111...
Loading from 131.108.13.111:

Erasing device...  ... erased
Loading from 131.108.13.111:
!!.!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! [OK -
2251320/4194304 bytes]

Verifying checksum...  OK (0x97FA)
Flash copy took 79292 msecs
%FLH: Re-booting system after download
Loading abc/igs-kf.914 at 0x3000040, size = 2251320 bytes [OK]

F3: 2183364+67924+259584 at 0x3000060


              Restricted Rights Legend

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

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


Cisco Internetwork Operating System Software
Cisco IOS (tm) GS Software (GS7), Version 10.3
Copyright (c) 1986-1995 by cisco Systems, Inc.
Compiled Tue 06-Dec-94 14:01 by smith
Image text-base: 0x00001000, data-base: 0x005A9C94

cisco 2500 (68030) processor (revision 0x00) with 4092K/2048K bytes of
memory.
Processor board serial number 00000000
DDN X.25 software, Version 2.0, NET2 and BFE compliant.
ISDN software, Version 1.0.
Bridging software.
Enterprise software set supported.  (0x0)
1 Ethernet/IEEE 802.3 interface.
2 Serial network interfaces.
 --More--

1 ISDN Basic Rate interface.
32K bytes of non-volatile configuration memory.

4096K bytes of processor board System flash (Read ONLY)
```

# show microcode

To show the microcode bundled into a 7000 series system, use the **show microcode** EXEC command.

**show microcode**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show microcode** command:

```
Router# show micro
Microcode bundled in system

Card    Microcode    Target Hardware    Description
Type    Version      Version
----    ---------    ---------------    -----------
SP         2.3            11.x          SP version 2.3
EIP        1.1             1.x          EIP version 1.1
TRIP       1.2             1.x          TRIP version 1.2
FIP        1.4             2.x          FIP version 1.4
HIP        1.1             1.x          HIP version 1.1
SIP        1.1             1.x          SIP version 1.1
FSIP       1.1             1.x          FSIP version 1.1
```

# show version

Use the **show version** EXEC command to display the configuration of the system hardware, the software version, the names and sources of configuration files, and the boot images.

**show version**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show version** command from a Cisco 7000 series:

```
Router> show version

GS Software (GS7), Version 10.0
Copyright (c) 1986-1993 by cisco Systems, Inc.
Compiled Mon 11-Jan-93 14:44

System Bootstrap, Version 4.6(1)

Current date and time is Fri 2-26-1993 2:18:52
Boot date and time is Fri 1-29-1993 11:42:38
Router uptime is 3 weeks, 6 days, 14 hours, 36 minutes
System restarted by power-on
Running default software
Network configuration file is “Router”, booted via tftp from 131.108.2.333

RP1 (68040) processor with 16384K bytes of memory.
X.25 software.
Bridging software.
1 Switch Processor.
1 TRIP controller (4 Token Ring).
4 Token Ring/IEEE 802.5 interface.
1 AIP controller (1(ATM)
1 ATM network interface
4096K bytes of flash memory on embedded flash (in RP1).
Configuration register is 0x0
```

Table 3-7 describes significant fields shown in the display.

**Table 3-7 Show Version Field Descriptions**

| Field | Description |
|---|---|
| GS Software, Version 10.0 | Always specify the complete version number when reporting a possible software problem. In the example output, the version number is 10.0. |
| System Bootstrap, Version | Bootstrap version string. |

| Field | Description |
|---|---|
| Current date and time<br>Boot date and time<br>Router uptime is | Current date and time, the date and time the system was last booted, and *uptime*, or the amount of time the system has been up and running. |
| System restarted by power-on | Also displayed is a log of how the system was last booted, both as a result of normal system startup and of system error. For example, information can be displayed to indicate a bus error that is generally the result of an attempt to access a nonexistent address, as follows:<br>`System restarted by bus error at PC 0xC4CA, address 0x210C0C0` |
| Running default software | If the software was booted over the network, the Internet address of the boot host is shown. If the software was loaded from onboard ROM, this line reads "running default software." In addition, the names and sources of the host and network configuration files are shown. |
| RP1.... | The remaining output shows the hardware configuration and any nonstandard software options. The configuration register contents are displayed in hexadecimal notation. |

The output of the **show version** EXEC command can also provide certain messages, such as bus error messages. If such error messages appear, report the complete text of this message to your technical support specialist.

# tftp-server system

To specify that the router operate as a TFTP server, use the **tftp-server system** global configuration command. To remove a previously defined filename, use the **no tftp-server system** command with the appropriate filename and, optionally, the IP access-list number.

**tftp-server system** [**flash:**][*partition-number***:**]*filename* [*access-list-number*]
**no tftp-server system** *filename* [*access-list-number*]

## Syntax Description

| | |
|---|---|
| *filename* | Name you give the system image in Flash memory. |
| *access-list-number* | (Optional) IP access-list number. |
| **flash:** | (Optional) Specifies TFTP server operation from the file in the first partition of Flash memory. |
| *partition-number***:** | (Optional) Specifies TFTP server operation from the file in the specified partition of Flash memory. If the partition number is not specified, the file in the first partition is used. |

## Default

Disabled

## Command Mode

Global configuration

## Usage Guidelines

You can specify multiple filenames by repeating the **tftp-server system** command. The system sends a copy of the system image contained in ROM or one of the system images contained in Flash memory to any host that issues a TFTP read request with this filename.

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

- If you omit *filename* from the **tftp-server system** command, the TFTP request is rejected.
- If the specified *filename* exists in Flash memory, a copy of the Flash image is sent.
- On all systems but the Cisco 4500, if the specified *filename* is not found in Flash memory, the ROM image is sent.

Images that run from ROM cannot be loaded over the network. Therefore, it does not make sense to use TFTP to offer the ROMs on these images.

## Examples

Assuming there is a file in Flash memory named *version-9.0*, the following example causes the router to send, via TFTP, a copy of the Flash software when it receives a TFTP read request for the file *version-9.0*. The requesting host is checked against access list 22.

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

The following example causes the router to send, via TFTP, a copy of the file *flash:2:igs-bpx-1* when the requesting side specifies the name *flash:2:igs-bpx-l*:

```
tftp-server system flash:2:igs-bpx-l
```

## Related Command

A dagger (†) indicates that the command is documented in another chapter.

**access-list** †

# verify flash

To verify the checksums of files in Flash memory, use the **verify flash** EXEC command.

**verify flash**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

This command performs the same action as the **copy verify flash** command.

## Example

The following example illustrates how to use this command:

```
Router# verify flash

System flash partition information:
Partition   Size     Used    Free    Bank-Size   State       Copy-Mode
    1       4096K    2048K   2048K   2048K       Read Only   RXBOOT-FLH
    2       4096K    2048K   2048K   2048K       Read/Write  Direct

[ Type ?<no> for partition directory; ? for full directory; q to abort]
```

The system will prompt only if there are two or more read/write partitions. If the partition entered is not valid, the process terminates. You can enter a partition number, **?** for directory display of all partitions, or **?***number* for directory display of a particular partition. The default is the first partition.

```
File  Length   Name/status
  1   3459720  master/igs-bfpx.100-4.3
[3459784 bytes used, 734520 available, 4194304 total]

Name of file to verify? master/igs-bfpx.100-4.3
Verifying checksum for 'master/igs-bfpx.100-4.3' (file # 1)...  OK
```

# write erase

To erase the configuration information in NVRAM, use the **write erase** EXEC command.

**write erase**

## Syntax Description

This command has no arguments or keywords.

**Command Mode**

EXEC

## Example

The following example illustrates how to erase the configuration in NVRAM:

```
write erase
```

# write memory

To copy the current configuration information to NVRAM, use the **write memory** EXEC command.

**write memory**

## Syntax Description

This command has no arguments or keywords.

**Command Mode**

EXEC

## Usage Guidelines

Use the **write memory** command in conjunction with the **reload** command to restart the router with the configuration information stored in NVRAM.

If you issue the **write memory** command from a bootstrap system image, you receive a warning instructing you to indicate whether you want your previous NVRAM configuration to be overwritten and some configuration commands lost. This warning does not display if NVRAM contains an invalid configuration or if the previous configuration in NVRAM was generated by a bootstrap system image.

## Examples

The following example illustrates how to copy the current configuration information to NVRAM:

```
Router# write memory
```

The following is an example of the warning the system provides if you are trying to save configuration information from bootstrap into the system:

```
Router(boot)# write memory

Warning: Attempting to overwrite an NVRAM configuration written by a full system image.
This bootstrap software does not support a full configuration command set. If you write
memory now, some configuration commands may be lost.
Overwrite the previous NVRAM configuration? [confirm]
```

Enter **no** to escape writing the configuration information to memory.

## Related Commands

**configure**
**reload**
**show configuration**

# write network

To copy the current configuration information to a network server, use the **write network** EXEC command.

**write network**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

This command copies the current configuration to a server host on the network. You are prompted for a destination host and filename.

## Example

The following example illustrates how to begin the prompts for writing configuration information to a network host:

```
Router# write network
Remote host [0.0.0.0]? 131.108.1.111
Name of configuration file to write [Router-confg]?
Write file Router-confg on host 131.108.1.111? [confirm]
#
Writing Router-confg !! [OK]
Router#
```

# write terminal

To display the current configuration information on the terminal, use the **write terminal** EXEC command.

**write terminal**

## Syntax Description

This command has no arguments or keywords.

**Command Mode**

EXEC

## Usage Guidelines

Use this command in conjunction with the **show configuration** command to compare the information in running memory to the information stored in NVRAM.

## Example

The following example illustrates how to display the current configuration information:

```
write terminal
```

## Related Commands

**configure**
**show configuration**

CHAPTER 4

# Terminal Lines and Modem Commands

The line configuration commands described in this chapter are used to configure virtual terminal lines, the console port, and the auxiliary port.

For line configuration command descriptions, refer to the “Configuring Terminal Lines and Modem Support” chapter in the *Router Products Configuration Guide*.

The **history** line configuration command is described with other user interface commands in the “User Interface Commands” chapter of this manual. The **access-class** line configuration command, which applies an IP access list to a line, is described in the “Managing the System” chapter in the *Router Products Configuration Guide*.

The user-level EXEC commands that set terminal parameters for the duration of a session are documented in the *Cisco Access Connection Guide.*

# absolute-timeout

To set the interval for closing the connection, use the **absolute-timeout** line configuration command. Use the **no** form of this command to restore the default.

**absolute-timeout** *minutes*

## Syntax Description

| | |
|---|---|
| *minutes* | Number of minutes after which the user's session is terminated. |

## Default

No timeout interval is automatically set.

## Command Mode

Line configuration

## Usage Guidelines

This command terminates the connection after the specified time period has elapsed, regardless of whether or not the connection is being used at the time of termination. You can specify an absolute timeout value for each port. The user is given 20 seconds' notice before the session is terminated. You can use this command with the **logout-warning** command, which notifies the user of an impending logout.

---

**Note** You can set this command and an AppleTalk Remote Access (ARA) protocol time-out for the same line; however, this command supersedes any time-outs set in ARA protocol. Additionally, ARA protocol users receive no notice of any impending termination if this interval is set.

---

## Example

The following example sets an interval of 60 minutes on line 5:

```
line 5
absolute-timeout 60
```

## Related Command

**session-timeout**
**logout-warning**

# activation-character

To define the character you type at a vacant terminal to begin a terminal session, use the **activation-character** line configuration command. Use the **no** form of this command to make any character activate a terminal.

**activation-character** *ascii-number*
**no activation-character**

## Syntax Description

| | |
|---|---|
| *ascii-number* | Decimal representation of the activation character. |

## Default

Return (decimal 13).

## Command Mode

Line configuration

## Usage Guidelines

See the "ASCII Character Set" appendix for a list of ASCII characters.

---

**Note** If you are using **autoselect**, let the activation character default to Return and let the **exec-character-bits** command default to 7. If you change these defaults, the application does not recognize the activation request.

---

## Example

The following example sets the activation character for the console to Delete, which is decimal 127:

```
line console
activation-character 127
```

# autobaud

To set the line for automatic baud detection, use the **autobaud** line configuration command. Use the **no autobaud** command to restore the default.

**autobaud**
**no autobaud**

## Syntax Description

This command has no arguments or keywords.

## Default

No autobaud detection

## Command Mode

Line configuration

## Usage Guidelines

This command pertains to the auxiliary port only.

The autobaud detection supports a range from 300 to 19200 baud. A line set for autobaud cannot be used for outgoing connections. Nor can you set autobaud capability on a line using 19200 baud when the parity bit is set because of hardware limitations.

## Example

The following example sets the auxiliary port for autobaud detection:

```
line aux 0
autobaud
```

# autocommand

To configure the router to execute a command or list of commands automatically when a user connects to a particular line, use the **autocommand** line configuration command.

**autocommand** *command*

## Syntax Description

| | |
|---|---|
| *command* | Any appropriate EXEC command, including the host name and any switches that occur with the EXEC command. |

## Default

Automatic responses are not configured.

## Command Mode

Line configuration

## Usage Guidelines

This command applies to the auxiliary port only.

## Example

The following example forces an automatic connection to a host named host21 (which could be an IP address). In addition, the UNIX UUCP application specifies TCP socket 25, and the **/stream** switch enables a raw TCP stream with no Telnet control sequences.

```
line vty 4
autocommand connect host21 uucp /stream
```

# autohangup

To configure automatic line disconnect, use the **autohangup** line configuration command. The command causes the EXEC to issue the **exit** command when the last connection closes.

**autohangup**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Line configuration

## Usage Guidelines

This command is useful for UNIX UUCP applications that automatically disconnect lines because UUCP scripts cannot issue the **exit** command to hang up the telephone.

## Example

The following example enables automatic line disconnect on the auxiliary port:

```
line aux 0
autohangup
```

# autoselect

To configure a line to start an ARA, Point-to-Point Protocol (PPP), or SLIP session, use the **autoselect** line configuration command. Use the **no** form of this command to disable this function on a line.

**autoselect** {**arap** | **ppp** | **slip**} | **during-login**
**no autoselect**

## Syntax Description

| | |
|---|---|
| **arap** | Configures the router to allow an ARA session to start up automatically. |
| **ppp** | Configures the router to allow a PPP session to start up automatically. |
| **slip** | Configures the router to allow a SLIP session to start up automatically. |
| **during-login** | (Optional) The user receives a username and/or password prompt without pressing the Return key. After the user logs in, the autoselect function begins. |

## Default

Configures the router to allow an ARA session to start up automatically.

## Command Mode

Line configuration

## Usage Guidelines

This command eliminates the need for users to enter an EXEC command to start an ARA, PPP, or SLIP session.

---

**Note** SLIP does not support authentication. For PPP and ARA protocol, you must enable authentication.

---

The **autoselect** command configures the router to identify the type of connection being requested. For example, when a user on a Macintosh running ARA selects the Connect button, the router automatically starts an ARA protocol session. If, on the other hand, the user is running SLIP or PPP and uses the **autoselect ppp** or **autoselect slip** command, the router automatically starts a PPP or SLIP session, respectively. This command is appropriate for lines used to make different types of connections.

A line that does not have **autoselect** configured regards an attempt to open a connection as noise. Then when the router does not respond, the user client times out.

---

**Note** After the modem connection is established, a Return is required to evoke a response such as the username prompt. You might need to update your scripts to include this requirement. Additionally, let the activation character default to Return, and the **exec-character-bits** default to 7. If you change these defaults, the application does not recognize the activation request.

---

## Examples

The following example enables ARA on a line:

```
line 3
arap enable
autoselect arap
```

The following example enables PPP on a line:

```
line 7
autoselect ppp
```

The following example enables ARA on a line and allows logins from users with a modified CCL script and an unmodified script to log in:

```
line 3
arap enable
autoselect arap
autoselect during-login
arap noguest if-needed
```

## Related Commands

**ppp authentication chap**
**ppp authentication pap**
**arap use-tacacs**
**ppp use-tacacs**

# banner exec

To display a message on terminals with an interactive EXEC, use the **banner exec** global configuration command. This command specifies a message to be displayed on when an EXEC process is created (line activated, or incoming connection to VTY).

**banner exec** *d message d*

## Syntax Description

| | |
|---|---|
| *d* | Delimiting character of your choice—a pound sign (#), for example. You cannot use the delimiting character in the banner message. |
| *message* | Message text. |

## Default

Banners are not displayed.

## Command Mode

Global configuration

## Usage Guidelines

Follow the command with one or more blank spaces and a delimiting character of your choice. Then enter one or more lines of text, terminating the message with the second occurrence of the delimiting character.

## Example

The following example sets an EXEC message. The dollar sign ($) is used as a delimiting character.

```
banner exec $
Session activated. Enter commands at the prompt.
$
```

## Related Commands

**banner incoming**
**banner motd**
**exec-banner**

# banner incoming

To specify a message used when you have an incoming connection to a line from a host on the network, use the **banner incoming** global configuration command. An incoming connection is one initiated from the network side of the router. The EXEC banner can be suppressed on certain lines using the **no exec-banner** line configuration command. This line should *not* display the EXEC or MOTD banners when an EXEC is created.

**banner incoming** *d message d*

## Syntax Description

| | |
|---|---|
| *d* | Delimiting character of your choice—a pound sign (#), for example. You cannot use the delimiting character in the banner message. |
| *message* | Message text. |

## Default

No incoming banner is displayed.

## Command Mode

Global configuration

## Usage Guidelines

Follow the command with one or more blank spaces and a delimiting character of your choice. Then enter one or more lines of text, terminating the message with the second occurrence of the delimiting character.

## Example

The following example sets an incoming connection message. The pound sign (#) is used as a delimiting character.

```
banner incoming #
Welcome to Rhesus.
#
```

## Related Commands

**banner exec**
**banner motd**
**exec-banner**

# banner motd

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

**banner motd** *d message d*

## Syntax Description

| | |
|---|---|
| *d* | Delimiting character of your choice—a pound sign (#), for example. You cannot use the delimiting character in the banner message. |
| *message* | Message text. |

## Default

Disabled

## Command Mode

Global configuration

## Usage Guidelines

Follow the command with one or more blank spaces and a delimiting character of your choice. Then enter one or more lines of text, terminating the message with the second occurrence of the delimiting character.

This message-of-the-day banner is displayed to all terminals connected, and is useful for sending messages that affect all users; impending system shutdowns, for example.

The **banner** command without any keywords specified defaults to the **banner motd** command. When a new **banner motd** command is added to the configuration, it overwrites the existing **banner** command (no keyword specified). Similarly, if a **banner** command is added to the configuration, any exiting **banner motd** command is overwritten.

## Example

The following example sets a message-of-the-day banner. The pound sign (#) is used as a delimiting character.

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

## Related Commands

**banner exec**
**banner incoming**
**exec-banner**

# busy-message

To create a “host failed” message that displays when a connection fails, use the **busy-message** global configuration command. Use the **no busy-message** command to disable the “host failed” message from displaying on the specified host.

**busy-message** *hostname d message d*
**no busy-message** *hostname*

## Syntax Description

| | |
|---|---|
| *hostname* | Name of the host that cannot be reached. |
| *d* | Delimiting character of your choice—a pound sign (#), for example. You cannot use the delimiting character in the message. |
| *message* | Message text. |

## Default

The “host failed” message is not displayed.

## Command Mode

Global configuration

## Usage Guidelines

This command applies only to Telnet connections.

Follow the **busy-message** command with one or more blank spaces and a delimiting character of your choice. Then enter one or more lines of text, terminating the message with the second occurrence of the delimiting character.

Defining a “host failed” message for a host prevents all router-initiated user messages, including the initial message that indicates the connection is “Trying...” The **busy-message** command can be used in the **autocommand** command to suppress these messages.

## Example

The following example sets a message that will be displayed on the terminal whenever an attempt to connect to the host named dross fails. The pound sign (#) is used as a delimiting character.

```
busy-message dross #
Cannot connect to host. Contact the computer center.
#
```

# databits

To set the number of data bits per character that are interpreted and generated by hardware, use the **databits** line configuration command.

**databits** {**5** | **6** | **7** | **8**}

## Syntax Description

| | |
|---|---|
| **5** | Five data bits per character. |
| **6** | Six data bits per character. |
| **7** | Seven data bits per character. |
| **8** | Eight data bits per character. |

## Default

8 data bits per character

## Command Mode

Line configuration

## Usage Guidelines

This command pertains to the auxiliary port only.

The **databits** line configuration command can be used to mask the high bit on input from devices that generate 7 data bits with parity. If parity is being generated, specify 7 data bits per character. If no parity generation is in effect, specify 8 data bits per character. The other keywords are supplied for compatibility with older devices and generally are not used.

## Example

The following example changes the data bits to 7 on the auxiliary port:

```
line aux 0
databits 7
```

## Related Commands

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**terminal data-character-bits** ††
**terminal databits** ††

# data-character-bits

To set the number of data bits per character that are interpreted and generated by software, use the **data-character-bits** line configuration command.

**data-character-bits** {**7** | **8**}

## Syntax Description

| | |
|---|---|
| **7** | Seven data bits per character. |
| **8** | Eight data bits per character. |

## Default

8 data bits per character

## Command Mode

Line configuration

## Usage Guidelines

The **data-character-bits** line configuration command is used primarily to strip parity from X.25 connections on IGS or Cisco 3000 routers with the protocol translation software option. The **data-character-bits** line configuration command does not work on hardwired lines.

## Example

The following example sets the number of data bits per character for virtual terminal line 1 to 7:

```
line vty 1
data-character-bits 7
```

# default-value exec-character-bits

To define the EXEC character width for either 7 bits or 8 bits, use the **default-value exec-character-bits** global configuration command.

**default-value exec-character-bits** {**7** | **8**}

## Syntax Description

| | |
|---|---|
| **7** | Selects the 7-bit ASCII character set. |
| **8** | Selects the full 8-bit ASCII character set. |

**Default**

7-bit ASCII character set

## Command Mode

Global configuration

## Usage Guidelines

Configuring the EXEC character width to 8 bits allows you to add graphical and international characters in banners, prompts, and so forth. However, setting the EXEC character width to 8 bits can also cause failures. If a user on a terminal that is sending parity enters the command **help**, an “unrecognized command” message appears because the system is reading all 8 bits, although the eighth bit is not needed for the **help** command.

## Example

The following example selects the full 8-bit ASCII character set for EXEC banners and prompts:

```
default-value exec-character-bits 8
```

## Related Commands

Two daggers (††) indicate that the command is documented in the *Cisco Access Connnection Guide.*

**default-value special-character-bits**
**exec-character-bits**
**special-character-bits**
**terminal exec-character-bits** ††
**terminal special-character-bits** ††

# default-value special-character-bits

To configure the flow control default value from a 7-bit width to an 8-bit width, use the **default-value special-character-bits** global configuration command.

**default-value special-character-bits** {**7** | **8**}

## Syntax Description

| | |
|---|---|
| **7** | Selects the 7-bit character set. |
| **8** | Selects the full 8-bit character set. |

**Default**

7-bit character set

## Command Mode

Global configuration

## Usage Guidelines

Configuring the special character width to 8 bits allows you to add graphical and international characters in banners, prompts, and so forth.

## Example

The following example selects the full 8-bit special character set:

```
default-value special-character-bits 8
```

## Related Commands

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**default-value exec-character-bits**
**exec-character-bits**
**special-character-bits**
**terminal exec-character-bits** ††
**terminal special-character-bits** ††

# disconnect-character

To define a character to disconnect a session, use the **disconnect-character** line configuration command. This command defines the character you enter to end a terminal session. Use the **no disconnect-character** command to remove the disconnect character.

**disconnect-character** *ascii-number*
**no disconnect-character**

## Syntax Description

| | |
|---|---|
| *ascii-number* | ASCII decimal representation of the session disconnect character. |

## Default

No disconnect character is defined.

## Command Mode

Line configuration

## Usage Guidelines

The Break character is represented by zero; NULL cannot be represented.

To use the session disconnect character in normal communications, precede it with the escape character. See the "ASCII Character Set" appendix for a list of ASCII characters.

## Example

The following example sets the disconnect character for virtual terminal line 4 to Escape, which is ASCII character 27:

```
line vty 4
disconnect-character 27
```

# dispatch-character

To define a character that causes a packet to be sent, use the **dispatch-character** line configuration command. Use the **no dispatch-character** command to remove the definition of the specified dispatch character.

**dispatch-character** *ascii-number1* [*ascii-number2* . . . *ascii-number*]
**no dispatch-character** *ascii-number1* [*ascii-number2* . . . *ascii-number*]

## Syntax Description

| | |
|---|---|
| *ascii-number* | ASCII decimal representation of the character, such as Return (ASCII decimal 13) for line-at-a-time transmissions. |

## Default

No dispatch character is defined.

## Command Mode

Line configuration

## Usage Guidelines

This **dispatch-character** command defines a dispatch character that causes a packet to be sent even if the dispatch timer has not expired. It causes the router to attempt to buffer characters into larger-sized packets for transmission to the remote host. The router normally dispatches each character as it is typed.

This command can take multiple arguments, so you can define any number of characters as dispatch characters.

## Example

The following example specifies the Return character as the dispatch character:

```
line vty 4
dispatch-character 13
```

## Related Command

**dispatch-timeout**

# dispatch-timeout

To set the character dispatch timer, use the **dispatch-timeout** line configuration command. Use the **no dispatch-timeout** command to remove the timeout definition.

**dispatch-timeout** *milliseconds*
**no dispatch-timeout**

## Syntax Description

| | |
|---|---|
| *milliseconds* | Integer that specifies the number of milliseconds the router waits after putting the first character into a packet buffer before sending the packet. During this interval, more characters may be added to the packet, which increases the processing efficiency of the remote host. |

## Default

No dispatch timeout is defined.

## Command Mode

Line configuration

## Usage Guidelines

The **dispatch-timeout** line configuration command causes the router to buffer characters into packets for transmission to the remote host. The router sends a packet a specified amount of time after the first character is put in the buffer. The router normally dispatches each character as it is entered. You can use the **dispatch-timeout** and **dispatch-character** line configuration commands together. In this case, the router dispatches a packet each time the dispatch character is entered, or after the specified dispatch timeout interval, depending on which condition is met first.

---

**Note** The router's response might appear intermittent if the timeout interval is greater than 100 milliseconds and remote echoing is used.

---

## Example

The following example sets the dispatch timer to 80 milliseconds:

```
line vty 0 4
dispatch-timeout 80
```

## Related Command

**dispatch-characterr**

# escape-character

To define a system escape character, use the **escape-character** line configuration command. The **no escape-character** command sets the escape character to Break.

**escape-character** *ascii-number*
**no escape-character**

## Syntax Description

| | |
|---|---|
| *ascii-number* | Either the ASCII decimal representation of the character or a control sequence (Ctrl-E, for example). Ctrl-^ is the default. |

## Default

Ctrl-^

## Command Mode

Line configuration

## Usage Guidelines

The Break key cannot be used as an escape character on the console terminal because the operating software interprets Break as an instruction to halt the system. To send the escape character to the other side, press Ctrl-^ twice.

See the "ASCII Character Set" appendix for a list of ASCII characters.

## Example

The following example sets the escape character to Ctrl-P, which is ASCII character 16:

```
line console
escape-character 16
```

# exec

To allow an EXEC process on a line, use the **exec** line configuration command. The **no exec** command turns off the EXEC process for the line specified.

**exec**
**no exec**

## Syntax Description

This command has no arguments or keywords.

## Default

By default, the router starts EXECs on all lines.

## Command Mode

Line configuration

## Usage Guidelines

When you want to allow an outgoing connection *only* for a line, use the **no exec** command. When a user tries to Telnet to a line with the **no exec** command configured, the user will get no response when pressing the Return key at the login screen.

## Example

The following example illustrates how to turn off the EXEC on line 7. You might want to do this on the auxiliary port if the attached device (for example, the control port of a rack of modems) sends unsolicited data to the router. An EXEC would start if this happened, making the line unavailable.

```
line 7
no exec
```

# exec-banner

To control whether banners are displayed or suppressed, use the **exec-banner** line configuration command. This command determines whether the router will display the EXEC banner or the message-of-the-day (MOTD) banner when an EXEC is created. The **no exec-banner** command suppresses the banner messages.

**exec-banner**
**no exec-banner**

## Syntax Description

This command has no arguments or keywords.

## Default

By default, the messages defined with **banner motd** and **banner exec** commands are displayed on all lines.

## Command Mode

Line configuration

## Example

The following example suppresses the banner on virtual terminal lines 0 to 4:

```
line aux 0
no exec-banner
```

## Related Commands

**banner exec**
**banner motd**

# exec-character-bits

To configure the character widths of EXEC and configuration command characters, use the **exec-character-bits** line configuration command.

**exec-character-bits** {**7** | **8**}

## Syntax Description

| | |
|---|---|
| **7** | Selects the 7-bit character set. |
| **8** | Selects the full 8-bit character set for use of international and graphical characters in banner messages, prompts, and so forth. |

## Default

7-bit ASCII character set

## Command Mode

Line configuration

## Usage Guidelines

Setting the EXEC character width to 8 allows you to use special graphical and international characters in banners, prompts, and so forth. However, setting the EXEC character width to 8 bits can cause failures. If a user on a terminal that is sending parity enters the command **help**, an "unrecognized command" message appears because the system is reading all 8 bits, although the eighth bit is not needed for the **help** command.

---

**Note** If you are using the **autoselect** command, set the **activation-character** to the default Return and **exec-character-bits** to the default 7. If you change these defaults, the application does not recognize the activation request.

---

## Example

The following example allows full 8-bit international character sets by default, except for the console, which is an ASCII terminal. It illustrates use of the **default-value exec-character-bits** global configuration command and the **exec-character-bits** line configuration command.

```
default-value exec-character-bits 8
line 0
exec-character-bits 7
```

## Related Commands

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**default-value exec-character-bits**
**default-value special-character-bits**
**special-character-bits**
**terminal exec-character-bits** ††
**terminal special-character-bits** ††

# exec-timeout

To set the interval that the EXEC command interpreter waits until user input is detected, use the **exec-timeout** line configuration command. The **no exec-timeout** command removes the timeout definition.

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

## Syntax Description

| | |
|---|---|
| *minutes* | Integer that specifies the number of minutes. |
| *seconds* | (Optional) Additional time intervals in seconds. An interval of zero specifies no time-outs. |

## Default

10 minutes

## Command Mode

Line configuration

## Usage Guidelines

If no input is detected, the EXEC resumes the current connection, or if no connections exist, it returns the terminal to the idle state and disconnects the incoming session.

The **no** version of this command has the same effect as the **exec-timeout 0** command.

## Examples

The following example sets a time interval of 2 minutes, 30 seconds:

```
line console
exec-timeout 2 30
```

The following example sets a time interval of 10 seconds:

```
line console
exec-timeout 0 10
```

# flowcontrol

To set the method of data flow control between the terminal or other serial device and the router, use the **flowcontrol** line configuration command. To disable flow control, use the **no** form of this command.

**flowcontrol** {**none** | **software** [**in** | **out**] | **hardware** [**in** | **out**]}
**no flowcontrol** {**none** | **software** [**in** | **out**] | **hardware** [**in** | **out**]}

## Syntax Description

| | |
|---|---|
| **none** | Turns off flow control. |
| **software** | Sets software flow control. An optional keyword specifies the direction: **in** causes the router to listen to flow control from the attached device, and **out** causes the router to send flow control information to the attached device. If you do not specify a direction, both are assumed. |
| **hardware** | Sets hardware flow control. An optional keyword specifies the direction: **in** causes the router to listen to flow control from the attached device, and **out** causes the router to send flow control information to the attached device. If you do not specify a direction, both are assumed. For more information about hardware flow control, see the hardware installation and maintenance manual for your router. |

## Default

Flow control is disabled.

## Command Mode

Line configuration

## Usage Guidelines

This command pertains to the auxiliary port only.

When software flow control is set, the default stop and start characters are Ctrl-S and Ctrl-Q (XOFF and XON). You can change them with the **stop-character** and **start-character** commands.

## Example

The following example sets hardware flow control on the auxiliary port:

```
line aux 0
flowcontrol hardware
```

## Related Commands

**start-character**
**stop-character**

# hold-character

To define the local hold character used to pause output to the terminal screen, use the **hold-character** line configuration command. The **no hold-character** command restores the default.

**hold-character** *ascii-number*
**no hold-character**

## Syntax Description

| | |
|---|---|
| *ascii-number* | Either the ASCII decimal representation of the hold character or a control sequence (for example, Ctrl-P). |

## Default

No hold character is defined.

## Command Mode

Line configuration

## Usage Guidelines

The Break character is represented by zero; NULL cannot be represented. 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. See the "ASCII Character Set" appendix for a list of ASCII characters.

## Example

The following example sets the hold character to Ctrl-S, which is ASCII decimal 19:

```
line aux 0
hold-character 19
```

## Related Command

Two daggers (††) indicate that the command is documented in the *Cisco Access Connnection Guide.*

**terminal hold-character** ††

# length

To set the terminal screen length, use the **length** line configuration command.

**length** *screen-length*

## Syntax Description

| | |
|---|---|
| *screen-length* | Number of lines on the screen. A value of zero disables pausing between screens of output. |

## Default

24 lines

## Command Mode

Line configuration

## Usage Guidelines

Not all commands recognize the configured screen length. For example, the **show terminal** command assumes a screen length of 24 lines or more. The router software uses the value of this command to determine when to pause during multiple-screen output.

## Example

The following example illustrates how to disable the screen pause function on the console terminal:

```
line console
terminal-type VT220
length 0
```

# line

To configure a console port line, auxiliary port line, or virtual terminal lines, use the **line** global configuration command.

**line** [**aux** | **console** | **vty**] *line-number* [*ending-line-number*]

## Syntax Description

| | |
|---|---|
| **aux** | (Optional) Enables the auxiliary RS-232 DTE port. Must be addressed as relative line 0. The auxiliary port can be used for modem support and asynchronous connections. |
| **console** | (Optional) Specifies the console terminal line. The console port is DCE. |
| **vty** | (Optional) Specifies a virtual terminal for remote console access. |
| *line-number* | Specifies the relative number of the terminal line (or the first line in a contiguous group) you want to configure when the line type is specified. Numbering begins with zero. |
| *ending-line-number* | (Optional) Specifies the relative number of the last line in a contiguous group you want to configure. If you omit the keyword, then *line-number* and *ending-line-number* are absolute rather than relative line numbers. |

## Default

Lines are not configured.

## Command Mode

Global configuration

## Usage Guidelines

If you include one of the optional type keywords (**aux**, **console**, or **vty**), the line number is treated as a relative line number. If you enter the **line** command without an optional type keyword, the line number is treated as an absolute line number. Absolute line numbers increment consecutively and can be difficult to manage on large systems.

You can set communication parameters, specify autobaud connections, configure terminal operating parameters, and more for any of the terminal lines on the router.

The relative line number of the auxiliary port must be 0. See the **modem** line configuration command to set up modem support on the auxiliary port. The absolute line number of the auxiliary port is 1.

Virtual terminal lines are used to allow remote access to the router. A virtual terminal line is not associated with either the console or auxiliary port. You can address a single line or a consecutive range of lines with the **line** command. A line number is necessary, though, and you will receive an error message if you forget to include it.

## Examples

The following example starts configuration for virtual terminal lines 0 to 4:

```
line vty 0 4
```

The following example configures the auxiliary port with a line speed of 2400 baud and enables the EXEC:

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

## Related Commands

Two daggers indicate that the command is documented in the *Cisco Access Connection Guide*.

**show line**
**show users all** ††

# location

To record the location of a serial device, use the **location** line configuration command. The **no location** command removes the description.

**location** *text*
**no location**

## Syntax Description

| | |
|---|---|
| *text* | Location description. |

## Default

Locations of serial devices are not recorded.

## Command Mode

Line configuration

## Usage Guidelines

The **location** command enters information about the device location and status. Use the EXEC command **show users all** to display the location information.

## Example

The following example identifies the location of the console:

```
line console
location Building 3, Basement
```

## Related Command

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**show users all** ††

# lockable

To enable the EXEC command **lock**, use the **lockable** global configuration command The **no lockable** command reinstates the default, which does not allow the terminal to be locked.

**lockable**
**no lockable**

## Syntax Description

This command has no arguments or keywords.

## Default

Not lockable

## Command Mode

Global configuration

## Usage Guidelines

This command allows a terminal to be temporarily inaccessible by use of a temporary password.

## Example

The following example sets the terminal to the lockable state:

```
lockable
```

## Related Command

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**lock** ††

# login (line configuration)

To enable password checking at login, use the **login** line configuration command. Use the **no login** command to disable password checking and allow connections without a password.

**login** [**local** | **tacacs**]
**no login**

## Syntax Description

| | |
|---|---|
| **local** | (Optional) Selects local password checking. Authentication is based on the username specified with the **username** global configuration command. |
| **tacacs** | (Optional) Selects the TACACS-style user ID and password-checking mechanism. |

## Default

By default, virtual terminals require a password. If you do not set a password for a virtual terminal, it will respond to attempted connections by displaying an error message and closing the connection.

## Command Mode

Line configuration

## Usage Guidelines

If you specify **login** without the **local** or **tacacs** option, authentication is based on the password specified with the **password** line configuration command.

---

**Note** This command cannot be used with Authentication, Authorization, and Accounting (AAA)/TACACS+. Use the **login authentication** command instead.

---

## Examples

The following example sets the password *letmein* on virtual terminal line 4:

```
line vty 4
password letmein
login
```

The following example illustrates how to enable the TACACS-style user ID and password-checking mechanism:

```
line 0
password mypassword
login tacacs
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**enable password** †
**password**
**username** †

# login authentication

To enable AAA/TACACS+ authentication for logins, use the **login authentication** line configuration command. Use the **no** form of the command to return to the default.

**login authentication** {**default** | *list-name*}
**no login authentication** {**default** | *list-name*}

## Syntax Description

| | |
|---|---|
| **default** | Uses the default list created with the **aaa authentication login** command. |
| *list-name* | Uses the indicated list created with the **aaa authentication login** command. |

**Caution** If you use a *list-name* value that has not been configured with the **aaa authentication login** command, you will disable logins on this line.

## Default

Login authentication uses the default set with **aaa authentication login** command. If no default is set, the local user database is checked. No authentication is performed on the console.

## Command Mode

Line configuration

## Usage Guideline

This command is a per-line command used with AAA, and specifies the name of a list of TACACS+ authentication processes to try at login. If no list is specified, the default list is used (whether or not it is specified in the command line). You create defaults and lists by using the **aaa authentication login** command. Note that entering the **no** version of **login authentication** has the same effect as entering the command with the **default** argument.

Before issuing this command, create a list of authentication processes by using the global configuration **aaa authentication login** command.

## Examples

The following example specifies that the default AAA authentication is to be used on line 4:

```
line 4
login authentication default
```

The following example specifies that the AAA authentication list called *MIS-access* is to be used on line 7:

```
line 7
login authentication MIS-access
```

## Related Command

**aaa authentication login**

# login-string

To define a string of characters that the router sends to a host after a successful Telnet connection, use the **login-string** global configuration command. This command applies only to rlogin and Telnet sessions. The **no login-string** command removes the login string.

**login-string** *hostname d message* [**%***sec***p**] [**%***sec***w**] [**%b**] *d*
**no login-string** *hostname*

## Syntax Description

| | |
|---|---|
| *hostname* | Specifies the name of the host. |
| *d* | Sets a delimiting character of your choice—a pound sign (#) for example. You cannot use the delimiting character in the busy message. |
| *message* | Specifies the login string. |
| **%***sec***p** | (Optional) Sets a pause in seconds. To insert pauses into the login string, embed a percent sign (%) followed by the number of seconds to pause and the letter "p." |
| **%***sec***w** | (Optional) Prevents users from issuing commands or keystrokes during a pause. |
| **%b** | (Optional) Sends a Break character. |

## Default

No login strings are defined.

## Command Mode

Global configuration

## Usage Guidelines

Follow the command with one or more blank spaces and a delimiting character of your choice. Then enter one or more lines of text, terminating the message with the second occurrence of the delimiting character. To use a percent sign in the login string, precede it with another percent sign; that is, type the characters "%%." The options can be used anywhere within the message string.

## Example

In the following example, the value %5p causes a 5-second pause:

```
login-string office #ATDT 555-1234
%5p hello
#
```

# modem answer-timeout

To set the amount of time that the router waits for CTS after raising DTR in response to RING, use the **modem answer-timeout** line configuration command. The **no** form of this command reverts the router to the default value.

**modem answer-timeout** *seconds*
**no modem answer-timeout**

## Syntax Description

| | |
|---|---|
| *seconds* | Specifies the timeout interval in seconds. |

## Default

15 seconds

## Command Mode

Line configuration

## Usage Guidelines

This command applies to the auxiliary port only. It is useful for modems that take a long time to synchronize to the appropriate line speed.

## Example

The following example sets the timeout interval to 20 seconds:

```
line aux 0
modem answer-timeout 20
```

## Related Commands

**modem callin**
**modem in-out**

# modem callin

To support dial-in modems that use DTR to control the off-hook status of the modem, use the **modem callin** line configuration command. In response to RING, the modem raises the DTR signal, which answers the modem. At the end of the session, the router lowers DTR, which disconnects the modem. The **no** form of this command disables this feature.

**modem callin**
**no modem callin**

## Syntax Description

This command has no arguments or keywords.

## Default

No modem control

## Command Mode

Line configuration

## Usage Guidelines

This command applies to the auxiliary port only.

## Example

The following example causes the modem connected to the router to raise DTR in response to RING:

```
line aux 0
modem callin
```

## Related Commands

**modem answer-timeout**
**modem in-out**

# modem callout

To configure a line for reverse connections, use the **modem callout** line configuration command. The **no** form of this command disables this feature.

**modem callout**
**no modem callout**

## Syntax Description

This command has no arguments or keywords.

## Default

No modem control

## Command Mode

Line configuration

## Usage Guidelines

This command applies to the auxiliary port only and supports ports connected to computers that are designed to be connected to modems.

## Example

The following example configures the line for reverse connections:

```
line aux 0
modem callout
```

## Related Commands

**modem in-outt**
**rotary**

# modem cts-required

To configure a line to require a Clear To Send (CTS) signal, use the **modem cts-required** line configuration command. Use the **no** form of this command to disable this feature.

**modem cts-required**
**no modem cts-required**

## Syntax Description

This command has no arguments or keywords.

## Default

No modem control

## Command Mode

Line configuration

## Usage Guidelines

This command applies to the auxiliary port only. It supports lines that either the user or the network can activate. It is useful for closing connections from a user's terminal when the terminal is turned off and for preventing disabled printers and other devices in a rotary group from being considered.

## Example

The following example configures a line to require a CTS signal:

```
line aux 0
modem cts-required
```

## Related Command

**rotary**

# modem dtr-active

# telnet break-on-ip

To configure the router to generate a hardware Break signal upon receiving an Interrupt Process (IP) command, use the **telnet break-on-ip** line configuration command.

**telnet break-on-ip**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled.

## Command Mode

Line configuration

## Usage Guidelines

This command causes the system to generate a hardware Break signal on the RS-232 line that is associated with a reverse Telnet connection. It is useful when a Telnet Interrupt Process (IP) command is received on that connection because it can control the translation of Telnet IP commands into X.25 Break indications. It is also a useful workaround in the following situations:

- Several user Telnet programs send an IP command, but cannot send a Telnet break signal.
- Some Telnet programs implement a Break signal that sends an IP command.
- Some RS-232 hardware devices use a hardware Break signal for various purposes. A hardware Break signal is generated when a Telnet Break command is received.

## Example

In the following example, the auxiliary port is configured with the **telnet break-on-ip** command. The location text indicates that this refers to the high-speed modem.

```
line aux 0
location high-speed modem
telnet break-on-ip
```

## Related Commands

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**connect** ††
**telnet** (EXEC) ††
**terminal telnet break-on-ip** ††

# telnet refuse-negotiations

To configure a line using Telnet to refuse to negotiate full-duplex, remote echo options on incoming connections, use the **telnet refuse-negotiations** line configuration command.

**telnet refuse-negotiations**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled.

## Command Mode

Line configuration

## Usage Guidelines

This command is used on reverse Telnet connections to allow the router to refuse these requests from the other end. This command suppresses negotiation of the Telnet Remote Echo and Suppress Go Ahead options.

## Example

The following example shows how to set the auxiliary port to refuse full-duplex, remote echo requests:

```
line aux 0
telnet refuse-negotiations
```

## Related Commands

Two daggers (††) indicate that the command is documented in the *Cisco Access Connnection Guide*.

**connect** ††
**telnet** (EXEC) ††
**terminal telnet refuse-negotiations** ††

# telnet speed

To allow the router to negotiate transmission speed of the line to a connected device, use the **telnet speed** line configuration command.

**telnet speed** *default-speed maximum-speed*

## Syntax Description

| | |
|---|---|
| *default-speed* | Line speed (in bps) that the router will use if the device on the other end of the connection has not specified a speed. |
| *maximum-speed* | Maximum speed (in bps) that the device on the port will use. |

## Default

Disabled

## Command Mode

Line configuration

## Usage Guidelines

Negotiates speeds on reverse Telnet lines. You can match line speeds on remote systems in reverse Telnet, on host machines hooked up to a router to access the network, or on a group of console lines hooked up to the router, 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.

## Example

The following example allows the router to negotiate a bit rate on the line using the Telnet option. If no speed is negotiated, the line will run at 2400 bits per second. If the remote host requests a speed of greater than 9600 bps, then 9600 will be used.

```
line aux 0
telnet speed 2400 9600
```

## Related Commands

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**connect** ††
**telnet** (EXEC) ††
**terminal telnet speed**††

# telnet sync-on-break

To configure the router to cause an incoming connection to send a Telnet synchronize signal when it receives a Telnet Break signal, use the **telnet sync-on-break** line configuration command.

**telnet sync-on-break**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled.

## Command Mode

Line configuration

## Usage Guidelines

Causes a reverse Telnet line to send a Telnet Synchronize signal when it receives a Telnet Break signal. This option is used very rarely to ensure the ordering of break reception with respect to data characters sent after the break.

## Example

In the following example, the auxiliary port is configured with the **telnet sync-on-break** command:

```
line aux 0
telnet sync-on-break
```

## Related Commands

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**connect** ††
**telnet** (EXEC) ††
**terminal telnet sync-on-break** ††

# telnet transparent

To configure the router to send a carriage return (CR) as a CR followed by a NULL instead of a CR followed by a line feed (LF), use the **telnet transparent** line configuration command.

**telnet transparent**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled.

## Command Mode

Line configuration

## Usage Guidelines

This command is useful for coping with different interpretations of end-of-line handling in the Telnet protocol specification.

## Example

The following example causes the router, when sending a CR, to send a CR followed by a NULL character:

```
line aux 0
telnet transparent
```

## Related Commands

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**connect** ††
**telnet** (EXEC) ††
**terminal telnet transparent** ††

# terminal-type

To specify the type of terminal connected to a line, use the **terminal-type** line configuration command. The command records the type of terminal connected to the line. The **no terminal-type** command removes any information about the type of terminal and resets the line to the default terminal emulation.

**terminal-type** *terminal-name*
**no terminal-type**

## Syntax Description

| | |
|---|---|
| *terminal-name* | Terminal name and type. |

## Default

VT100

## Command Mode

Line configuration

## Usage Guidelines

The argument *terminal-name* provides a record of the terminal type and allows terminal negotiation of display management by hosts that provide that type of service.

## Example

The following example defines the terminal on the console as a type VT220:

```
line console
terminal-type VT220
```

## Related Command

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**terminal terminal-type** ††

# transport input

To allow the system administrator to define which protocols to use to connect to a specific line of the router, use the **transport input** line configuration command.

**transport input** {**mop** | **telnet** | **none**}

## Syntax Description

| | |
|---|---|
| **mop** | Selects the MOP protocol. |
| **telnet** | Specifies all types of incoming TCP/IP connections. |
| **none** | Prevents any protocol selection on the line. This makes the port unusable by incoming connections. |

## Default

Both protocols allowed on the line

## Command Mode

Line configuration

## Usage Guidelines

You can specify one protocol, multiple protocols, or else specify **none**.

This command can be useful in distributing resources among different types of users, or making certain that only specific hosts can access a particular port. When using protocol translation, the **transport input** command is also useful in controlling exactly which protocols can be translated to other protocols when using two-step translation.

Access lists for each individual protocol may be defined in addition to the allowances created by the **transport input** command.

## Example

The following example sets the preferred incoming protocol to Telnet:

```
line vty 0 32
transport input telnet
```

## Related Commands

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**terminal transport input** ††
**transport output**
**transport preferred**

# transport output

To determine the protocols that can be used for outgoing connections from a line, use the **transport output** line configuration command.

**transport output** {**telnet** | **none**}

## Syntax Description

| | |
|---|---|
| **telnet** | Selects the TCP/IP Telnet protocol. It allows a user at one site to establish a TCP connection to a login server at another site. |
| **none** | Prevents any protocol selection on the line. The system normally assumes that any unrecognized command is a host name. If the protocol is set to **none**, the system no longer makes that assumption. No connection will be attempted if the command is not recognized. |

## Default

Telnet

## Command Mode

Line configuration

## Example

The following example prevents any protocol selection:

```
transport output none
```

## Related Commands

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**terminal transport output** ††
**transport input**
**transport preferred**

# transport preferred

To specify the transport protocol the router uses if the user does not specify one when initiating a connection, use the **transport preferred** line configuration command.

**transport preferred** {**telnet** | **none**}

## Syntax Description

| | |
|---|---|
| **telnet** | Selects the TCP/IP Telnet protocol. It allows a user at one site to establish a TCP connection to a login server at another site. |
| **none** | Prevents any protocol selection on the line. The system normally assumes that any unrecognized command is a host name. If the protocol is set to **none**, the system no longer makes that assumption. No connection will be attempted if the command is not recognized. |

## Default

Telnet

## Command Mode

Line configuration

## Usage Guidelines

Specify **transport preferred none** to prevent errant connection attempts.

## Example

The following example sets the preferred protocol to Telnet on virtual terminal line 1:

```
line vty 1
transport preferred telnet
```

## Related Commands

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**terminal transport preferred** ††
**transport input**
**transport outputt**

# txspeed

To set the terminal transmit baud rate (to terminal), use the **txspeed** line configuration command.

**txspeed** *bps*

## Syntax Description

| | |
|---|---|
| *bps* | Baud rate in bits per second (bps); see Table 4-5 for settings. |

## Default

9600 bps

## Command Mode

Line configuration

## Usage Guidelines

Set the speed to match the baud rate of whatever device you have connected to the port. Some baud rates available on devices connected to the port might not be supported on the router. The router will indicate if the speed you select is not supported. Use Table 4-5 as a guide for setting the line speeds.

**Table 4-5 Router Line Speeds in Bits per Second**

| Router Model | Baud Rates |
|---|---|
| Cisco 7000, AGS+, CGS, MGS | 50, 75, 110, 134, 150, 200, 300, 600, 1050, 1200, 2000, 2400, 4800, 9600, 19200, 38400 |
| Cisco 2500 series, Cisco 3000, Cisco 4000 series | 75, 110, 134, 150, 300, 600, 1200, 2000, 2400, 4800, 1800, 9600, 19200, 38400 |

## Example

The following example sets the auxiliary line transmit speed to 2400 bps:

```
line aux 0
txspeed 2400
```

## Related Commands

**rxspeed**
**speed**

# vacant-message

To display an idle terminal message, use the **vacant-message** line configuration command. The command enables the banner to be displayed on the screen of an idle terminal. The **vacant-message** command without any arguments restores the default message. The **no vacant-message** command removes the default vacant message or any other vacant message that may have been set.

**vacant-message** [*d message d*]
**no vacant-message**

## Syntax Description

| | |
|---|---|
| *d* | (Optional) A delimiting character of your choice—a pound sign (#), for example. You cannot use the delimiting character in the banner message. |
| *message* | (Optional) Vacant terminal message. |

## Default

The format of the default vacant message is as follows:

```
<blank lines>
hostname tty# is now available
<blank lines>
Press RETURN to get started.
```

This message is generated by the system.

## Command Mode

Line configuration

## Usage Guidelines

Follow the command with one or more blank spaces and a delimiting character of your choice. Then enter one or more lines of text, terminating the message with the second occurrence of the delimiting character.

---

**Note** For a rotary group, you only need to define the message for the first line in the group.

---

## Example

The following example turns on the system banner and displays this message:

```
line 0
vacant-message #
                Welcome to Cisco Systems, Inc.
                 Press Return to get started.
#
```

# width

To set the terminal screen width, use the **width** line configuration command. This command sets the number of character columns displayed on the attached terminal.

**width** *characters*

## Syntax Description

| | |
|---|---|
| *characters* | Integer that specifies the number of character columns displayed on the terminal. |

## Default

80 character columns

## Command Mode

Line configuration

## Usage Guidelines

The rlogin protocol uses the *characters* argument to set up terminal parameters on a remote host.

Some hosts can learn the values for both length and width specified with the **line** and **width** commands.

## Example

The following example changes the character columns to 132 for the console terminal:

```
line console
location console terminal
width 132
```

## Related Command

Two daggers (††) indicate that the command is documented in the *Cisco Access Connection Guide*.

**terminal width** ††

CHAPTER 5

# System Management Commands

This chapter describes the commands used to manage the router system and its performance on the network. In general, system or network management falls into the following categories. The categories are described in this chapter unless specified otherwise.

- Configuration Management

  The configuration of network devices determines the behavior of the network. To manage device configurations, you need to list and compare configuration files on running devices, store configuration files on network servers for shared access, and perform software installations and upgrades. (Configuration management commands required to perform these tasks are described in the chapter entitled “System Image, Microcode Image, and Configuration File Load Commands.”)

  Other configuration management tasks include naming the router, setting router time services, configuring for synchronous logging of unsolicited messages and debug output, and configuring SNMP support. Configuration management commands required to perform these tasks are described this chapter.

- Security Management

  To manage security on the network, you need to restrict access to the system. You can do so on several different levels:

  — Assign passwords (and encrypt them) to restrict access to terminal lines, login connections, or privileged EXEC mode.

  — Establish one of three versions of Terminal Access Controller Access Control System (TACACS) protection for network servers that have shared access: TACACS, extended TACACS, or TACACS+, which is coupled with the Authentication, Authorization, and Accounting (AAA) model.

  — Restrict login connections to specific users with a username authentication system.

  — Control access on serial interfaces with Challenge Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP).

  — Create access lists to filter traffic to and from specific destinations. Subsequent chapters that describe the routing protocols in detail define access lists. This section provides general guidelines for creating access lists.

  — Create security labels for Internet Protocol (IP) datagrams using the Internet Protocol Security Option (IPSO), as described in the chapter entitled “IP Commands.”

- Enable accounting for Internet Protocol (IP) access list violations and display the accounting data. For information on the IP accounting access-violations feature and commands, see the “Configuring IP” chapter of the *Router Products Configuration Guide* and the “IP Commands” chapter later in this publication.

  Security management commands required to perform these tasks are described this chapter.

- Fault Management

  To manage network faults, you need to discover, isolate, and fix the problems. You can discover problems with the system’s monitoring commands, isolate problems with the system’s test commands, and resolve problems with other commands, including **debug**.

  This chapter describes general fault management commands. For detailed troubleshooting procedures and a variety of scenarios, see the *Troubleshooting Internetworking Systems* guide. For complete details on all **debug** commands, see the *Debug Command Reference* publication.

- System Performance Management

  To manage system performance, you need to monitor and determine response time, error rates, and availability. Once these factors are determined, you can perform load-balancing and modify system parameters to enhance performance. For example, priority queuing allows you to prioritize traffic order. You can configure fast and autonomous switching to improve network throughput, as described in the “Configuring Interfaces” chapter of the *Router Products Configuration Guide*.

  See the *Internetwork Design Guide* for additional information.

- Accounting Management

  Accounting management allows you to track both individual and group usage of network resources. You can then reallocate resources as needed. For example, you can change the system timers and configure TCP keepalives. See also the IP accounting feature in the “Configuring IP” chapter of the *Router Products Configuration Guide*. Additionally, the AAA/TACACS+ **aaa accounting** command allows you to set start-stop accounting for any or all of the listed functions for this command.

For system management configuration tasks and examples, refer to the chapter entitled “Managing the System” in the *Router Products Configuration Guide*.

# aaa accounting

To enable AAA accounting of requested services for billing or security purposes when using TACACS+, use the **aaa accounting** global configuration command. Use the **no** form of this command to disable accounting.

**aaa accounting** {**system** | **network** | **connection** | **exec** | **command** *level*} {**start-stop** | **wait-start** | **stop-only**} **tacacs+**
**no aaa accounting** {**system** | **network** | **connection** | **exec** | **command** *level*}

## Syntax Description

| | |
|---|---|
| **system** | Performs accounting for all system-level events not associated with users, such as reloads. |
| **network** | Runs accounting for all network-related service requests, including SLIP, PPP, PPP NCPs, and ARAP. |
| **connection** | Runs accounting for outbound Telnet and rlogin. |
| **exec** | Runs accounting for Execs (user shells). This keyword might return user profile information such as **autocommand** information. |
| **command** | Runs accounting for all commands at the specified privilege level. |
| *level* | Command level that should be accounted. Valid entries are 0 through 15. |
| **start-stop** | Sends a start record accounting notice at the beginning of a process and a stop record is sent at the end of a process. The start accounting record is sent in the background. The requested user process begins regardless of whether or not the start accounting record was received by the accounting server. |
| **wait-start** | As in **start-stop**, sends both a start and a stop accounting record to the accounting server. However, if you use the **wait-start** keyword, the requested user service does not begin until the start accounting record is acknowledged. A stop accounting record is also sent. |
| **stop-only** | Sends a stop record accounting notice at the end of the requested user process. |

## Default

AAA accounting is not enabled.

## Command Mode

Global configuration

## Usage Guideline

The **aaa accounting** command allows you to set start-stop accounting for any or all of the functions listed in “Syntax Description.” For minimal accounting control, issue the **stop-only** keyword, which sends a stop record accounting notice at the end of the requested user process. For additional accounting control, you can issue the **start-stop** command, where TACACS+ sends a start accounting notice at the beginning of the requested process and a stop accounting notice at the end of the process. You can further control access and accounting by issuing the **wait-start** command, which ensures that the start notice is received by the TACACS+ server before granting the user's process request. Accounting is done only to the TACACS+ server.

---

**Note** This command, along with **aaa authorization**, replaces the **tacacs-server authenticate** command in previous versions of TACACS, and can be used only with AAA/TACACS+.

---

## Examples

In the following example, accounting is set for outbound Telnet and rlogin, and both a start and stop accounting notice is sent to the TACACS+ server:

```
aaa accounting connection start-stop tacacs+
```

In the following example, accounting is set for privilege level 15 commands, with a wait-start restriction:

```
aaa accounting command 15 wait-start tacacs+
```

## Related Commands

**aaa authorization**
**aaa new-model**

# aaa authentication arap

To enable an AAA authentication method for ARA users using TACACS+, use the **aaa authentication arap** global configuration command. Use the **no** form of the command to disable this authentication.

**aaa authentication arap** {**default** | *list-name*} *method1* [...[*method4*]]
**no aaa authentication arap** {**default** | *list-name*} *method1* [...[*method4*]]

## Syntax Description

| | |
|---|---|
| **default** | Uses the listed methods that follow this argument as the default list of methods when a user logs in. |
| *list-name* | Character string used to name the following list of authentication methods tried when a user logs in. |
| *method* | One of the keywords described in Table 5-1. |

## Default

If the **default** list is not set, only the local user database is checked. This version has the same effect as the following command:

```
aaa authentication arap default local
```

## Command Mode

Global configuration

## Usage Guideline

The list names and default that you set using the **aaa authentication arap** command are used with the **arap authentication** command. These lists can contain up to four authentication methods that are used when a user tries to log in with ARA.

Create a list by entering the **aaa authentication arap** *list-name method* command, where *list-name* is any character string used to name this list, such as *MIS-access*. The *method* argument identifies the list of methods the authentication algorithm tries in the given sequence. You can enter up to four methods, which are described in Table 5-1.

To create a default list that is used if no list is specified in the **arap authentication** command, use the **default** keyword followed by the methods you wish to be used in default situations.

The additional methods of authentication are used only if the previous method returns an error, not if it fails.

Use the **write terminal** command to view lists of authentication methods.

**Table 5-1 AAA Authentication ARAP Method Descriptions**

| Keyword | Description |
|---|---|
| **if-needed** | Does not authenticate if the user has already been authenticated on a TTY line. |
| **line** | Uses the line password for authentication. |
| **local** | Uses the local username database for authentication. |
| **tacacs+** | Uses TACACS+ authentication. |

**Note** This command cannot be used with TACACS or extended TACACS.

## Examples

The following example creates a list called *MIS-access*, which first tries TACACS+ authentication and then none:

```
aaa authentication arap MIS-access tacacs+ none
```

The following example creates the same list, but sets it as the default list that is used for all ARA protocol authentications if no other list is specified:

```
aaa authentication arap default tacacs+ none
```

## Related Commands

**aaa authentication local-override**
**aaa new-model**
**arap authentication**

# aaa authentication enable default

To enable AAA authentication to determine if a user can access the privileged command level with TACACS+, use the **aaa authentication enable default** global configuration command. Use the **no** form of the command to disable this authorization method.

**aaa authentication enable default** *method1* [...[*method4*]]
**no aaa authentication enable default** *method1* [...[*method4*]]

## Syntax Description

| | |
|---|---|
| *method* | At least one and up to four of the keywords described in Table 5-2. |

## Default

If the **default** list is not set, only the enable password is checked. This version has the same effect as the following command:

```
aaa authentication enable default enable
```

On the console, the enable password is used if it exists. If no password is set, the process will succeed anyway.

## Command Mode

Global configuration

## Usage Guideline

Use the **aaa authentication enable default** command to create a series of authentication methods that are used to determine if a user can access privileged command level. You can specify up to four authentication methods. Method keywords are described in Table 5-2. The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

If a default authentication routine is not set for a function, the default is **none** and no authentication is performed. Use the **write terminal** command to view currently configured lists of authentication methods.

**Table 5-2 AAA Authentication Enable Default Method Descriptions**

| Keyword | Description |
|---|---|
| **enable** | Uses the enable password for authentication. |
| **line** | Uses the line password for authentication. |
| **none** | Uses no authentication. |
| **tacacs+** | Uses TACACS+ authentication. |

**Note** This command cannot be used with TACACS or extended TACACS.

## Example

The following example creates an authentication list that first tries to contact a TACACS+ server. If no server can be found, then AAA tries to use the enable password. If this also returns an error (because no enable password is configured on the server), the user is allowed access with no authentication.

```
aaa authentication enable default tacacs+ enable none
```

## Related Commands

**aaa authentication local-override**
**aaa authorization**
**aaa new-model**
**enable password**

# aaa authentication local-override

To have the router check the local user database for authentication before attempting another form of authentication, use the **aaa authentication local-override** global configuration command. Use the **no** form of the command to disable the override.

**aaa authentication local-override**
**no aaa authentication local-override**

## Syntax Description

This command has no arguments or keywords.

## Default

Override is disabled.

## Command Mode

Global configuration

## Usage Guideline

This command is useful when you want to configure an override to the normal authentication process for certain personnel such as system administrators.

When this override is set, the user is always prompted for the username. The system then checks to see if the entered username corresponds to a local account. If the username does not correspond to one in the local database, login proceeds with the methods configured with other **aaa** commands (such as **aaa authentication login**). Note when using this command that `Username:` is fixed as the first prompt.

## Example

The following example enables AAA authentication override:

```
aaa authentication local-override
```

## Related Commands

**aaa authentication arap**
**aaa authentication enable default**
**aaa authentication login**
**aaa authentication ppp**
**aaa new-model**

# aaa authentication login

To set AAA authentication at login when using TACACS+, use the **aaa authentication login** global configuration command. Use the **no** form of the command to disable AAA authentication.

**aaa authentication login** {**default** | *list-name*} *method1* [...[*method4*]]
**no aaa authentication login** {**default** | *list-name*} *method1* [...[*method4*]]

## Syntax Description

| | |
|---|---|
| **default** | Uses the listed authentication methods that follow this argument as the default list of methods when a user logs in. |
| *list-name* | Character string used to name the following list of authentication methods tried when a user logs in. |
| *method* | At least one and up to four of the keywords described in Table 5-3. |

## Default

If the **default** list is not set, only the local user database is checked. This version has the same effect as the following command:

```
aaa authentication login default local
```

**Note** On the console, login will succeed without any authentication checks if **default** is not set.

## Command Mode

Global configuration

## Usage Guideline

The default and optional list names that you create with the **aaa authentication login** command are used with the **login authentication** command.

Create a list by entering the **aaa authentication** *list-name method* command, where *list-name* is any character string used to name this list, such as *MIS-access*. The *method* argument identifies the list of methods the authentication algorithm tries, in the given sequence. Method keywords are described in Table 5-3.

To create a default list that is used if no list is assigned to a line with the **login authentication** command, use the default argument followed by the methods you want in default situations.

The additional methods of authentication are used only if the previous method returns an error, not if it fails. To ensure that the authentication will succeed even if all methods return an error, specify **none** as the final method in the command line.

If authentication is not specifically set for a line, the default is to deny access—no authentication is performed. Use the **write terminal** command to view currently configured lists of authentication methods.

**Table 5-3 AAA Authentication Login Method Descriptions**

| Keyword | Description |
|---|---|
| **enable** | Uses the enable password for authentication. |
| **line** | Uses the line password for authentication. |
| **local** | Uses the local username database for authentication. |
| **none** | Uses no authentication. |
| **tacacs+** | Uses TACACS+ authentication. |

**Note** This command cannot be used with TACACS or extended TACACS.

## Examples

The following example creates an AAA authentication list called *MIS-access*. This authentication first tries to contact a TACACS+ server. If no server is found, TACACS+ returns an error and AAA tries to use the enable password. If this also returns an error (because no enable password is configured on the server), the user is allowed access with no authentication.

```
aaa authentication login MIS-access tacacs+ enable none
```

The following example creates the same list, but sets it as the default list that is used for all login authentications if no other list is specified:

```
aaa authentication login default tacacs+ enable none
```

## Related Commands

**aaa authentication local-override**
**aaa new-model**
**login authentication**

# aaa authentication ppp

To specify one or more AAA authentication methods for use on serial interfaces running PPP when using TACACS+, use the **aaa authentication ppp** global configuration command. Use the **no** form of the command to disable authentication.

**aaa authentication ppp** {**default** | *list-name*} *method1* [...[*method4*]]
**no aaa authentication ppp** {**default** | *list-name*} *method1* [...[*method4*]]

## Syntax Description

| | |
|---|---|
| **default** | Uses the listed authentication methods that follow this argument as the default list of methods when a user logs in. |
| *list-name* | Character string used to name the following list of authentication methods tried when a user logs in. |
| *method* | At least one and up to four of the keywords described in Table 5-4. |

## Default

If the **default** list is not set, only the local user database is checked. This version has the same effect as the following command:

```
aaa authentication ppp default local
```

## Command Mode

Global configuration

## Usage Guideline

The lists that you create with the **aaa authentication ppp** command are used with the **ppp authentication** command. These lists contain up to four authentication methods that are used when a user tries to log in to the serial interface.

Create a list by entering the **aaa authentication ppp** *list-name method* command, where *list-name* is any character string used to name this list, such as *MIS-access*. The *method* argument identifies the list of methods the authentication algorithm tries in the given sequence. You can enter up to four methods. Method keywords are described in Table 5-4.

The additional methods of authentication are only used if the previous method returns an error, not if it fails. Specify **none** as the final method in the command line to have authentication succeed even if all methods return an error.

If authentication is not specifically set for a function, the default is **none** and no authentication is performed. Use the **write terminal** command to view lists of authentication methods.

**Table 5-4 AAA Authentication PPP Method Descriptions**

| Keyword | Description |
|---|---|
| **if-needed** | Does not authenticate if user has already been authenticated on a TTY line. |
| **local** | Uses the local username database for authentication. |
| **none** | Uses no authentication. |
| **tacacs+** | Uses TACACS+ authentication. |

**Note** This command cannot be used with TACACS or extended TACACS.

## Example

The following example creates an AAA authentication list called *MIS-access* for serial lines that use PPP. This authentication first tries to contact a TACACS+ server. If this action returns an error, the user is allowed access with no authentication.

```
aaa authentication MIS-access ppp tacacs+ none
```

## Related Commands

**aaa authentication local-override**
**aaa new-model**
**ppp authentication**

# aaa authorization

To set parameters that restrict a user's network access based on TACACS+ authorization, use the **aaa authorization** global configuration command. To disable authorization for a function, use the **no** form of the command.

**aaa authorization** {**network** | **connection** | **exec** | **command** *level*} *methods*
**no aaa authorization** {**network** | **connection** | **exec** | **command** *level*}

## Syntax Description

| | |
|---|---|
| **network** | Performs authorization for all network-related service requests, including SLIP, PPP, PPP NCPs, and ARAP. |
| **connection** | Runs authorization for outbound Telnet and rlogin. |
| **exec** | Runs authorization to determine if the user is allowed to run an Exec shell. This keyword might return user profile information such as **autocommand** information. |
| **command** | Runs authorization for all commands at the specified privilege level. |
| *level* | Specific command level that should be authorized. Valid entries are 0 through 15. |
| *methods* | Table 5-5 lists the *methods* keywords. |

## Default

Authorization is disabled for all actions (equivalent to the keyword *none*).

## Command Mode

Global configuration

## Usage Guideline

Use the **aaa authorization** command to create a list of one and up to four authorization methods that can be used when a user accesses the specified function.

The additional methods of authorization are only used if the previous method returns an error, not if it fails. Specify **none** as the final method in the command line to have authorization succeed even if all methods return an error.

**Table 5-5 AAA Authorization Method Descriptions**

| Keyword | Description |
|---|---|
| **tacacs+** | Requests authorization information from the TACACS+ server. |
| **if-authenticated** | Allows the user to access the requested function if the user is authenticated. |
| **none** | No authorization is performed. |
| **local** | Uses the local database for authorization. |

If authorization is not specifically set for a function, the default is **none** and no authorization is performed.

---

**Note** This command, along with **aaa accounting**, replaces the **tacacs-server** suite of commands in previous versions of TACACS.

---

## Examples

The following example specifies that TACACS+ authorization is used for all network-related requests. If this authorization method returns an error (if the TACACS+ server cannot be contacted), no authorization is performed and the request is successful.

```
aaa authorization network tacacs+ none
```

The following example specifies that TACACS+ authorization is run for level 15 commands. If this authorization method returns an error (if the TACACS+ server cannot be contacted), no authorization is performed and the request succeeds.

```
aaa authorization command 15 tacacs+ none
```

## Related Commands

**aaa accounting**
**aaa new-model**

# aaa new-model

To enable the AAA access control model that includes TACACS+, issue the **aaa new-model** global configuration command. Use the **no** form of the command to disable this functionality.

**aaa new-model**
**no aaa new-model**

## Syntax Description

This command has no arguments or keywords.

## Default

AAA/TACACS+ is not enabled.

## Command Mode

Global configuration

## Usage Guideline

This command enables the AAA access control system and TACACS+. If you initialize this functionality and later decide to use TACACS or extended TACACS, issue the **no** version of this command and then enable the version of TACACS you want to use.

## Example

The following example initializes AAA and TACACS+:

```
aaa new-model
```

## Related Commands

**aaa accounting**
**aaa authentication arap**
**aaa authentication enable default**
**aaa authentication local-override**
**aaa authentication login**
**aaa authentication ppp**
**aaa authorization**

# alias

To create a command alias, use the **alias** global configuration command. Use the **no alias** command to delete all aliases in a command mode or to delete a specific alias, and to revert to the original command syntax.

**alias** *mode alias-name alias-command-line*
**no alias** *mode* [*alias-name*]

## Syntax Description

| | |
|---|---|
| *mode* | Command mode of the original and alias commands. See Table 5-6 for a list of options for this argument. |
| *alias-name* | Command alias. |
| *alias-command-line* | Original command syntax. |

## Defaults

Default aliases are in EXEC mode as follows:

| Command Alias | Original Command |
|---|---|
| **h** | **help** |
| **lo** | **logout** |
| **p** | **ping** |
| **r** | **resume** |
| **s** | **show** |
| **w** | **where** |

## Command Mode

Global configuration

## Usage Guidelines

You can use simple words or abbreviations as aliases. The aliases in the Default section are predefined. They can be turned off using the **no alias** command.

Table 5-6 shows the acceptable options for the *mode* argument in the **alias** global configuration command.

**Table 5-6 Mode Argument Options**

| Argument Options | Mode |
|---|---|
| **configuration** | Global configuration |
| **controller** | Controller configuration |
| **exec** | EXEC |
| **hub** | Hub configuration |

| Argument Options | Mode |
|---|---|
| **interface** | Interface configuration |
| **ipx-router** | IPX router configuration |
| **line** | Line configuration |
| **map-class** | Map class configuration |
| **map-list** | Map list configuration |
| **route-map** | Route map configuration |
| **router** | Router configuration |

See the summary of command modes in the user interface chapter in the *Router Products Configuration Guide* for more information about command modes.

When you use online help, command aliases are indicated by an asterisk (*), as follows:

```
Router#lo?
*lo=logout  lock  login  logout
```

When you use online help, aliases that contain spaces (for example, *telnet device.cisco.com 25*) are displayed as follows:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#alias exec device-mail telnet device.cisco.com 25
Router(config)# end
Router# device-mail?
*device-mail="telnet device.cisco.com 25"
```

When you use online help, the alias is expanded and replaced with the original command, as shown in the following example with the *td* alias:

```
Router(config)#alias exec td trace device
Router(config)#^Z
Router#t?
*td="trace device"  telnet  terminal  test  tn3270
trace
```

To list only commands and omit aliases, begin your input line with a space. In the following example, the alias *td* is not shown, because there is a space before the **t?** command line.

```
Router# t?
telnet  terminal  test  tn3270  trace
```

As with commands, you can use online help to display the arguments and keywords that can follow a command alias. In the following example, the alias **td** is created to represent the command **telet device**. The **/debug** and **/line** switches can be added to **telnet device** to modify the command:

```
Router(config)# alias exec td telnet device
Router(config)# ^Z
Router#td ?
      /debug     Enable telnet debugging mode
      /line      Enable telnet line mode
      ...
      whois      Whois port
      <cr>

Router# telnet device
```

You must enter the complete syntax for the **alias** command. Partial syntax for aliases are not accepted. In the following example, the parser does not recognize the command *t* as indicating the alias *td*.

```
bones# t
% Ambiguous command:  "t"
```

## Example

In the following example, the alias *fixmyrt* is created for the EXEC-mode command **clear ip route 198.92.116.16**.

```
alias exec fixmyrt clear ip route 198.92.116.16
```

## Related Command

**show aliases**

# arap authentication

To enable TACACS+ authentication for ARA on a line, use the **arap authentication** line configuration command. Use the **no** form of the command to disable authentication for an ARA line.

**arap authentication** {**default** | *list-name*}
**no arap authentication** {**default** | *list-name*}

## Syntax Description

| | |
|---|---|
| **default** | Use the default list created with the **aaa authentication arap** command. |
| *list-name* | Use the indicated list created with the **aaa authentication arap** command. |

## Default

ARA protocol authentication uses the default set with **aaa authentication arap** command. If no default has been set, the local user database is checked.

## Command Mode

Line configuration

## Usage Guideline

This command is a per-line command that specifies the name of a list of AAA authentication methods to try at login. If no list is specified, the default list is used (whether or not it is specified in the command line). You create defaults and lists with the **aaa authentication arap** command. Entering the **no** version of **arap authentication** has the same effect as entering the command with the **default** argument.

Before issuing this command, create a list of authentication processes by using the **aaa authentication arap** global configuration command.

**Caution** If you use a *list-name* that was not configured with the **aaa authentication arap** command, ARA protocol will be disabled on this line.

## Example

The following example specifies that the TACACS+ authentication list called MIS-access is used on ARA line 7:

```
line 7
arap authentication MIS-access
```

## Related Command

**aaa authentication arap**

# buffers

Use the **buffers** global configuration command to make adjustments to initial buffer pool settings and to the limits at which temporary buffers are created and destroyed. Use the **no** form of this command to return the buffers to their default size.

**buffers** {**small** | **middle** | **big** | **verybig** | **large** | **huge** | *type number*}{**permanent** | **max-free** | **min-free** | **initial** } *number*

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

## Syntax Description

| | |
|---|---|
| **small** | Buffer size of this public buffer pool is 104 bytes. |
| **middle** | Buffer size of this public buffer pool is 600 bytes. |
| **big** | Buffer size of this public buffer pool is 1524 bytes. |
| **verybig** | Buffer size of this public buffer pool is 4520 bytes. |
| **large** | Buffer size is of this public buffer pool 5024 bytes. |
| **huge** | Default buffer size of this public buffer pool is 18024 bytes. This value can be configured with the **buffers huge size** command. |
| *type* | Interface type of the interface buffer pool. Value cannot be **fddi**. |
| *number* | Interface number of the interface buffer pool. |
| **permanent** | Number of permanent buffers that the system tries to create and keep. Permanent buffers are normally not trimmed by the system. |
| **max-free** | Maximum number of free or unallocated buffers in a buffer pool. |
| **min-free** | Minimum number of free or unallocated buffers in a buffer pool. |
| **initial** | Number of additional temporary buffers that are to be allocated when the system is reloaded. This keyword can be used to ensure that the system has necessary buffers immediately after reloading in a high-traffic environment. |
| *number* | Number of buffers to be allocated. |

## Default

The default number of buffers in a pool is determined by the hardware configuration and can be displayed with the EXEC **show buffers** command.

## Command Mode

Global configuration

## Usage Guidelines

Normally you need not adjust these parameters; do so only after consulting with technical support personnel. Improper settings can adversely impact system performance.

You cannot configure FDDI buffers.

## Examples of Public Buffer Pool Tuning

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

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

In the following example, the permanent buffer pool allocation for big buffers is increased to 200:

```
buffers big permanent 200
```

## Example of Interface Buffer Pool Tuning

A general guideline is to display buffers with the **show buffers all** command, observe which buffer pool is depleted, and increase that one.

In the following example, the permanent Ethernet 0 interface buffer pool on a Cisco 4000 is increased to 96 because the Ethernet 0 buffer pool is depleted:

```
buffers ethernet 0 permanent 96
```

## Related Commands

**buffers huge size**
**show buffers**

# buffers huge size

Use the **buffers huge size** global configuration command to dynamically resize all huge buffers to the value you specify. Use the **no buffers huge size** command to restore the default buffer values.

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

## Syntax Description

| | |
|---|---|
| *number* | Number of buffers to be allocated. |

## Default

18024 buffers

## Command Mode

Global configuration

## Usage Guidelines

Use only after consulting with technical support personnel. The buffer size cannot be lowered below the default.

## Example

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

```
buffers huge size 20000
```

## Related Commands

**buffers**
**show buffers**

# calendar set

To set the system calendar for a Cisco 7000 system or a Cisco 4500 system, use the **calendar set** EXEC command.

**calendar set** *hh:mm:ss day month year*
**calendar set** *hh:mm:ss month day year*

## Syntax Description

| | |
|---|---|
| *hh:mm:ss* | Current time in hours (military format), minutes, and seconds. |
| *day* | Current day (by date) in the month. |
| *month* | Current month (by name). |
| *year* | Current year (no abbreviation). |

## Command Mode

EXEC

## Usage Guidelines

Once you set the Cisco 7000 calendar or the Cisco 4500 calendar, the system clock will be automatically set when the system is restarted or when the **clock read-calendar** EXEC command is issued. The calendar maintains its accuracy, even after a power failure or system reboot has occurred. The time specified in this command is relative to the configured time zone.

## Example

In the following example, the system calendar is manually set to 1:32 p.m. on July 23, 1993:

```
calendar set 13:32:00 23 July 1993
```

## Related Commands

**clock read-calendar**
**clock set**
**clock summer-time**
**clock timezone**
**clock update-calendar**

# cdp enable

To enable Cisco Discovery Protocol (CDP) on an interface, use the **cdp enable** interface configuration command. Use the **no** form of this command to disable CDP on an interface.

**cdp enable**
**no cdp enable**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

CDP is enabled by default at the global level, but it must be enabled on each interface in order to send or receive CDP information.

## Example

In the following example, CDP is enabled on Ethernet interface 0:

```
interface ethernet 0
cdp enable
```

## Related Command

**cdp run**

# cdp holdtime

To specify the amount of time the receiving device should hold a CDP packet from your router before discarding it, use the **cdp holdtime** global configuration command. Use the **no** form of this command to revert to the default setting.

**cdp holdtime** *seconds*
**no cdp holdtime**

## Syntax Description

| | |
|---|---|
| *seconds* | Specifies the hold time to be sent in the CDP update packets. |

## Default

180 seconds

## Command Mode

Global configuration

## Usage Guidelines

CDP packets are sent with time-to-live, or hold time, that is nonzero after an interface is enabled and a hold time of 0 immediately before an interface is idled down.

The CDP hold time must be set to a higher number of seconds than the time between CDP transmissions, which is set using the **cdp timer** command.

## Example

In the following example, the CDP packets being sent from your device should be held by the receiving device for 60 seconds before being discarded. You might want to set the hold time lower than the default setting of 180 seconds if information about your device changes often and you want the receiving devices to purge this information more quickly.

```
cdp holdtime 60
```

## Related Commands

**cdp timer**
**show cdp**

# cdp run

To enable CDP on your router, use the **cdp run** global configuration command. Use the **no** form of this command to disable CDP.

**cdp run**
**no cdp run**

## Syntax Description

This command has no arguments or keywords.

## Default

Enabled

## Command Mode

Global configuration

## Usage Guidelines

CDP is enabled on your router by default, which means the router will receive CDP information. However, to receive CDP packets it must be enabled on interfaces, using the **cdp enable** interface configuration command.

## Example

In the following example, CDP is disabled for the router:

```
no cdp run
```

## Related Command

**cdp enable**

# cdp timer

To specify how often your router will send CDP updates, use the **cdp timer** global configuration command. Use the **no** form of this command to revert to the default setting.

**cdp timer** *seconds*
**no cdp timer**

## Syntax Description

| | |
|---|---|
| *seconds* | Specifies how often your router will send CDP updates. |

## Default

60 seconds

## Command Mode

Global configuration

## Usage Guidelines

The trade-off with sending more frequent transmissions is providing up-to-date information versus using bandwidth more often.

## Example

In the folowing example, CDP updates will be sent from your router every 80 seconds, less frequently than the default setting of 60 seconds. You might want to make this change if you are concerned about preserving bandwidth.

```
cdp timer 80
```

## Related Commands

**cdp holdtime**
**show cdp**

# clear cdp counters

To reset CDP traffic counters to zero (0) on your router, use the **clear cdp counters** privileged EXEC command.

**clear cdp counters**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Privileged EXEC

## Example

In the following example, the CDP counters have been cleared. The **show cdp traffic** output shows that all of the traffic counters have been reset to zero (0).

```
Router# clear cdp counters
Router# show cdp traffic

CDP counters :
        Packets output: 0, Input: 0
        Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
        No memory: 0, Invalid packet: 0, Fragmented: 0
```

## Related Commands

**clear cdp table**
**show cdp traffic**

# clear cdp table

To clear the table that contains CDP information about neighbors, use the **clear cdp table** privileged EXEC command.

**clear cdp table**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Privileged EXEC

## Example

In the following example, the CDP table is cleared. The output of the **show cdp neighbors** command shows that all information has been deleted from the table.

```
Router# clear cdp table

CDP-AD: Deleted table entry for neon.cisco.com, interface Ethernet0
CDP-AD: Deleted table entry for neon.cisco.com, interface Serial0
Router# show cdp neighbors


Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
```

## Related Commands

**clear cdp counters**
**show cdp neighbors**

# clock calendar-valid

To configure the Cisco 7000 series or the Cisco 4500 as a time source for a network based on its calendar, use the **clock calendar-valid** global configuration command. Use the **no** form of this command to set the router so that the calendar is not an authoritative time source.

**clock calendar-valid**
**no clock calendar-valid**

## Syntax Description

This command has no arguments or keywords.

## Default

Neither the Cisco 7000 nor the Cisco 4500 are not configured as a time source.

## Command Mode

Global configuration

## Usage Guidelines

Use this command if no outside time source is available.

## Example

In the following example, the Cisco 7000 is configured as the time source for a network based on its calendar:

```
clock calendar-valid
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**ntp master**
**vines time use-system** †

# clock read-calendar

To manually read the calendar into either the Cisco 7000 or the Cisco 4500 system clock, use the **clock read-calendar** EXEC command.

**clock read-calendar**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

When either the Cisco 7000 series or the Cisco 4500 calendar is rebooted, the calendar is automatically read into the system clock. However, you may use this command to manually read the calendar setting into the system clock. This command is useful if the **calendar set** command has been used to change the setting of the calendar.

## Example

In the following example, the system clock is configured to set its date and time by the calendar setting:

```
clock read-calendar
```

## Related Commands

**calendar set**
**clock set**
**clock update-calendar**
**ntp update-calendar**

# clock set

To manually set the system clock, use the **clock set** EXEC command.

**clock set** *hh:mm:ss day month year*
**clock set** *hh:mm:ss month day year*

## Syntax Description

| | |
|---|---|
| *hh:mm:ss* | Current time in hours (military format), minutes, and seconds. |
| *day* | Current day (by date) in the month. |
| *month* | Current month (by name). |
| *year* | Current year (no abbreviation). |

## Command Mode

EXEC

## Usage Guidelines

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 Cisco 7000 with calendar capability, you do not need to set the system clock. Use this command if no other time sources are available. The time specified in this command is relative to the configured time zone.

## Example

In the following example, the system clock is manually set to 1:32 p.m. on July 23, 1993:

```
clock set 13:32:00 23 July 1993
```

## Related Commands

**calendar set**
**clock read-calendar**
**clock summer-time**
**clock timezone**

# clock summer-time

To configure the system to automatically switch to summer time (daylight savings time), use one of the formats of the **clock summer-time** configuration command. Use the **no** form of this command to configure the router not to automatically switch to summer time.

**clock summer-time** *zone* **recurring** [*week day month hh:mm week day month hh:mm* [*offset*]]
**clock summer-time** *zone* **date** *date month year hh:mm date month year hh:mm* [*offset*]
**clock summer-time** *zone* **date** *month date year hh:mm month date year hh:mm* [*offset*]
**no clock summer-time**

## Syntax Description

| | |
|---|---|
| *zone* | Name of the time zone (PDT, ...) to be displayed when summer time is in effect. |
| *week* | Week of the month (1 to 5 or **last**). |
| *day* | Day of the week (Sunday, Monday, ...). |
| *date* | Date of the month (1 to 31). |
| *month* | Month (January, February, ...). |
| *year* | Year (1993 to 2035). |
| *hh:mm* | Time (military format) in hours and minutes. |
| *offset* | (Optional) Number of minutes to add during summer time (default is 60). |

## Default

Summer time is disabled. If **clock summer-time** *zone* **recurring** is specified without parameters, the summer time rules default to United States rules. Default of *offset* is 60.

## Command Mode

Global configuration

## Usage Guidelines

Use this command if you want to automatically switch to summer time (for display purposes only). Use the **recurring** form of the command if the local summer time rules are of this form. Use the **date** form to specify a start and end date for summer time if you cannot use the first form.

In both forms of the command, the first part of the command specifies when summer time begins, and the second part specifies when it ends. All times are relative to the local time zone. The start time is relative to standard time. The end time is relative to summer time. If the starting month is after the ending month, the system assumes that you are in the Southern Hemisphere.

## Examples

In the following example, summer time starts on the first Sunday in April at 02:00 and ends on the last Sunday in October at 02:00:

```
clock summer-time PDT recurring 1 Sunday April 2:00 last Sunday October 2:00
```

If you live in a place where summer time does not follow the pattern in the first example, you could set it to start on October 12, 1993 at 02:00, and end on April 28, 1994 at 02:00, with the following example:

```
clock summer-time date 12 October 1993 2:00 28 April 1994 2:00
```

## Related Commands

**calendar set**
**clock timezone**

# clock timezone

To set the time zone for display purposes, use the **clock timezone** global configuration command. To set the time to Coordinated Universal Time (UTC), use the **no clock timezone** command.

**clock timezone** *zone hours* [*minutes*]
**no clock timezone**

## Syntax Description

| | |
|---|---|
| *zone* | Name of the time zone to be displayed when standard time is in effect. |
| *hours* | Hours offset from UTC. |
| *minutes* | (Optional) Minutes offset from UTC. |

## Default

UTC

## Command Mode

Global configuration

## Usage Guidelines

The system internally keeps time in UTC, so this command is used only for display purposes and when the time is manually set.

## Example

In the following example, the timezone is set to Pacific Standard Time and is offset 8 hours behind UTC:

```
clock timezone PST -8
```

## Related Commands

**calendar set**
**clock set**
**clock summer-time**
**show clock**

# clock update-calendar

To set the Cisco 7000 or Cisco 4500 calendar from the system clock, use the **clock update-calendar** EXEC command.

**clock update-calendar**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

If the system clock and calendar are not synchronized, and the system clock is more accurate, use this command to update the Cisco 7000 series or Cisco 4500 calendar to the correct date and time.

## Example

In the following example, the current time is copied from the system clock to the Cisco 7000 calendar:

```
clock update-calendar
```

## Related Commands

**clock read-calendar**
**ntp update-calendar**

# custom-queue-list

To assign a custom queue list to an interface, use the **custom-queue-list** interface configuration command. To remove a specific list or all list assignments, use the **no** form of the command.

**custom-queue-list** *list*
**no custom-queue-list** [*list]*

## Syntax Description

| | |
|---|---|
| *list* | Number of the custom queue list you want to assign to the interface. An integer from 1 to 10. |

## Default

No custom queue list is assigned.

## Command Mode

Interface configuration

## Usage Guidelines

Only one queue list can be assigned per interface. Use this command in place of the **priority-list** command (not in addition to it). Custom queuing allows a fairness not provided with priority queuing. With custom queuing, you can control the interfaces' available bandwidth when it is unable to accommodate the aggregate traffic enqueued. Associated with each output queue is a configurable byte count, which specifies how many bytes of data should be delivered from the current queue by the system before the system moves on to the next queue. When a particular queue is being processed, packets are sent until the number of bytes sent exceeds the queue byte count or until the queue is empty.

## Example

In the following example, custom queue list number 3 is assigned to serial interface 0:

```
interface serial 0
custom-queue-list 3
```

## Related Commands

**queue-list default**
**queue-list interface**
**queue-list protocol**
**queue-list queue byte-count**
**queue-list queue limit**
**queue-list stun**

# enable

To log onto the router at a specifed level, use the **enable** EXEC command.

**enable** *level*

## Syntax Description

| | |
|---|---|
| *level* | (Optional) Privilege level to log in to on the router. |

## Default

Level 15

## Command Mode

EXEC

## Example

In the following example, the user is logging on to privilege level 5 on the router:

```
enable 5
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**privilege level**
**disable** †

# enable last-resort

To specify what happens if the TACACS servers used by the **enable** command do not respond, use the **enable last-resort** global configuration command. The **no** form of this command restores the default.

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

## Syntax Description

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

## Default

Default action is to fail.

## Command Mode

Global configuration

## Usage Guideline

The secondary authentication is used only if the first attempt fails. The secondary authentication does not occur if the first authentication is only unsuccessful.

---

**Note** This command is not used in AAA/TACACS+ and has been replaced by the **aaa authentication** suite of commands.

---

## Example

In the following example, if the TACACS servers do not respond to the **enable** command, the user can enable by entering the privileged level password:

```
enable last-resort password
```

## Related Command

A dagger (†) indicates that the command is documented in another chapter.

**enable** †

# enable password

To configure the enable password for a given level, use the **enable password** global configuration command. Use the **no** form of this command to remove the enable password for a given level.

**enable password** [**level** *level*] [*encryption-type*] *password*
**no enable password** [**level** *level*]

## Syntax Description

| | |
|---|---|
| *level* | (Optional) Level for which the password applies. You can specify up to sixteen privilege levels, using numbers 0 through 15. Level 1 is normal EXEC-mode user privileges. If this argument is not specified, the privilege leve ldefaults to 15 (traditional enable privileges). |
| *encryption-type* | (Optional) Type of password encryption. Can be 0 or 7. 0 indicates that the password that follows has not yet been encrypted. 7 indicates that the password has been encrypted using Cisco-proprietary encryption. |
| *password* | Password for the specified level or highest level if none is specified. |

## Default

No password is defined.

## Command Mode

Global configuration

## Usage Guidelines

Use this command with the **level** option to define a privilege level. Once the level and the password are specified, give the password to the users you want to have access at this level. Use the **privilege level (global)** configuration command to specifiy the commands that are accessible at the specified level.

You will not ordinarily enter an encryption type. Typically, you will only enter encryption type if you cut and paste a password that has already encrypted by the system back into this command.

Enable or disable password encryption with the **service password-encryption** command. If you enter a value for the the encryption-type argument, but have not enabled encryption, the encryption type will be treated as part of the password.

## Example

In the following example, the password *pswd2* is enabled for privilege level 2:

```
enable password level 2 pswd2
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**disable** †
**enable** †
**privilege level (global)**
**service password-encryption**
**show privilege**

# enable secret

To specify an additional layer of security over the **enable password** command, use the **enable secret** command. Use the **no** form of the command to turn off the enable secret function.

**enable secret** *password*
**no enable secret** *password*

## Syntax Description

| | |
|---|---|
| *password* | The **enable secret** password. This password should be different from the password created with the **enable password** command for additional security. |

## Default

Disabled

## Command Mode

Global configuration

## Usage Guidelines

The **enable secret** command is used in conjunction with the **enable password** command to provide an additional layer of security over the enable password. This process provides better security in two ways: first by enforcing the use of an additional password; second, by storing this second password using a non-reversible cryptographic function. This encryption method is especially useful in environments where the password crosses a network or is stored on a TFTP server.

If you use the same password for **enable password** and **enable secret**, you will receive an error message warning you that this practice is not recommended. The system will prompt you again for a password. You can reenter the password you use for enable password, and the system will accept it the second time. But if you do, you undermine the additional security that the **enable secret** command provides.

---

**Note** After you set a password using **enable secret**, a password set using the **enable password** command will no longer work unless enable secret is disabled or an older version of software is being used, such as when running an older rxboot image. Additionally, you cannot recover a lost password that has been encrypted by any method.

---

## Examples

The following example specifies an enable secret password of gobbledeegook:

```
enable secret gobbledeegook
```

After specifying an enable secret password, users must enter this password to gain access. Any passwords set through enable password will no longer work.

```
Password: gobbledeegook
```

# enable use-tacacs

To enable use of the TACACS to determine whether a user can access the privileged command level, use the **enable use-tacacs** global configuration command. Use the **no** form of this command to disable TACACS verification.

**enable use-tacacs**
**no enable use-tacacs**

**Caution** If you use the **enable use-tacacs** command, you must also use the **tacacs-server authenticate enable** command, or else you will be locked out of the router.

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Global configuration

## Usage Guidelines

When you add this command to the configuration file, the EXEC **enable** command prompts for a new username and password pair. This pair is then passed to the TACACS server for authentication. If you are using extended TACACS, it also will pass any existing UNIX user identification code to the server.

**Note** This command initializes TACACS. Use the **tacacs server-extended** command to initialize extended TACACS, or use the **aaa new-model** command to initialize AAA/TACACS+.

## Example

The following example sets TACACS verification on the privileged EXEC-level login sequence:

```
enable use-tacacs
tacacs-server authenticate enable
```

## Related Command

**tacacs-server authenticate enable**

# hostname

To specify or modify the host name for the network server, use the **hostname** global configuration command. The host name is used in prompts and default configuration filenames. The **setup** command facility also prompts for a host name at startup.

**hostname** *name*

## Syntax Description

| | |
|---|---|
| *name* | New host name for the network server; the name is case sensitive. |

## Default

The factory-assigned default host name is *router*.

## Command Mode

Global configuration

## Usage Guidelines

The order of display at startup is banner message-of-the-day (MOTD), then login and password prompts, then EXEC banner.

## Example

The following example changes the host name to *sandbox*:

```
hostname sandbox
```

# load-interval

To change the length of time for which data is used to compute load statistics, use the **load-interval** interface configuration command. Use the **no** form of this command to revert to the default setting.

**load-interval** *seconds*
**no load-interval** *seconds*

## Syntax Description

| | |
|---|---|
| *seconds* | Length of time for which data is used to compute load statistics. A value that is a multiple of thirty, between 30 and 600 (30, 60, 90, 120, and so forth). |

## Default

300 seconds (or 5 minutes)

## Command Mode

Interface configuration

## Usage Guidelines

If you want load computations to be more reactive to short bursts of traffic, rather than averaged over five-minute periods, you can shorten the length of time over which load averages are computed.

If the load interval is set to thirty seconds, new data is used for load calculations over a thirty-second period. This data is used to compute load statistics, including input rate in bits and packets per second, output rate in bits and packets per second, load, and reliability.

Load data is gathered every five seconds on the router. This data is used for a weighted average calculation in which more-recent load data has more weight in the computation than older load data. If the load interval is set to thirty seconds, the average is computed for the last thirty seconds of load data.

The **load-interval** command allows you to change the default interval of five minutes to a shorter or longer period of time. If you change it to a shorter period of time, the input and output statistics that are displayed when you use the **show interface** command will be more current, and based on more instantaneous data, rather than reflecting a more average load over a longer period of time.

This command is often used for dial backup purposes, to increase or decrease the likelihood of a backup interface being implemented, but it can be used on any interface.

## Example

In the following example, the default five-minute average is set it to a thirty-second average. A burst in traffic that would not trigger a dial backup for an interface configured with the default five-minute interval might trigger a dial backup for this interface that is set for a shorter, thirty-second interval.

```
interface serial 0
load-interval 30
```

## Related Command

A dagger (†) indicates that the command is documented in another chapter.

**show interfaces** †

# logging

To log messages to a syslog server host, use the **logging** global configuration command. The **no logging** command deletes the syslog server with the specified address from the list of syslogs.

**logging** *host*
**no logging** *host*

## Syntax Description

| | |
|---|---|
| *host* | Name or IP address of the host to be used as a syslog server. |

## Default

No messages are logged to a syslog server host.

## Command Mode

Global configuration

## Usage Guidelines

This command identifies a syslog server host to receive logging messages. By issuing this command more than once, you build a list of syslog servers that receive logging messages.

## Example

The following example logs messages to a host named *johnson*:

```
logging johnson
```

## Related Commands

**logging trap**
**service timestamps**

# logging buffered

To log messages to an internal buffer, use the **logging buffered** global configuration command. The **no logging buffered** command cancels the use of the buffer and writes messages to the console terminal, which is the default.

**logging buffered**
**no logging buffered**

## Syntax Description

This command has no arguments or keywords.

## Default

The router displays all messages to the console terminal.

## Command Mode

Global configuration

## Usage Guidelines

This command copies logging messages to an internal buffer instead of writing them to the console terminal. The buffer is circular in nature, so newer messages overwrite older messages.

To display the messages that are logged in the buffer, use the EXEC command **show logging**. The first message displayed is the oldest message in the buffer.

## Example

The following example illustrates how to enable logging to an internal buffer:

```
logging buffered
```

# logging console

To limit messages logged to the console based on severity, use the **logging console** global configuration command. The **no logging console** command disables logging to the console terminal.

**logging console** *level*
**no logging console**

## Syntax Description

| | |
|---|---|
| *level* | Limits the logging of messages displayed on the console terminal to the named level. See Table 5-7 for a list of the *level* keywords. |

## Default

**debugging**

## Command Mode

Global configuration

## Usage Guidelines

Specifying a *level* causes messages at that level and numerically lower levels to be displayed at the console terminal.

The EXEC command **show logging** displays the addresses and levels associated with the current logging setup, as well as any other logging statistics.

**Table 5-7 Error Message Logging Priorities**

| Level Name | 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 |

## Example

The following example changes the level of messages displayed to the console terminal to **alerts**, which means alerts and emergencies are displayed:

```
logging console alerts
```

## Related Command

**logging facility**

# logging facility

To configure the syslog facility in which error messages are sent, use the **logging facility** global configuration command. To revert to the default of local7, use the **no logging facility** global configuration command.

**logging facility** *facility-type*
**no logging facility**

## Syntax Description

| | |
|---|---|
| *facility-type* | Syslog facility. See Table 5-8 for the *facility-type* keywords. |

## Default

**local7**

## Command Mode

Global configuration

## Usage Guidelines

Table 5-8 describes the acceptable options for the *facility-type* keyword.

**Table 5-8 Logging Facility Facility-Type Keywords**

| Keyword | Description |
|---|---|
| **auth** | Authorization system |
| **cron** | Cron facility |
| **daemon** | System daemon |
| **kern** | Kernel |
| **local0–7** | Reserved for locally defined messages |
| **lpr** | Line printer system |
| **mail** | Mail system |
| **news** | USENET news |
| **sys9** | System use |
| **sys10** | System use |
| **sys11** | System use |
| **sys12** | System use |
| **sys13** | System use |
| **sys14** | System use |
| **syslog** | System log |
| **user** | User process |
| **uucp** | UNIX-to-UNIX copy system |

## Example

The following example configures the syslog facility to Kernel:

```
logging facility kern
```

## Related Command

**logging console**

# logging monitor

To limit messages logged to the terminal lines (monitors) based on severity, use the **logging monitor** global configuration command. This command limits the logging messages displayed on terminal lines other than the console line to messages with a level at or above *level*. The **no logging monitor** command disables logging to terminal lines other than the console line.

**logging monitor** *level*
**no logging monitor**

## Syntax Description

| | |
|---|---|
| *level* | One of the *level* keywords listed in Table 5-7. |

## Default

**debugging**

## Command Mode

Global configuration

## Usage Guidelines

Specifying a *level* causes messages at that level and numerically lower levels to be displayed to the monitor.

## Example

The following example specifies that only messages of the levels **errors**, **critical**, **alerts**, and **emergencies** be displayed on terminals:

```
logging monitor errors
```

## Related Command

A double dagger (††) indicates that the command is documented in the *Cisco Access Connection Guide* publication.

**terminal monitor** ††

# logging on

To control logging of error messages, use the **logging on** global configuration command. This command enables or disables message logging to all destinations except the console terminal. The **no logging on** command enables logging to the console terminal only.

**logging on**
**no logging on**

## Syntax Description

This command has no arguments or keywords.

## Default

The router logs messages to the console terminal.

## Command Mode

Global configuration

## Example

The following example shows how to direct error messages to the console terminal only:

```
no logging on
```

# logging synchronous

To synchronize unsolicited messages and **debug** output with solicited router output and prompts for a specific console port line, auxiliary port line, or virtual terminal line, use the **logging synchronous** line configuration command. Use the no form of the command to disable synchronization of unsolicited messages and debug output.

**logging synchronous** [**level** *severity-level* | **all**] [**limit** *number-of-buffers*]
**no logging synchronous** [**level** *severity-level* | **all**] [**limit** *number-of-buffers*]

## Syntax Description

| | |
|---|---|
| **level** *severity-level* | (Optional) Specifies the message severity level. Messages with a severity level equal to or higher than this value are printed asynchronously. When specifying a severity level number, consider that for the logging system, low numbers indicate greater severity and high numbers indicate lesser severity. The default value is 2. |
| **all** | (Optional) Specifies that all messages are printed asynchronously, regardless of the severity level. |
| **limit** *number-of-buffers* | (Optional) Specifies the number of buffers to be queued for the terminal after which new messages are dropped. The default value is 20. |

## Defaults

This feature is turned off by default.

If you do not specify a severity level, the default value of 2 is assumed.

If you do not specify the maximum number of buffers to be queued, the default value of 20 is assumed.

## Command Mode

Line configuration

## Usage Guidelines

When synchronous logging of unsolicited messages and **debug** output is turned on, unsolicited router output is displayed on the console or printed after solicited router output is displayed or printed. Unsolicited messages and **debug** output is displayed on the console after the prompt for user input is returned. This is to keep unsolicited messages and **debug** output from being interspersed with solicited router output and prompts. After the unsolicited messages are displayed, the console displays the user prompt again.

When specifying a severity level number, consider that for the logging system, low numbers indicate greater severity and high numbers indicate lesser severity.

When a terminal line's message-queue limit is reached, new messages are dropped from the line, although these messages might be displayed on other lines. If messages are dropped, the notice "%SYS-3-MSGLOST *number-of-messages* due to overflow" follows any messages that are displayed. This notice is displayed only on the terminal that lost the messages. It is not sent to any other lines, any logging servers, or the logging buffer.

**Caution** By configuring abnormally large message-queue limits and setting the terminal to "terminal monitor" on a terminal that is accessible to intruders, you expose yourself to "denial of service" attacks. An intruder could carry out the attack by putting the terminal in synchronous output mode, making a Telnet connection to a remote host, and leaving the connection idle. This could cause large numbers of messages to be generated and queued, and these messages would consume all available RAM. Although unlikely to occur, you should guard against this type of attack through proper configuration.

## Example

The following example identifies line 4 and enables synchronous logging for line 4 with a severity level of 6. Then the example identifies another line, line 2, and enables synchronous logging for line 2 with a severity level of 7 and specifies a maximum number of buffers to be 70000:

```
line 4
logging synchronous level 6
line 2
logging synchronous level 7 limit 70000
```

## Related Command

A dagger (†) indicates that the command is documented in another chapter.

**line**†

# logging trap

To limit messages logged to the syslog servers based on severity, use the **logging trap** global configuration command. The command limits the logging of error messages sent to syslog servers to only those messages at the specified level. The **no logging trap** command disables logging to syslog servers.

**logging trap** *level*
**no logging trap**

## Syntax Description

| | |
|---|---|
| *level* | One of the *level* keywords listed in Table 5-7. |

## Default

**informational**

## Command Mode

Global configuration

## Usage Guidelines

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

Table 5-7 lists the syslog definitions that correspond to the debugging message levels. Additionally, there are four categories of messages generated by the software, as follows:

- Error messages about software or hardware malfunctions at the LOG_ERR level.
- Output for the debug commands at the LOG_WARNING level.
- Interface up/down transitions and system restarts at the LOG_NOTICE level.
- Reload requests and low process stacks are at the LOG_INFO level.

Use the **logging** and **logging trap** commands to send messages to a UNIX syslog server.

## Example

The following example logs messages to a host named *johnson*:

```
logging johnson
logging trap notifications
```

## Related Command

**logging**

# login authentication

To enable TACACS+ authentication for logins, use the **login authentication** line configuration command. Use the **no** form of the command to return to the default.

**login authentication** {**default** | *list-name*}
**no login authentication** {**default** | *list-name*}

**Caution** If you use a *list-name* value that was not configured with the **aaa authentication login** command, you will disable login on this line.

## Syntax Description

| | |
|---|---|
| **default** | Uses the default list created with the **aaa authentication login** command. |
| *list-name* | Uses the indicated list created with the **aaa authentication login** command. |

## Default

Uses the default set with **aaa authentication login**.

## Command Mode

Line configuration

## Usage Guideline

This command is a per-line command used with AAA that specifies the name of a list of TACACS+ authentication methods to try at login. If no list is specified, the default list is used (whether or not it is specified in the command line). You create defaults and lists with the **aaa authentication login** command. Entering the **no** version of **login authentication** has the same effect as entering the command with the **default** argument.

Before issuing this command, create a list of authentication processes by using the global configuration **aaa authentication login** command.

## Examples

The following example specifies that the default AAA authentication is to be used on line 4:

```
line 4
login authentication default
```

The following example specifies that the AAA authentication list called MIS-access is to be used on line 7:

```
line 7
login authentication MIS-access
```

### Related Command

**aaa authentication login**

# ntp access-group

To control access to the system's Network Time Protocol (NTP) services, use the **ntp access-group** global configuration command. To remove access control to the system's NTP services, use the **no ntp access-group** command.

**ntp access-group** {**query-only** | **serve-only** | **serve** | **peer**} *access-list-number*
**no ntp access-group** {**query-only** | **serve-only** | **serve** | **peer**}

## Syntax Description

| | |
|---|---|
| **query-only** | Allows only NTP control queries. See RFC 1305 (NTP version 3). |
| **serve-only** | Allows only time requests. |
| **serve** | Allows time requests and NTP control queries, but does not allow the system to synchronize to the remote system. |
| **peer** | Allows time requests and NTP control queries and allows the system to synchronize to the remote system. |
| *access-list-number* | Number (1 to 99) of a standard IP access list. |

## Default

No access control (full access granted to all systems)

## Command Mode

Global configuration

## Usage Guidelines

The access group options are scanned in the following order from least restrictive to most restrictive:

1. peer
2. serve
3. serve-only
4. query-only

Access is granted for the first match that is found. If no access groups are specified, all access is granted to all sources. If any access groups are specified, only the specified access is granted. This facility provides minimal security for the time services of the system. However, it can be circumvented by a determined programmer. If tighter security is desired, use the NTP authentication facility.

## Example

In the following example, the system is configured to allow itself to be synchronized by a peer from access list 99. However, the system restricts access to allow only time requests from access list 42.

```
ntp access-group peer 99
ntp access-group serve-only 42
```

## Related Command

A dagger (†) indicates that the command is documented in another chapter.

**access-list** †

# ntp authenticate

To enable Network Time Protocol (NTP) authentication, use the **ntp authenticate** global configuration command. Use the **no** form of this command to disable the feature.

**ntp authenticate**
**no ntp authenticate**

## Syntax Description

This command has no keywords or arguments.

## Default

No authentication

## Command Mode

Global configuration

## Usage Guidelines

Use this command if you want authentication. If this command is specified, the system will not synchronize to a system unless it carries one of the authentication keys specified in the **ntp trusted-key** command.

## Example

The following example enables NTP authentication:

```
ntp authenticate
```

## Related Commands

**ntp authentication-key**
**ntp trusted-key**

# ntp authentication-key

To define an authentication key for Network Time Protocol (NTP) , use the **ntp authentication-key** global configuration command. Use the **no** form of this command to remove the authentication key for NTP.

**ntp authentication-key** *number* **md5** *value*
**no ntp authentication-key** *number*

## Syntax Description

| | |
|---|---|
| *number* | Key number (1 to 4294967295). |
| *value* | Key value (an arbitrary string of up to eight characters). |

## Default

No authentication key is defined for NTP.

## Command Mode

Global configuration

## Usage Guidelines

Use this command to define authentication keys for use with other NTP commands in order to provide a higher degree of security.

**Note** When this command is written to NVRAM, the key is encrypted so that it is not displayed when the configuration is viewed.

## Example

The following example sets authentication key 10 to *aNiceKey*:

```
ntp authentication-key 10 md5 aNiceKey
```

## Related Commands

**ntp authenticate**
**ntp peer**
**ntp server**
**ntp trusted-key**

# ntp broadcast

To specify that a specific interface should send Network Time Protocol (NTP) broadcast packets, use the **ntp broadcast** interface configuration command. Use the **no** form of the command to disable this capability.

**ntp broadcast** [**version** *number*]
**no ntp broadcast**

## Syntax Description

| | |
|---|---|
| **version** *number* | (Optional) Number from 1 to 3 indicating the NTP version. |

## Default

Disabled

## Command Mode

Interface configuration

## Examples

In the following example, Ethernet interface 0 is configured to send NTP version 2 packets:

```
interface ethernet0
ntp broadcast version 2
```

## Related Commands

**ntp broadcast client**
**ntp broadcastdelay**

# ntp broadcast client

To allow the system to receive NTP broadcast packets on an interface, use the **ntp broadcast client** command. Use the **no** form of the command to disable this capability.

**ntp broadcast client**
**no ntp broadcast client**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

Use this command to allow the system to listen to broadcast packets on an interface-by-interface basis.

## Example

In the following example, the router synchronizes to NTP packets broadcasted on Ethernet interface 1:

```
interface ethernet1
ntp broadcast client
```

## Related Commands

**ntp broadcast**
**ntp broadcastdelay**

# ntp broadcastdelay

To set the estimated round-trip delay between the router and a Network Time Protocol (NTP) broadcast server, use the **ntp broadcastdelay** global configuration command. Use the **no** form of this command to revert to the default value.

**ntp broadcastdelay** *microseconds*
**no ntp broadcastdelay**

## Syntax Description

| | |
|---|---|
| *microseconds* | Estimated round-trip time (in microseconds) for NTP broadcasts. The range is from 1 to 999999. |

## Default

3000 microseconds

## Command Mode

Global configuration

## Usage Guidelines

Use this command when the router is configured as a broadcast client and the round-trip delay on the network is other than 3000 microseconds.

## Example

In the following example, the estimated round-trip delay between the router and the broadcast client is set to 5000 microseconds:

```
ntp broadcastdelay 5000
```

## Related Commands

**ntp broadcast**
**ntp broadcast client**

# ntp clock-period

Do not enter this command; it is documented for informational purposes only. The system automatically generates this command as Network Time Protocol (NTP) determines the clock error and compensates.

As NTP compensates for the error in the system clock, it keeps track of the correction factor for this error. The system automatically saves this value into the system configuration using the **ntp clock-period** global configuration command. The system uses the **no** form of this command to revert to the default.

**ntp clock-period** *value*
**no ntp clock-period**

## Syntax Description

| | |
|---|---|
| *value* | Amount to add to the system clock for each clock hardware tick (in units of 2-32 seconds). |

## Default

17179869 (4 milliseconds)

## Command Mode

Global configuration

## Usage Guidelines

If a **write memory** command is entered to save the configuration to NVRAM, this command will automatically be added to the configuration. It is a good idea to perform this task after NTP has been running for a week or so; this will help NTP synchronize more quickly if the system is restarted.

# ntp disable

To prevent an interface from receiving Network TIme Protocol (NTP) packets, use the **ntp disable** interface configuration command. To enable receipt of NTP packets on an interface, use the **no ntp disable** command.

**ntp disable**
**no ntp disable**

## Syntax Description

This command has no arguments or keywords.

## Default

Enabled

## Command Mode

Interface configuration

## Usage Guidelines

This command provides a simple method of access control.

## Example

In the following example, Ethernet interface 0 is prevented from receiving NTP packets:

```
interface ethernet0
ntp disable
```

# ntp master

To configure the router as a Network Time Protocol (NTP) master clock to which peers synchronize themselves when an external NTP source is not available, use the **ntp maste**r global configuration command. To disable the master clock function, use the **no ntp master** command.

**ntp master** [*stratum*]
**no ntp master** [*stratum*]

## Syntax Description

| | |
|---|---|
| *stratum* | (Optional) Number from 1 to 15. Indicates the NTP stratum number that the system will claim. |

## Default

By default, the master clock function is disabled. When enabled, the default stratum is 8.

## Command Mode

Global configuration

## Usage Guidelines

Since our implementation of NTP does not support directly attached radio or atomic clocks, the router is normally synchronized, directly or indirectly, to an external system that has such a clock. In a network without Internet connectivity, such a time source may not be available. The **ntp master** command is used in such cases.

If the system has **ntp master** configured, and it cannot reach any clock with a lower stratum number, the system will claim to be synchronized at the configured stratum number, and other systems will be willing to synchronize to it via NTP.

**Note** The system clock must have been set from some source, including manually, before **ntp master** will have any effect. This protects against distributing erroneous time after the system is restarted.

**Caution** Use this command with **extreme** 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.

## Example

In the following example, the router is configured as an NTP master clock to which peers may synchronize:

```
ntp master 10
```

## Related Command

**clock calendar-valid**

# ntp peer

To configure the router's system clock to synchronize a peer or to be synchronized by a peer, use the **ntp peer** global configuration command. To disable this capability, use the **no ntp peer** command.

**ntp peer** *ip-address* [**version** *number*] [**key** *keyid*] [**source** *interface*] [**prefer**]
**no ntp peer** *ip-address*

## Syntax Description

| | |
|---|---|
| *ip-address* | IP address of the peer providing, or being provided, the clock synchronization. |
| **version** | (Optional) Defines the Network Time Protocol (NTP) version number. |
| *number* | (Optional) NTP version number (1 to 3). |
| **key** | (Optional) Defines the authentication key. |
| *keyid* | (Optional) Authentication key to use when sending packets to this peer. |
| **source** | (Optional) Names the interface. |
| *interface* | (Optional) Name of the interface from which to pick the IP source address. |
| **prefer** | (Optional) Makes this peer the preferred peer that provides synchronization. |

## Default

No peers are configured by default. If a peer is configured, the default NTP version number is 3, no authentication key is used, and the source IP address is taken from the outgoing interface.

## Command Mode

Global configuration

## Usage Guidelines

Use this command if you want to allow this machine to synchronize with the peer, or vice versa. Using the **prefer** keyword will reduce switching back and forth between peers.

If you are using the default version of 3 and NTP synchronization does not occur, try using NTP version number 2. Many NTP servers on the Internet run version 2.

## Example

In the following example, the router is configured to allow its system clock to be synchronized with the clock of the peer (or vice versa) at IP address 131.108.22.33 using NTP version 2. The source IP address will be the address of Ethernet 0.

```
ntp peer 131.108.22.33 version 2 source Ethernet 0
```

## Related Commands

**ntp authentication-key**
**ntp server**
**ntp source**

# ntp server

To allow the router's system clock to be synchronized by a time server, use the **ntp server** global configuration command. To disable this capability, use the **no ntp server** command.

**ntp server** *ip-address* [**version** *number*] [**key** *keyid*] [**source** *interface*] [**prefer**]
**no ntp server** *ip-address*

## Syntax Description

| | |
|---|---|
| *ip-address* | IP address of the time server providing the clock synchronization. |
| **version** | (Optional) Defines the Network Time Protocol (NTP) version number. |
| *number* | (Optional) NTP version number (1 to 3). |
| **key** | (Optional) Defines the authentication key. |
| *keyid* | (Optional) Authentication key to use when sending packets to this peer. |
| **source** | (Optional) Identifies the interface from which to pick the IP source address. |
| *interface* | (Optional) Name of the interface from which to pick the IP source address. |
| **prefer** | (Optional) Makes this server the preferred server that provides synchronization. |

## Default

No peers are configured by default. If a peer is configured, the default NTP version number is 3, no authentication key is used, and the source IP address is taken from the outgoing interface.

## Command Mode

Global configuration

## Usage Guidelines

Use this command if you want to allow this machine to synchronize with the specified server. The server will not synchronize to this machine.

Using the **prefer** keyword will reduce switching back and forth between servers.

If you are using the default version of 3 and NTP synchronization does not occur, try using NTP version number 2. Many NTP servers on the Internet run version 2.

## Example

In the following example, the router is configured to allow its system clock to be synchronized with the clock of the peer at IP address 128.108.22.44 using NTP version 2:

```
ntp server 128.108.22.44 version 2
```

## Related Commands

**ntp authentication-key**
**ntp peer**
**ntp source**

# ntp source

To use a particular source address in Network Time Protocol (NTP) packets, use the **ntp source** global configuration command. Use the **no** form of this command to remove the specified source address.

**ntp source** *interface*
**no ntp source**

## Syntax Description

| | |
|---|---|
| *interface* | Any valid system interface name. |

## Default

Source address is determined by the outgoing interface.

## Command Mode

Global configuration

## Usage Guidelines

Use this command when you want to use a particular source IP address for all NTP packets. The address is taken from the named interface. This command is useful if the address on an interface cannot be used as the destination for reply packets. If the **source** keyword is present on an **ntp server** or **ntp peer** command, that value overrides the global value.

## Example

In the following example, the router is configured to use the IP address of Ethernet 0 as the source address of all outgoing NTP packets:

```
ntp source ethernet 0
```

## Related Commands

**ntp peer**
**ntp server**

# ntp trusted-key

If you want to authenticate the identity of a system to which Network Time Protocol (NTP) will synchronize, use the **ntp trusted-key** global configuration command. Use the **no** form of this command to disable authentication of the identity of the system.

**ntp trusted-key** *key-number*
**no ntp trusted-key** *key-number*

## Syntax Description

| | |
|---|---|
| *key-number* | Key number of authentication key to be trusted. |

## Default

Disabled

## Command Mode

Global configuration

## Usage Guidelines

If authentication is enabled, use this command to define one or more key numbers (corresponding to the keys defined with the**ntp authentication-key** command) that a peer NTP system must provide in its NTP packets, in order for this system to synchronize to it. This provides protection against accidentally synchronizing the system to a system that is not trusted, since the other system must know the correct authentication key.

## Example

In the following example, the system is configured to synchronize only to systems providing authentication key 42 in its NTP packets:

```
ntp authenticate
ntp authentication-key 42 md5 aNiceKey
ntp trusted-key 42
```

## Related Commands

**ntp authenticate**
**ntp authentication-key**

# ntp update-calendar

To periodically update the Cisco 7000 calendar from Network Time Protocol (NTP), use the **ntp update-calendar** global configuration command. Use the **no** form of this command to disable this feature.

**ntp update-calendar**
**no ntp update-calendar**

## Syntax Description

This command has no arguments or keywords.

## Default

The Cisco 7000 calendar is not updated.

## Command Mode

Global configuration

## Usage Guidelines

If a Cisco 7000 is synchronized to an outside time source via NTP, it is a good idea to periodically update the calendar with the time learned from NTP. Otherwise, the calendar will tend to gradually lose or gain time. The calendar will be updated only if NTP has synchronized to an authoritative time server.

## Example

In the following example, the system is configured to periodically update the calendar from the system clock:

```
ntp update-calendar
```

## Related Commands

**clock read-calendar**
**clock update-calendar**

# ping (privileged)

Use the **ping** (packet internet groper) privileged EXEC command to diagnose basic network connectivity on Apollo, AppleTalk, CLNS, DECnet, IP, Novell IPX, VINES, or XNS networks.

**ping** [*protocol*] {*host* | *address*}

## Syntax Description

| | |
|---|---|
| *protocol* | (Optional) Protocol keyword, one of **apollo**, **appletalk**, **clns**, **decnet**, **ip**, **ipx**, **vines**, or **xns**. |
| *host* | Host name of system to ping. |
| *address* | Address of system to ping. |

## Command Mode

Privileged EXEC

## Usage Guidelines

The ping program sends an echo request packet to an address, then awaits a reply. Ping output can help you evaluate path-to-host reliability, delays over the path, and whether the host can be reached or is functioning.

To abort a ping session, type the escape sequence (by default, Ctrl-^ X, which is done by simultaneously pressing the Ctrl, Shift, and 6 keys, letting go, then pressing the X key).

Table 5-9 describes the test characters that the ping facility sends.

**Table 5-9 Ping Test Characters**

| Char | Meaning |
|---|---|
| ! | Each exclamation point indicates receipt of a reply. |
| . | Each period indicates the network server timed out while waiting for a reply. |
| U | A destination unreachable error PDU was received. |
| C | A congestion experienced packet was received. |
| I | User interrupted test. |
| ? | Unknown packet type. |
| & | Packet lifetime exceeded. |

**Note** Not all protocols require hosts to support pings, and for some protocols, the pings are Cisco-defined and are only answered by another Cisco router.

## Example

After you enter the **ping** command in privileged mode, the system prompts for one of the following keywords**: appletalk**, **clns**, **ip**, **novell**, **apollo**, **vines**, **decnet**, or **xns.** The default protocol is IP.

If you enter a host name or address on the same line as the **ping** command, the default action is taken as appropriate for the protocol type of that name or address.

While the precise dialog varies somewhat from protocol to protocol, all are similar to the ping session using default values shown in the following display.

```
Router# ping
Protocol [ip]:
Target IP address: 192.31.7.27
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.31.7.27, timeout is 2 seconds:
!!!!!
Success rate is 100 percent, round-trip min/avg/max = 1/2/4 ms
```

Table 5-10 describes the default **ping** fields shown in the display.

**Table 5-10 Ping Field Descriptions**

| Field | Description |
|---|---|
| Protocol [ip]: | Prompts for a supported protocol. Enter **appletalk**, **clns**, **ip**, **novell**, **apollo**, **vines**, **decnet**, or **xns**. Default: **ip**. |
| Target IP address: | Prompts for the IP address or host name of the destination node you plan to ping. If you have specified a supported protocol other than IP, enter an appropriate address for that protocol here. Default: none. |
| Repeat count [5]: | Number of ping packets that will be sent to the destination address. Default: 5. |
| Datagram size [100]: | Size of the ping packet (in bytes). Default: 100 bytes. |
| Timeout in seconds [2]: | Timeout interval. Default: 2 (seconds). |
| Extended commands [n]: | Specifies whether or not a series of additional commands appears. Many of the following displays and tables show and describe these commands. |
| Sweep range of sizes [n]: | Allows you to vary the sizes of the echo packets being sent. This capability is useful for determining the minimum sizes of the MTUs configured on the nodes along the path to the destination address. Packet fragmentation contributing to performance problems can then be reduced. |
| !!!!! | Each exclamation point (!) indicates receipt of a reply. A period (.) indicates the network server timed out while waiting for a reply. Other characters may appear in the ping output display, depending on the protocol type. |
| Success rate is 100 percent | Percentage of packets successfully echoed back to the router. Anything less than 80 percent is usually considered problematic. |
| round-trip min/avg/max = 1/2/4 ms | Round-trip travel time intervals for the protocol echo packets, including minimum/average/maximum (in milliseconds). |

## Related Command

**ping (user)**

# ping (user)

Use the **ping** (packet internet groper) user EXEC command to diagnose basic network connectivity on AppleTalk, CLNS, IP, Novell, Apollo, VINES, DECnet, or XNS networks.

**ping** [*protocol*] {*host* | *address*}

## Syntax Description

| | |
|---|---|
| *protocol* | (Optional) Protocol keyword, one of **apollo**, **appletalk**, **clns**, **decnet**, **ip**, **ipx**, **vines**, or **xns**. |
| *host* | Host name of system to ping. |
| *address* | Address of system to ping. |

## Command Mode

EXEC

## Usage Guidelines

The user-level ping feature provides a basic ping facility for users who do not have system privileges. This feature allows the router to perform the simple default ping functionality for a number of protocols. Only the nonverbose form of the **ping** command is supported for user-level pings.

If the system cannot map an address for a host name, it will return an "%Unrecognized host or address" error message.

To abort a ping session, type the escape sequence (by default, Ctrl-^ X, which is done by simultaneously pressing the Ctrl, Shift, and 6 keys, letting go, then pressing the X key).

Table 5-11 describes the test characters that the ping facility sends.

**Table 5-11 Ping Test Characters**

| Char | Meaning |
|---|---|
| ! | Each exclamation point indicates receipt of a reply. |
| . | Each period indicates the network server timed out while waiting for a reply. |
| U | A destination unreachable error PDU was received. |
| C | A congestion experienced packet was received. |
| I | User interrupted test. |
| ? | Unknown packet type. |
| & | Packet lifetime exceeded. |

## Example

The following display shows sample ping output when you ping the IP host named *donald*:

```
Router> ping donald
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 192.31.7.27, timeout is 2 seconds:
!!!!!
Success rate is 100 percent, round-trip min/avg/max = 1/3/4 ms
```

## Related Command

**ping (privileged)**

# ppp authentication

To enable Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) and to enable an AAA authentication method on an interface, use the **ppp authentication** interface configuration command. Use the **no** form of the command to disable this authentication.

**ppp authentication** {**chap** | **pap**} [**if-needed**] [*list-name*]
**no ppp authentication**

**Caution** If you use a *list-name* value that was not configured with the **aaa authentication ppp** command, you will disable PPP on this interface.

## Syntax Description

| | |
|---|---|
| **chap** | Enables CHAP on a serial interface. |
| **pap** | Enables PAP on a serial interface. |
| **if-needed** | (Optional) Used with TACACS and extended TACACS. Does not perform CHAP or PAP authentication if the user has already provided authentication. This option is available only on asynchronous interfaces. |
| *list-name* | (Optional) Used with AAA/TACACS+. Specifies the name of a list of AAA methods of authentication to use. If no listname is specified, the system uses the default. Lists and default are created with the **aaa authentication ppp** command. |

## Default

PPP authentication is not enabled.

## Command Mode

Interface configuration

## Usage Guidelines

Once you have enabled CHAP or PAP, the local router requires a password from remote devices. If the remote device does not support CHAP or PAP, no traffic is passed to that device.

If you are using **autoselect** on a TTY line, you will probably want to use the **ppp authentication** command to turn on PPP authentication for the corresponding interface.

If you specify the **if-needed** option, PPP authentication is not required when the user has already provided authentication. This option is useful if you are using the **autoselect** command, but it cannot be used with AAA/TACACS+.

The *list-name* argument can only be used when AAA/TACACS+ is initialized and cannot be used with the **if-needed** argument.

## Example

The following example enables CHAP on asynchronous interface 4, and uses the authentication list MIS-access:

```
interface async 4
encapsulation ppp
ppp authentication chap MIS-access
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**aaa authentication ppp**
**aaa new-model**
**autoselect** †
**encapsulation ppp**
**ppp use-tacacs**
**username**

# ppp use-tacacs

To enable TACACS for PPP authentication, use the **ppp use-tacacs** interface configuration command. Use the **no** form of the command to disable TACACS for PPP authentication.

**ppp use-tacacs** [**single-line**]
**no ppp use-tacacs**

## Syntax Description

| | |
|---|---|
| **single-line** | (Optional) Accept the username and password in the username field. This option applies only when using CHAP authentication. |

## Default

TACACS is not used for PPP authentication.

## Command Mode

Interface configuration

## Usage Guidelines

This is a per-interface command. Use this command only when you have set up an extended TACACS server. This command requires the new extended TACACS server.

When CHAP authentication is being used, the **ppp use-tacacs** command with the **single-line** option specifies that if a username and password are specified in the username, separated by an asterix (*), then a standard tacacs login query is performed using that username and password. If the username does not contain an asterix, then normal CHAP authentication is performed using TACACS.

This feature is useful when integrating TACACS with other authentication systems that require a clear-text version of the user’s password. Such systems include one-time password systems, token card systems, kerberos, and others.

**Caution** Normal CHAP authentications prevent the clear-text password from being transmitted over the link. When you use the single-line option, passwords will cross the link in the clear.

If the username and password are contained in the CHAP password, then the CHAP secret is not used by the Cisco system. Because most PPP clients will require that a secret be specified, you can use any arbitrary string; the Cisco system will ignore it.

**Note** This command is not used in AAA/TACACS+ and has been replaced with the **aaa authentication ppp** command.

## Examples

In the following example, asynchronous serial interface 1 is configured to use TACACS for CHAP authentication.

```
interface async 1
ppp authentication chap
ppp use-tacacs
```

In the following example, asynchronous serial interface 1 is configured to use TACACS for PAP authentication.

```
interface async 1
ppp authentication pap
ppp use-tacacs
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**ppp authentication chap**†
**ppp authentication pap**†
**tacacs-server extended**
**tacacs-server host**

# priority-group

To assign the specified priority list to an interface, use the **priority-group** interface configuration command. Use the **no priority-group** command to remove the specified **priority-group** assignment.

**priority-group** *list*
**no priority-group**

## Syntax Description

| | |
|---|---|
| *list* | Priority list number assigned to the interface. |

## Default

None

## Command Mode

Interface configuration

## Usage Guidelines

Only one list can be assigned per interface. Priority output queueing provides a mechanism to prioritize packets transmitted on an interface.

## Example

The following example causes packets on interface serial 0 to be classified by priority list 1:

```
interface serial 0
priority-group 1
```

## Related Commands

**priority-list**
**priority-list interface**
**priority-list queue-limit**
**priority-list stun**

# priority-list default

To assign a priority queue for those packets that do not match any other rule in the priority list, use the **priority-list default** global configuration command. Use the **no priority-list default** command to return to the default or assign **normal** as the default.

**priority-list** *list-number* **default** {**high | medium | normal | low**}
**no priority-list** *list-number* **default** {**high | medium | normal | low**}

## Syntax Description

| | |
|---|---|
| *list-number* | Arbitrary integer between 1 and 10 that identifies the priority list selected by the user. |
| **high** \| **medium** \| **normal** \| **low** | Priority queue level. |

## Default

The **normal** queue is assumed if you use the **no** form of the command.

## Command Mode

Global configuration

## Example

The following example sets the priority queue for those packets that do not match any other rule in the priority list to a low priority:

```
priority-list 1 default low
```

## Related Commands

**priority-group**
**show queueing**

# priority-list interface

To establish queuing priorities on packets entering from a given interface, use the **priority-list interface** global configuration command. Use the **no priority-list** command with the appropriate arguments to remove an entry from the list.

**priority-list** *list-number* **interface** *interface-type interface-number* {**high | medium | normal | low}**
**no priority-list** *list-number* **interface** *interface-type interface-number* {**high | medium | normal | low**}

## Syntax Description

| | |
|---|---|
| *list-number* | Arbitrary integer between 1 and 10 that identifies the priority list selected by the user. |
| *interface-type* | Specifies the name of the interface. |
| *interface-number* | Number of the specified interface. |
| **high** \| **medium** \| **normal** \| **low** | Priority queue level. |

## Default

No queuing priorities are established.

## Command Mode

Global configuration

## Example

The following example sets any packet type entering on Ethernet interface 0 to a medium priority:

```
priority-list 3 interface ethernet 0 medium
```

## Related Commands

**priority-group**
**show queueing**

# priority-list protocol

To establish queuing priorities based upon the protocol type, use the **priority-list protocol** global configuration command. Use the **no priority-list protocol** command with the appropriate list number to remove an entry from the list.

**priority-list** *list -number* **protocol** *protocol-name* {**high** | **medium** | **normal** | **low**}
*queue-keyword keyword-value*
**no priority-list** *list -number* **protocol**

## Syntax Description

| | |
|---|---|
| *list-number* | Arbitrary integer between 1 and 10 that identifies the priority list selected by the user. |
| *protocol-name* | Specifies the protocol type: **aarp**, **arp**, **apollo**, **appletalk**, **bridge** (transparent), **clns**, **clns_es**, **clns_is**, **compressedtcp**, **cmns**, **decnet**, **decnet_node**, **decnet_router-l1**, **decnet_router-l2, ip**, **ipx**, **pad**, **rsrb**, **stun**, **vines**, **xns**, and **x25**. |
| **high** \| **medium** \| **normal** \| **low** | Priority queue level. |
| *queue-keyword keyword-value* | Possible keywords are **fragments**, **gt**, **lt**, **list**, **tcp**, and **udp**. See Table 5-12. |

## Default

No queuing priorities are established.

## Command Mode

Global configuration

## Usage Guidelines

When using multiple rules for a single protocol, remember that the system reads the priority settings in order of appearance. When classifying a packet, the system searches the list of rules specified by **priority-list** commands for a matching protocol type. When a match is found, the packet is assigned to the appropriate queue. The list is searched in the order it is specified, and the first matching rule terminates the search.

The **decnet_router-l1** keyword refers to the multicast address for all level-1 routers, which are intra-area routers, and the **decnet_router-l2** keyword refers to all level 2 routers, which are interarea routers.

Use Table 5-12, Table 5-13, and Table 5-14 to configure the queuing priorities for your system.

**Table 5-12 Protocol Priority Queue Keywords and Values**

| Option | Description |
|---|---|
| **fragments** | Assigns the priority level defined to fragmented IP packets (for use with IP protocol only). More specifically, IP packets whose fragment offset field is nonzero are matched by this command. The initial fragment of a fragmented IP packet has a fragment offset of zero, so such packets are not matched by this command.<br>Note: Packets with a nonzero fragment offset do not contain TCP or UDP headers, so other instances of this command that use the **tcp** or **udp** keyword will always fail to match such packets. |
| **gt** *byte-count* | Specifies a greater-than count. The priority level assigned goes into effect when a packet exceeds the value entered for the argument *byte-count*. The size of the packet must also include additional bytes due to MAC encapsulation on the outgoing interface. |
| **lt** *byte-count* | Specifies a less-than count. The priority level assigned goes into effect when a packet size is less than the value entered for *byte-count*. The size of the packet must also include additional bytes due to MAC encapsulation on the outgoing interface. |
| **list** *list-number* | Assigns traffic priorities according to a specified list when used with Appletalk, bridging, IP, IPX, VINES, or XNS. The *list-number* argument is the access list number as specified by the **access-list** global configuration command for the specified *protocol-name*. For example, if the protocol is AppleTalk, *list-number* should be a valid AppleTalk access list number. |
| **tcp** *port* | Assigns the priority level defined to TCP segments originating from or destined to a specified port (for use with the IP protocol only). Table 5-13 lists common TCP services and their port numbers. |
| **udp** *port* | Assigns the priority level defined to UDP packets originating from or destined to the specified port (for use with the IP protocol only). Table 5-14 lists common UDP services and their port numbers. |

**Table 5-13 Common TCP Services and Their Port Numbers**

| Service | Port |
|---|---|
| Telnet | 23 |
| SMTP | 25 |

**Table 5-14 Common UDP Services and Their Port Numbers**

| Service | Port |
|---|---|
| TFTP | 69 |
| NFS | 2049 |
| SNMP | 161 |
| RPC | 111 |
| DNS | 53 |

**Note** The TCP and UDP ports listed in Table 5-13 and Table 5-14 include some of the more common port numbers. However, you can specify any port number to be prioritized; you are not limited to those listed.

Use the **no priority-list** global configuration command followed by the appropriate *list-number* argument and the **protocol** keyword to remove a priority list entry assigned by protocol type.

## Examples

The following example assigns 1 as the arbitrary priority list number, specifies DECnet as the protocol type, and assigns a high-priority level to the DECnet packets transmitted on this interface:

```
priority-list 1 protocol decnet high
```

The following example assigns a medium-priority level to every DECnet packet with a size greater than 200 bytes:

```
priority-list 2 protocol decnet medium gt 200
```

The following example assigns a medium-priority level to every DECnet packet with a size less than 200 bytes:

```
priority-list 4 protocol decnet medium lt 200
```

The following example assigns a high-priority level to traffic that matches IP access list 10:

```
priority-list 1 protocol ip high list 10
```

The following example assigns a medium-priority level to Telnet packets:

```
priority-list 4 protocol ip medium tcp 23
```

The following example assigns a medium-priority level to UDP Domain Name service packets:

```
priority-list 4 protocol ip medium udp 53
```

The following example assigns a high-priority level to traffic that matches Ethernet type code access list 201:

```
priority-list 1 protocol bridge high list 201
```

## Related Commands

**priority-group**
**show queueing**

# priority-list queue-limit

To specify the maximum number of packets that can be waiting in each of the priority queues, use the **priority-list queue-limit** global configuration command.The **no priority-list queue-limit** command selects the normal queue.

**priority-list** *list-number* **queue-limit** *high-limit medium-limit normal-limit low-limit*
**no priority-list** *list-number* **queue-limit**

## Syntax Description

| | |
|---|---|
| *list-number* | Arbitrary integer between 1 and 10 that identifies the priority list selected by the user. |
| *high-limit medium-limit normal-limit low-limit* | Priority queue maximum length. A value of 0 for any of the four arguments means that the queue can be of unlimited size for that particular queue. |

## Default

The default queue limit arguments are listed in Table 5-15.

**Table 5-15 Priority Queue Packet Limits**

| Priority Queue Argument | Packet Limits |
|---|---|
| *high-limit* | 20 |
| *medium-limit* | 40 |
| *normal-limit* | 60 |
| *low-limit* | 80 |

## Command Mode

Global configuration

## Usage Guidelines

If a priority queue overflows, excess packets are discarded and quench messages can be sent, if appropriate, for the protocol.

## Example

The following example sets the maximum packets in the priority queue to 10:

```
priority-list 2 queue-limit 10 40 60 80
```

## Related Commands

**priority-group**
**show queueing**

# priority-list stun

To establish queuing priorities based on the address of the serial link on a STUN connection, use the **priority-list stun** global configuration command. Use the **no priority-list stun** command with the appropriate arguments to remove an entry from the list.

**priority-list** *list-number* **stun** {**high** | **medium** | **normal** | **low**} **address** *group-number address*
**no priority-list** *list-number* **stun** {**high** | **medium** | **normal** | **low**} **address** *group-number address*

## Syntax Description

| | |
|---|---|
| *list-number* | Arbitrary integer between 1 and 10 that identifies the priority list selected by the user. |
| **high** \| **medium** \| **normal** \| **low** | Priority queue level. |
| **address** | Required keyword. |
| *group-number* | Group number used in the **stun group** command. |
| *address-number* | Address of the serial link. The format of the address is either a 1-byte hex value (for example, C1) for an SDLC link or one that is specified by the **stun schema** global configuration command. |

## Default

No queuing priorities are established.

## Command Mode

Global configuration

## Example

The following example illustrates how to prioritize STUN traffic over IP. STUN uses a special serial line protocol called STUN for the simple serial encapsulation and TCP port 1994 for the TCP encapsulation. The example assigns the same priority to STUN traffic over a serial link.

```
priority-list 4 ip high tcp 1994
priority-list 4 stun high address 3 C1
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**priority-group**
**show queueing**
**stun schema offset length format** †

# privilege level (global)

To set the privilege level for a command, use the **privilege level** global configuration command. Use the **no** form of this command to revert to default privileges for a given command.

**privilege** *mode* **level** *level command*
**no privilege** *mode* **level** *level command*

## Syntax Description

| | |
|---|---|
| *mode* | Configuration mode. See Table 5-6 in the description of the **alias** command for a list of acceptable options. |
| *level* | Privilege level to be associated with the specified command. You can specify up to sixteen privilege levels, using numbers 0 through 15. |
| *command* | Command to which privilege level is associated. |

## Default

Level 15 is the level of access permitted by the **enable** password.

Level 1 is normal EXEC-mode user privileges.

## Command Mode

Global configuration

## Usage Guidelines

Table 5-6 in the description of the **alias** command shows the acceptable options for the *mode* argument in the **privilege level** global configuration command.

The password for the privilege level defined using the **privilege level** global configuration mode is configured using the **enable password**command.

Level 0 can be used to specify a more-limited subset of commands for specific users or lines. For example, you can allow user “guest” to only use the **show users** and **exit** commands.

If you set a command to a privilege level, all commands that have a syntax that is a subset of the syntax of that command will also be set to that level. For example, if you set the command **show ip route** to level 15, if you do not set **show** commands and **show ip** commands to a different level, they will also be at privilege level 15.

## Example

In the following example, the **configure** command in global configuration mode is assigned a privilege level of 14. Only users who know the level 14 password will be able to use the **configure** command.

```
privilege exec level 14 configure
enable password level 14 pswd14
```

## Related Commands

**enable password**
**privilege level (line)**

# privilege level (line)

To set the default privilege level for a line, use the **privilege level** line configuration command. Use the **no** form of this command to restore the default user privilege level to the line.

**privilege level** *level*
**no privilege level**

## Syntax Description

| | |
|---|---|
| *level* | Privilege level to be associated with the specified line. |

## Default

Level 15 is the level of access permitted by the enable password.

Level 1 is normal EXEC-mode user privileges.

## Command Mode

Line configuration

## Usage Guidelines

The privilege level that is set using this command can be overridden by a user logging in to the line and enabling a different privilege level. The user can lower the privilege level by using the **disable** command. If they know the password to a higher privilege level, they can use that password to to enable the higher privilege level.

Level 0 can be used to specify a more limited subset of commands for specific users or lines. For example, you can allow user “guest” to only use the **show users** and **exit** commands.

You might specify a high level of privilege for your console line if you are able to restrict who uses that line.

## Example

In the following example, the auxiliary line is configured for privilege level 5. Anyone who is using the auxiliary line will have privilege level 5 by default.

```
line aux 0
privilege level 5
```

## Related Commands

**enable password**
**privilege level (line)**

# prompt

To customize the router prompt, use the **prompt** global configuration command. To revert to the default router prompt, use the **no** form of this command.

**prompt** *string*
**no prompt** [*string*]

## Syntax Description

| | |
|---|---|
| *string* | Router prompt. It can consist of all printing characters and the escape sequences listed in Table 5-16 in the "Usage Guidelines" section. |

## Default

The default router prompt is either *Router* or the router name defined with the **hostname** global configuration command, followed by an angle bracket (>) for EXEC mode or a pound sign (#) for privileged EXEC mode.

## Command Mode

Global configuration

## Usage Guidelines

You can include escape sequences when specifying the router prompt. All escape sequences are preceded by a percent sign (%). Table 5-16 lists the valid escape sequences.

**Table 5-16 Custom Router Prompt Escape Sequences**

| Escape Sequence | Interpretation |
|---|---|
| **%h** | Router's host name. This is either *Router* or the name defined with the **hostname** global configuration command. |
| **%n** | Physical terminal line (TTY) number of the EXEC user. |
| **%p** | Prompt character itself. It is either an angle bracket (>) for EXEC mode or a pound sign (#) for privileged EXEC mode. |
| **%s** | Space. |
| **%t** | Tab. |
| **%%** | Percent sign (%) |

Specifying the command **prompt %h** has the same effect as issuing the **no prompt** command.

## Examples

The following example changes the EXEC prompt to include the TTY number, followed by the router name and a space:

```
prompt TTY%n@%h%s%p
```

The following are examples of user and privileged EXEC prompts that result from the previous command:

```
TTY17@Router1 >
TTY17SRouter1 #
```

## Related Command

**hostname**

# queue-list default

To assign a priority queue for those packets that do not match any other rule in the queue list, use the **queue-list default** global configuration command. To restore the default value, use the **no queue-list default** command.

**queue-list** *list-number* **default** *queue-number*
**no queue-list** *list-number* **default** *queue-number*

## Syntax Description

| | |
|---|---|
| *list-number* | Number of the queue list. An integer from 1 to 10. |
| *queue-number* | Number of the queue. An integer from 1 to 10. |

## Default

Queue number 1

## Command Mode

Global configuration

## Usage Guidelines

Queue number 0 is a system queue. It is emptied before any of the other queues are processed. The system enqueues high-priority packets, such as keepalives, to this queue.

## Example

In the following example, the default queue for list 10 is set to queue number 2:

```
queue-list 10 default 2
```

## Related Commands

**custom-queue-list**
**show queueing**

# queue-list interface

To establish queuing priorities on packets entering on an interface, use the **queue-list interface** global configuration command. To remove an entry from the list, use the **no** form of the command.

**queue-list** *list-number* **interface** *interface-type interface-number queue-number*
**no queue-list** *list-number* **interface** *queue-number*

## Syntax Description

| | |
|---|---|
| *list-number* | Number of the queue list. An integer from 1 to 10. |
| *interface-type* | Required argument that specifies the name of the interface. |
| *interface-number* | Number of the specified interface. |
| *queue-number* | Number of the queue. An integer from 1 to 10. |

## Default

No queuing priorities are established.

## Command Mode

Global configuration

## Example

In the following example, queue list 4 established queuing priorities for packets entering on interface tunnel 3. The queue number assigned is 10.

```
queue-list 4 interface tunnel 3 10
```

## Related Commands

**custom-queue-list**
**show queueing**

# queue-list protocol

To establish queuing priority based upon the protocol type, use the **queue-list protocol** global configuration command. Use the **no queue-list protocol** command with the appropriate list number to remove an entry from the list.

**queue-list** *list-number* **protocol** *protocol-name queue-number queue-keyword keyword-value*
**no queue-list** *list-number* **protocol** *protocol-name*

## Syntax Description

| | |
|---|---|
| *list-number* | Number of the queue list. An integer from 1 to 10. |
| *protocol-name* | Required argument that specifies the protocol type: **aarp**, **arp**, **apollo**, **appletalk**, **bridge** (transparent), **clns**, **clns_es**, **clns_is**, **compressedtcp**, **cmns**, **decnet**, **decnet_node**, **decnet_routerl1**, **decnet_routerl2**, **ip**, **ipx**, **pad**, **rsrb**, **stun**, **vines**, **xns**, and **x25**. |
| *queue-number* | Number of the queue. An integer from 1 to 10. |
| *queue-keyword keyword-value* | Possible keywords are **gt**, **lt**, **list**, **tcp**, and **udp**. See Table 5-12. |

## Default

No queuing priorities are established.

## Command Mode

Global configuration

## Usage Guidelines

When classifying a packet, the system searches the list of rules specified by **queue-list** commands for a matching protocol type. When a match is found, the packet is assigned to the appropriate queue. The list is searched in the order it is specified, and the first matching rule terminates the search.

The **decnet_router-l1** keyword refers to the multicast address for all level-1 routers, which are intra-area routers, and the **decnet_router-l2** keyword refers to all level 2 routers, which are interarea routers.

Use Table 5-12, Table 5-13, and Table 5-14 from the **priority-list protocol** command to configure custom queuing for your system.

## Examples

The following example assigns 1 as the custom queue list, specifies DECnet as the protocol type, and assigns 3 as a queue number to the packets transmitted on this interface:

```
queue-list 1 protocol decnet 3
```

The following example assigns DECnet packets with a size greater than 200 bytes to queue number 2:

```
queue-list 2 protocol decnet 2 gt 200
```

The following example assigns DECnet packets with a size less than 200 bytes to queue number 2:

```
queue-list 4 protocol decnet 2 lt 200
```

The following example assigns traffic that matches IP access list 10 to queue number 1:

```
queue-list 1 protocol ip 1 list 10
```

The following example assigns Telnet packets to queue number 2:

```
queue-list 4 protocol ip 2 tcp 23
```

The following example assigns UDP Domain Name service packets to queue number 2:

```
queue-list 4 protocol ip 2 udp 53
```

The following example assigns traffic that matches Ethernet type code access list 201 to queue number 1:

```
queue-list 1 protocol bridge 1 list 201
```

## Related Commands

**custom-queue-list**
**show queueing**

# queue-list queue byte-count

To designate the byte size allowed per queue, use the **queue-list queue byte-count** global configuration command. To return the byte size to the default value, use the **no** form of the command.

**queue-list** *list-number* **queue** *queue-number* **byte-count** *byte-count-number*
**no queue-list** *list-number* **queue** *queue-number* **byte-count** *byte-count-number*

## Syntax Description

| | |
|---|---|
| *list-number* | Number of the queue list. An integer from 1 to 10. |
| *queue-number* | Number of the queue. An integer from 1 to 10. |
| *byte-count-number* | Specifies the lower boundary on how many bytes the system allows to be delivered from a given queue during a particular cycle. |

## Default

1500 bytes

## Command Mode

Global configuration

## Example

In the following example, queue list 9 establishes the byte-count as 1400 for queue number 10:

```
queue-list 9 queue 10 byte-count 1400
```

## Related Commands

**custom-queue-list**
**show queueing**

# queue-list queue limit

To designate the queue length limit for a queue, use the **queue-list queue limit** global configuration command. To return the queue length to the default value, use the **no** form of the command.

**queue-list** *list-number* **queue** *queue-number* **limit** *limit-number*
**no queue-list** *list-number* **queue** *queue-number* **limit** *limit-number*

## Syntax Description

| | |
|---|---|
| *list-number* | Number of the queue list. An integer from 1 to 10. |
| *queue-number* | Number of the queue. An integer from 1 to 10. |
| *limit-number* | Maximum number of packets which can be enqueued at any time. Range is 0 to 32767 queue entries. |

## Default

20 entries

## Command Mode

Global configuration

## Example

In the following example, the queue length of queue 10 is increased to 40:

```
queue-list 5 queue 10 limit 40
```

## Related Commands

**custom-queue-list**
**show queueing**

# queue-list stun

To establish queuing priorities based on the address of the serial link on a STUN connection, use the **queue-list stun** global configuration command. Use the **no queue-list stun** command with the appropriate arguments to remove an entry from the list.

**queue-list** *list-number* **stun** *queue-number* **address** *group-number address-number*
**no queue-list** *list-number* **stun** *queue-number* **address** *group-number address-number*

## Syntax Description

| | |
|---|---|
| *list-number* | Number of the queue list. An integer from 1 to 10. |
| queue-number | Queue number in the range from 1 to 10. |
| **address** | Required keyword. |
| *group-number* | Group number used in the **stun group** command. |
| *address-number* | Address of the serial link. The format of the address is either a 1-byte hex value (for example, C1) for an SDLC link or one that is specified by the **stun schema** configuration command. |

## Default

None

## Command Mode

Global configuration

## Example

The following example causes the system to place STUN traffic matching the STUN group number 2 and address C1 onto queue number 3:

```
queue-list 3 stun 3 address 2 c1
```

## Related Commands

**custom-queue-list**
**show queueing**
**stun schema offset length format** †

# scheduler-interval

To control the maximum amount of time that can elapse without running the lowest-priority system processes, use the **scheduler-interval** global configuration command. The **no scheduler-interval** command restores the default.

**scheduler-interval** *milliseconds*
**no scheduler-interval**

## Syntax Description

| | |
|---|---|
| *milliseconds* | Integer that specifies the interval, in milliseconds. The minimum interval that you can specify is 500 milliseconds; there is no maximum value. |

## Default

High-priority operations are allowed to use as much of the central processor as needed.

## Command Mode

Global configuration

## Usage Guidelines

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, give priority to the system process scheduler.

## Example

The following example changes the low-priority process schedule to an interval of 750 milliseconds:

```
scheduler-interval 750
```

# service exec-wait

To delay the startup of the EXEC on noisy lines, use the **service exec-wait** global configuration command. Use the **no service exec-wait** command to disable this feature.

**service exec-wait**
**no service exec-wait**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Global configuration

## Usage Guidelines

This command delays startup of the EXEC until the line has been idle (no traffic seen) for 3 seconds. The default is to enable the line immediately on modem activation.

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 may be interpreted as usernames and passwords, causing authentication failure before the user gets a chance to type a username/password. The command is not useful on non-modem lines or lines without some kind of login configured.

## Example

The following example delays the startup of the EXEC:

```
service exec-wait
```

# service finger

To allow Finger protocol requests (defined in RFC 742) to be made of the network server, use the **service finger** global configuration command. This service is equivalent to issuing a remote **show users** command. The **no service finger** command removes this service.

**service finger**
**no service finger**

## Syntax Description

This command has no arguments or keywords.

## Default

Enabled

## Command Mode

Global configuration

## Example

The following is an example of how to disable the Finger protocol:

```
no service finger
```

# service nagle

To enable the Nagle congestion control algorithm, use the **service nagle** global configuration command. Use the **no service nagle** command to disable this feature.

**service nagle**
**no service nagle**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Global configuration

## Usage Guidelines

When using a standard TCP implementation to send keystrokes between machines, TCP tends to send one packet for each keystroke typed. On larger networks, many small packets use up bandwidth and contribute to congestion.

John Nagle's algorithm (RFC 896) helps alleviate the small-packet problem in TCP. In general, it works this way: 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 them out to the network at a rate matching the round-trip time of the given connection. This method is usually a good for all TCP-based traffic. However, do not use the **service nagle** command if you have XRemote users on X Window sessions.

## Example

The following example enables the Nagle algorithm on the router:

```
service nagle
```

# service password-encryption

To encrypt passwords, use the **service password-encryption** global configuration command. Use the **no service password-encryption** command to disable this service.

**service password-encryption**
**no service password-encryption**

## Syntax Description

This command has no arguments or keywords.

## Default

No encryption

## Command Mode

Global configuration

## Usage Guidelines

The actual encryption process occurs when the current configuration is written or when a password is configured. Password encryption can be applied to both the privileged command password and to console and virtual terminal line access passwords.

When password encryption is enabled, the encrypted form of the passwords is displayed when a **show configuration** command is entered.

**Note** It is not possible to recover a lost encrypted password.

## Example

The following example causes password encryption to take place:

```
service password-encryption
```

# service tcp-keepalives

To generate keepalive packets on idle network connections, use the **service tcp-keepalives** global configuration command. The **no service tcp-keepalives** command with the appropriate keyword disables the keepalives.

**service tcp-keepalives** {**in** | **out**}
**no service tcp-keepalives** {**in** | **out**}

## Syntax Description

| | |
|---|---|
| **in** | Generates keepalives on incoming connections (initiated by remote host). |
| **out** | Generates keepalives on outgoing connections (initiated by a user). |

## Default

Disabled

## Command Mode

Global configuration

## Example

The following example generates keepalives on incoming TCP connections:

```
service tcp-keepalives in
```

# service telnet-zero-idle

To set the TCP window to zero (0) when the Telnet connection is idle, use the **service telnet-zero-idle** global configuration command. Use the **no service telnet-zero-idle** command to disable this feature.

**service telnet-zero-idle**
**no service telnet-zero-idle**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Global configuration

## Usage Guidelines

Normally, data sent to non-current 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 or the EXEC is sitting in command mode), 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.

## Example

The following example sets the TCP window to zero when the Telnet connection is idle:

```
service telnet-zero-idle
```

## Related Command

**resume**

# service timestamps

To configure the system to timestamp debugging or logging messages, use one of the **service timestamps** global configuration commands. Use the **no service timestamps** command to disable this service.

**service timestamps** [*type* **uptime**]
**service timestamps** *type* **datetime** [**msec**] [**localtime**] [**show-timezone**]
**no service timestamps** [*type*]

## Syntax Description

| | |
|---|---|
| *type* | Type of message to timestamp: **debug** or **log**. |
| **uptime** | (Optional) Timestamp with time since the system was rebooted. |
| **datetime** | Timestamp with the date and time. |
| **msec** | (Optional) Include milliseconds in the date and timestamp. |
| **localtime** | (Optional) Timestamp relative to the local time zone. |
| **show-timezone** | (Optional) Include the time zone name in the timestamp. |

## Default

No timestamping.

If **service timestamps** is specified with no arguments or keywords, default is **service timestamps debug uptime**.

The default for **service timestamps** *type* **datetime** is to format the time in UTC, with no milliseconds and no time zone name.

The command **no service timestamps** by itself disables timestamps for both debug and log messages.

## Command Mode

Global configuration

## Usage Guidelines

Timestamps can be added to either debugging or logging messages independently. The **uptime** form of the command adds timestamps in the format HHHH:MM:SS, indicating the time since the system was rebooted. The **datetime** form of the command adds timestamps in the format MMM DD HH:MM:SS, indicating the date and time according to the system clock. If the system clock has not been set, the date and time are preceded by an asterisk (*) to indicate that the date and time are probably not correct.

## Examples

The following example enables timestamps on debugging messages, showing the time since reboot:

```
service timestamps debug uptime
```

The following example enables timestamps on logging messages, showing the current time and date relative to the local time zone, with the time zone name included:

```
service timestamps log datetime localtime show-timezone
```

## Related Commands

**clock set**
**debug** (Refer to the *Debug Command Reference* publication.)
**ntp**

# show aliases

To display all alias commands, or the alias commands in a specified mode, use the **show aliases** EXEC command.

**show aliases** [*mode*]

## Syntax Description

| | |
|---|---|
| *mode* | (Optional) Command mode. See Table 5-6 in the description of the **alias** command for acceptable options for the *mode* argument. |

## Command Mode

EXEC

## Usage Guidelines

All of the modes listed in Table 5-6 have their own prompts, except for the null interface mode. For example, the prompt for interface configuration mode is Router(config-if).

## Sample Display

The following is sample output from the **show aliases exec** commands. The aliases configured for commands in EXEC mode are displayed.

```
Router# show aliases exec

Exec mode aliases:
  h                     help
  lo                    logout
  p                     ping
  r                     resume
  s                     show
  w                     where
```

## Related Command

**alias**

# show buffers

Use the **show buffers** EXEC command to display statistics for the buffer pools on the network server.

**show buffers** [*type number* | **alloc** [**dump**]]

## Syntax Description

| | |
|---|---|
| *type number* | (Optional) Displays interface pool information. If the specified interface *type* and *number* has its own buffer pool, displays information for that pool. Value of *type* can be **ethernet**, **serial**, **tokenring, fddi, bri, atm, e1, t1**. |
| **alloc** | (Optional) Displays a brief listing of all allocated buffers. |
| **dump** | (Optional) Dumps all allocated buffers. This keyword must be used with the **alloc** keyword, not by itself. |

## Command Mode

EXEC

## Sample Displays

The following is sample output from the **show buffers** command with no arguments, showing all buffer pool information:

```
Router#show buffers
Buffer elements:
     398 in free list (500 max allowed)
     1266 hits, 0 misses, 0 created

Public buffer pools:
Small buffers, 104 bytes (total 50, permanent 50):
     50 in free list (20 min, 150 max allowed)
     551 hits, 0 misses, 0 trims, 0 created
Middle buffers, 600 bytes (total 25, permanent 25):
     25 in free list (10 min, 150 max allowed)
     39 hits, 0 misses, 0 trims, 0 created
Big buffers, 1524 bytes (total 50, permanent 50):
     49 in free list (5 min, 150 max allowed)
     27 hits, 0 misses, 0 trims, 0 created
VeryBig buffers, 4520 bytes (total 10, permanent 10):
     10 in free list (0 min, 100 max allowed)
     0 hits, 0 misses, 0 trims, 0 created
Large buffers, 5024 bytes (total 0, permanent 0):
     0 in free list (0 min, 10 max allowed)
     0 hits, 0 misses, 0 trims, 0 created
Huge buffers, 18024 bytes (total 0, permanent 0):
     0 in free list (0 min, 4 max allowed)
     0 hits, 0 misses, 0 trims, 0 created

Interface buffer pools:
Ethernet0 buffers, 1524 bytes (total 64, permanent 64):
     16 in free list (0 min, 64 max allowed)
     48 hits, 0 fallbacks
     16 max cache size, 16 in cache
Ethernet1 buffers, 1524 bytes (total 64, permanent 64):
```

```
     16 in free list (0 min, 64 max allowed)
     48 hits, 0 fallbacks
     16 max cache size, 16 in cache
Serial0 buffers, 1524 bytes (total 64, permanent 64):
     16 in free list (0 min, 64 max allowed)
     48 hits, 0 fallbacks
     16 max cache size, 16 in cache
Serial1 buffers, 1524 bytes (total 64, permanent 64):
     16 in free list (0 min, 64 max allowed)
     48 hits, 0 fallbacks
     16 max cache size, 16 in cache
TokenRing0 buffers, 4516 bytes (total 48, permanent 48):
     0 in free list (0 min, 48 max allowed)
     48 hits, 0 fallbacks
     16 max cache size, 16 in cache
TokenRing1 buffers, 4516 bytes (total 32, permanent 32):
     32 in free list (0 min, 48 max allowed)
     16 hits, 0 fallbacks

0 failures (0 no memory)
```

Table 5-17 describes significant fields shown in the display.

**Table 5-17 Show Buffers Field Descriptions**

| Field | Description |
|---|---|
| Buffer elements | Buffer elements are small structures used as placeholders for buffers in internal operating system queues. Buffer elements are used when a buffer may need to be on more than one queue. |
| Free | Total number of the currently unallocated buffer elements. |
| Max Free | Maximum number of buffers that are available for allocation. |
| Hit | Count of successful attempts to allocate a buffer when needed. |
| Miss | Count of buffer allocation attempts that resulted in growing the buffer pool in order to allocate a buffer. |
| Created | Count of new buffers created to satisfy buffer allocation attempts when the available buffers in the pool have already been allocated. |
| Public buffer pools | |
| Pool Name | Name of blocks of memory used to hold network packets. The sizes of these buffers can vary as follows: small, middle, big, large, verylarge, and huge. |
| Buffer Size | Size of this type of buffer, in bytes. |
| Total | Total number of this type of buffer. |
| Perm | Number of these buffers that are permanent. |
| Free | Number of available or unallocated buffers in that pool. |
| Min Free | Minimum number of free or unallocated buffers in the buffer pool |
| Max Free | Maximum number of free or unallocated buffers in the buffer pool |
| Hit | Count of successful attempts to allocate a buffer when needed. |
| Miss | Count of buffer allocation attemtps that resulted in growing the buffer pool in order to allocate a buffer. |
| Trim | Count of buffers released to the system because they were not being used. This field is displayed only for dynamic buffer pools, not interface buffer pools, which are static. |

| Field | Description |
|---|---|
| Created | Count of new buffers created in response to misses. This field is displayed only for dynamic buffer pools, not interface buffer pools, which are static. |
| Need | Difference between the number of permanent buffers of that type that present and the number of permanent buffers configured. Only displayed if non-zero. |
| Extra | Difference between the number of permanent buffers of that type configured and the number of permanent buffers present. Only displayed if non-zero. |
| Interface buffer pools | |
| Pool Name | Interface type and number. |
| Buffer Size | Size of this type of buffer, in bytes. |
| Total | Total number of this type of buffer. |
| Perm | Number of these buffers that are permanent. |
| Free | Number of available or unallocated buffers in that pool. |
| Min Free | Minimum number of free or unallocated buffers in the buffer pool. |
| Max Free | Maximum number of free or unallocated buffers in the buffer pool. |
| Hit | Count of successful attempts to allocate a buffer when needed. |
| Fall back | Count of buffer allocation attempts that resulted in falling back to the public buffer pool that is the smallest pool at least as big as the interface buffer pool. |
| Cache Max | Maximum number of buffers from that interface's pool that can bbe in that interface buffer pool's cache. Each interface buffer pool has its own cache. These are not additional to the permanent buffers; they come from the interface's buffer pools. Some interfaces place all of their buffers from the interface pool into the cache. In this case, it is normal for the Free column to display 0. |
| Cache Free | Number of unallocated buffers in the interface pool's buffer cache. |
| failures | Total number of allocation requests that have failed because no buffer was available for allocation; the datagram was lost. Such failures normally occur at interrupt level. |
| (no memory) | Number of failures that occurred because no memory was available to create a new buffer. |

The following is sample output from the **show buffers** command with an interface *type* and *number* :

```
Router#show buffers Ethernet 0
Ethernet0 buffers, 1524 bytes (total 64, permanent 64):
     16 in free list (0 min, 64 max allowed)
     48 hits, 0 fallbacks
     16 max cache size, 16 in cache
```

The following is sample output from the **show buffers** command when **alloc** is specified:

```
Router#show buffers alloc
Buffer elements:
     398 in free list (500 max allowed)
     1266 hits, 0 misses, 0 created

Public buffer pools:
Small buffers, 104 bytes (total 50, permanent 50):
     50 in free list (20 min, 150 max allowed)
     551 hits, 0 misses, 0 trims, 0 created
Middle buffers, 600 bytes (total 25, permanent 25):
     25 in free list (10 min, 150 max allowed)
```

```
     39 hits, 0 misses, 0 trims, 0 created
Big buffers, 1524 bytes (total 50, permanent 50):
     49 in free list (5 min, 150 max allowed)
     27 hits, 0 misses, 0 trims, 0 created
VeryBig buffers, 4520 bytes (total 10, permanent 10):
     10 in free list (0 min, 100 max allowed)
     0 hits, 0 misses, 0 trims, 0 created
Large buffers, 5024 bytes (total 0, permanent 0):
     0 in free list (0 min, 10 max allowed)
     0 hits, 0 misses, 0 trims, 0 created
Huge buffers, 18024 bytes (total 0, permanent 0):
     0 in free list (0 min, 4 max allowed)
     0 hits, 0 misses, 0 trims, 0 created

Interface buffer pools:
Ethernet0 buffers, 1524 bytes (total 64, permanent 64):
     16 in free list (0 min, 64 max allowed)
     48 hits, 0 fallbacks
     16 max cache size, 16 in cache
Ethernet1 buffers, 1524 bytes (total 64, permanent 64):
     16 in free list (0 min, 64 max allowed)
     48 hits, 0 fallbacks
     16 max cache size, 16 in cache
Serial0 buffers, 1524 bytes (total 64, permanent 64):
     16 in free list (0 min, 64 max allowed)
     48 hits, 0 fallbacks
     16 max cache size, 16 in cache
Serial1 buffers, 1524 bytes (total 64, permanent 64):
     16 in free list (0 min, 64 max allowed)
     48 hits, 0 fallbacks
     16 max cache size, 16 in cache
TokenRing0 buffers, 4516 bytes (total 48, permanent 48):
     0 in free list (0 min, 48 max allowed)
     48 hits, 0 fallbacks
     16 max cache size, 16 in cache
TokenRing1 buffers, 4516 bytes (total 32, permanent 32):
     32 in free list (0 min, 48 max allowed)
     16 hits, 0 fallbacks

0 failures (0 no memory)

Address  PakAddr  Data     Off Data  Pool  Ref Link Enc     Flags Output  Input
                  Area     set Size        Cnt Type Type    (Hex) Idb     Idb
--------------------------------------------------------------------------------
604B37A0 604B37C0 40004A38  62    60 Big     1   65    3        0 Et0
604C6DA0 604C6DC0 40007038  84     0 Ether   1    0    0        0
604C6F60 604C6F80 400076E4  84     0 Ether   1    0    0        0
604C7120 604C7140 40007D90  84     0 Ether   1    0    0        0
604C72E0 604C7300 4000843C  84     0 Ether   1    0    0        0
604C74A0 604C74C0 40008AE8  84     0 Ether   1    0    0        0
604C7660 604C7680 40009194  84     0 Ether   1    0    0        0
604C7820 604C7840 40009840  84     0 Ether   1    0    0        0
```

# show calendar

To display the calendar hardware setting for the Cisco 7000 or Cisco 4500, use the **show calendar** EXEC command:

**show calendar**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

You can compare the time and date shown with this command with the time and date listed via the **show clock** command to verify that the calendar and system clock are in sync with each other. The time displayed is relative to the configured time zone.

## Sample Display

In the following sample display, the hardware calendar indicates the timestamp of 12:13:44 p.m. on Friday, January 1, 1993:

```
Router# show calendar

12:13:44 PST Fri Jan 1 1993
```

## Related Command

**show clock**

# show cdp

To display global CDP information , including timer and hold-time information, use the **show cdp** privileged EXEC command.

**show cdp**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Privileged EXEC

## Sample Display

The following is sample output from the **show cdp** command. Global CDP timer and hold-time parameters are set to the defaults of 60 and 180 seconds, respectively.

```
Router# show cdp

Global CDP information:
        Sending CDP packets every 60 seconds
        Sending a holdtime value of 180 seconds
```

## Related Commands

**cdp holdtime**
**cdp timer**
**show cdp entry**
**show cdp neighbors**

# show cdp entry

To display information about a neighbor device listed in the CDP table, use the **show cdp entry** privileged EXEC command.

**show cdp entry** *entry-name* [**protocol** | **version**]

## Syntax Description

| | |
|---|---|
| *entry-name* | Name of neighbor about which you want information. |
| **protocol** | (Optional) Limits the display to information about the protocols enabled on a device. |
| **version** | (Optional) Limits the display to information about the version of software running on the device. |

## Command Mode

Privileged EXEC

## Sample Displays

The following is sample output from the **show cdp** entry command with no limits. Information about the neighbor *device.cisco.com* is displayed, including device ID, address and protocol, platform, interface, hold time, and version.

```
Router# show cdp entry device.cisco.com

Device ID: device.cisco.com
Entry address(es):
  IP address: 198.92.68.18
  CLNS address: 490001.1111.1111.1111.00
  DECnet address: 10.1
Platform: AGS,  Capabilities: Router Trans-Bridge
Interface: Ethernet0,  Port ID (outgoing port): Ethernet0
Holdtime : 155 sec

Version :
GS Software (GS3), Experimental Version 10.2(10302) [asmith 161]
Copyright (c) 1986-1994 by cisco Systems, Inc.
Compiled Mon 07-Nov-94 14:34
```

The following is sample output from the **show cdp entry privilege** command.Only information about the protocols enabled on *neon-cisco.com* is displayed.

```
Router# show cdp entry device.cisco.com protocol

Protocol information for device.cisco.com :
  IP address: 198.92.68.18
  CLNS address: 490001.1111.1111.1111.00
  DECnet address: 10.1
```

The following is sample output from the **show cdp entry version** command.Only information about the version of software running on *device.cisco.com* is displayed.

```
Router# show cdp entry device.cisco.com version

Version information for device.cisco.com :
```

```
  GS Software (GS3), Experimental Version 10.2(10302) [asmith 161]
Copyright (c) 1986-1994 by cisco Systems, Inc.
Compiled Mon 07-Nov-94 14:34
```

## Related Command

**show cdp neighbors**

# show cdp interface

To display information about the interfaces on which CDP is enabled, use the **show cdp interface** command.

**show cdp interface** [*type number*]

## Syntax Description

| | |
|---|---|
| *type* | (Optional) Type of interface about which you want information. |
| *number* | (Optional) Number of the interface about which you want information. |

## Command Mode

Privileged EXEC

## Sample Displays

The following sample output form the **show cdp interface** command. Status information and information about CDP timer and hold time settings is displayed for all interfaces on which CDP is enabled.

```
Router# show cdp interface

Serial0 is up, line protocol is up, encapsulation is SMDS
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Ethernet0 is up, line protocol is up, encapsulation is ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
```

The following is sample output from the **show cdp interface** command with an interface specified. Status information and information about CDP timer and holdtime settings is displayed for Ethernet interface 0 only.

```
Router# show cdp interface ethernet 0

Ethernet0 is up, line protocol is up, encapsulation is ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
```

# show cdp neighbors

To display information about neighbors, use the **show cdp neighbors** privileged EXEC command.

**show cdp neighbors** [*interface-type interface-number*] [**detail**]

## Syntax Description

| | |
|---|---|
| *interface-type* | (Optional) Type of the interface connected to the neighbors about which you want information. |
| *interface-number* | (Optional) Number of the interface connected to the neighbors about which you want information. |
| **detail** | (Optional) Displays detailed information about a neighbor (or neighbors) including network address, enabled protocols, hold time, and software version. |

## Command Mode

Privileged EXEC

## Sample Displays

The following is sample output from the **show cdp neighbors** command. Device ID, interface type and number, holdtime settings, capabilities, platform, and port ID information about the router's neighbors is displayed.

```
Router# show cdp neighbors

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
device.cisco.com      Eth 0           151         R T        AGS       Eth 0
device.cisco.com      Ser 0           165         R T        AGS       Ser 3
```

The following is sample output from the **show cdp neighbors detail** command. Additional detail is shown about the router's neighbors, including network address, enabled protocols, and software version:

```
Router# show cdp neighbors detail

Device ID: device.cisco.com
Entry address(es):
  IP address: 198.92.68.18
  CLNS address: 490001.1111.1111.1111.00
  DECnet address: 10.1
Platform: AGS,  Capabilities: Router Trans-Bridge
Interface: Ethernet0,  Port ID (outgoing port): Ethernet0
Holdtime : 143 sec

Version :
GS Software (GS3), Experimental Version 10.2(10302) [asmith 161]
Copyright (c) 1986-1994 by cisco Systems, Inc.
Compiled Mon 07-Nov-94 14:34
```

## Related Command

**show cdp entry**

# show cdp traffic

To display traffic information from the CDP table, use the **show cdp traffic** privileged EXEC command.

**show cdp traffic**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Privileged EXEC

## Sample Display

The following is sample output from the **show cdp traffic** command.

```
Router# show cdp traffic

CDP counters :
        Packets output: 94, Input: 75
        Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
        No memory: 0, Invalid packet: 0, Fragmented: 0
```

In this example, traffic information is displayed including the numbers of packets sent, the number of packets received, header syntax, checksum errors, failed encapsulations, memory problems, and invalid and fragmented packets is displayed. Header syntax indicates the number of packets CDP receives with that have an invalid header format.

# show clock

To display the system clock, use the **show clock** EXEC command:

**show clock** [**detail**]

## Syntax Description

| | |
|---|---|
| **detail** | (Optional) Indicates the clock source (NTP, VINES, 7000 calendar, and so forth) and the current summer-time setting (if any). |

## Command Mode

EXEC

## Usage Guidelines

The system clock keeps an "authoritative" flag that indicates whether or not the time is authoritative (believed to be accurate). If system clock has been set by a timing source (Cisco 7000 calendar, NTP, VINES, and so forth), the flag is set. If the time is not authoritative, it will be used only for display purposes. Until the clock is authoritative and the "authoritative" flag is set, the flag prevents the router from causing peers to synchronize to itself when the router time is invalid.

The symbol that precedes the **show clock** display indicates the following:

An asterisk `(*) i`ndicates not authoritative

A blank space indicates authoritative

A period (. ) indicates authoritative, but NTP is not synchronized.

## Sample Display

The following sample output shows that the current clock is authoritative and that the time source is NTP:

```
Router# show clock detail
15:29:03.158 PST Mon Mar 1 1993
Time source is NTP
Router#
```

## Related Commands

**clock set**
**show calendar**

# show environment

Use the **show environment** EXEC command to display temperature and voltage information on the AGS+ and Cisco 7000 series console.

**show environment**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

Once a minute a routine is run that gets environmental measurements from the CSC-ENVM card and stores the **show environment** output into a buffer. This buffer is displayed on the console when **show environment** is invoked.

If a measurement exceeds desired margins, but has not exceeded fatal margins, a warning message is printed to the system console. The system software queries the CSC-ENVM card for measurements once a minute, but warnings for a given testpoint are printed at most once every four hours. If a measurement is out of line within a four-hour period, an automatic warning message appears on the console. As noted above, you can query the CSC-ENVM using the **show environment** command at any time to determine if a measurement is at the warning tolerance.

## Sample Display

The following is sample output from the **show environment** command on the AGS+:

```
Router# show environment

Environmental controller firmware version 2.0
   Serial number is 00220846, calibrated on 2-14-92, by technician rma
   Internal temperature measured 34.3(C), shuts down at 43.0(C)
   Air flow appears good.
   +5 volt line measured at 5.061(V)
   +12 volt line measured at 12.120(V)
   -12 volt line measured at -11.936(V)
   -5 volt line measured at -4.986(V)
```

Table 5-18 describes significant fields shown in the display.

**Table 5-18 Show Environment Field Descriptions for AGS+**

| Field | Description |
|---|---|
| Serial number is 00220846 | Serial number of router. |
| calibrated on 2-14-92 | Date on which these measurements were taken. |
| by technician rma | ID (initials in this case) of the technician taking the measurement. |
| Internal temperature measured 34.3 (C) | Internal temperature of the router (in celsius). |

| Field | Description |
|---|---|
| shuts down at 43.0(C) | Temperature (in celsius) at which the router is administratively shut down to prevent internal damage. |
| Air flow appears good. | Air flow is adequate for proper router operation. |
| +5 volt line at 5.061(V) | Voltage measurement of the +5 volt line. |
| +12 volt line measured at 12.120(V) | Voltage measurement of the +12 volt line. |
| –12 volt line measured at –11.936(V) | Voltage measurement of the –12 volt line. |
| –5 volt line measured at –4.986(V) | Voltage measurement of the –5 volt line. |

The following is an example of a message that displays on the system console when a measurement has exceeded an acceptable margin:

```
Router#
ENVIRONMENTAL WARNING: Air flow appears marginal.
```

The following is an example of a message that displays on the system console when a measurement has exceeded an acceptable margin. In this example, the internal temperature reading is given:

```
Router#
ENVIRONMENTAL WARNING: Internal temperature measured 41.3(C)
```

The following is an example of a message that displays on the system console when a voltage measurement has exceeded an acceptable margin:

```
Router#
ENVIRONMENTAL WARNING:  +5 volt testpoint measured 5.310(V)
```

If the CSC-ENVM card on the AGS+ chassis detects that any of its voltage or temperature testpoints has exceeded maximum margins, it does the following in this order:

1. Saves the last measured values from each of the six testpoints 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 supply after a few milliseconds of delay.

The following is the message the system displays if voltage or temperature exceed maximum margins:

```
Router#
SHUTDOWN: air flow problem
```

For environmental specifications, refer to the *Hardware Installation and Maintenance* publication for your individual chassis.

The following example shows the typical **show environment** display on the Cisco 7000 when there are no warning conditions in the system. The date and time of the query are displayed, along with the data refresh information and a message indicating that there are no warning conditions.

```
Router> show environment
Environmental Statistics
  Environmental status as of 13:17:39 UTC Thu Oct 22 1992
  Data is 7 second(s) old, refresh in 53 second(s)

  All Environmental Measurements are within specifications
```

Table 5-19 describes the **show environment** display fields on the Cisco 7000.

**Table 5-19 Show Environment Field Descriptions for Cisco 7000**

| Field | Description |
|---|---|
| Environmental status as of... | Current date and time. |
| Data age and refresh | Environmental measurements are output into a buffer every 60 seconds, unless other higher-priority processes are running. |
| WARNING | If environmental measurements are not within specification, warning messages are displayed. |

# show environment all

Use the **show environment all** EXEC command to display temperature and voltage information on the Cisco 7000 series console.

**show environment all**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show environment all** command on the Cisco 7000 when there are no warning conditions in the system:

```
7000> show environment all

Environmental Statistics
  Environmental status as of 13:17:39 UTC Thu Oct 22 1992
  Data is 11 second(s) old, refresh in 49 second(s)

  All Environmental Measurements are within specifications

  Lower Power Supply: 700W, ON     Upper Power Supply: Not Installed

  No Intermittent Powerfails

  +12 volt measured at  12.05(V)
   +5 volt measured at   4.92(V)
  -12 volt measured at -12.00(V)
  +24 volt measured at  23.80(V)

Airflow temperature measured at 30(C)
Inlet temperature measured at 25(C)
```

In the following example, there have been two intermittent power failures since the router was turned on, and the lower power supply is not functioning. The last intermittent power failure occurred on Sunday, October 25, 1992, at 11:07 p.m.

```
7000# show environment all

Environmental Statistics
  Environmental status as of 23:19:47 UTC Sun Oct 25 1992
  Data is 6 second(s) old, refresh in 54 second(s)

  WARNING: Lower Power Supply is NON-OPERATIONAL

  Lower Power Supply:700W, OFF     Upper Power Supply: 700W, ON

  Intermittent Powerfail(s): 2     Last on 23:07:05 UTC Sun Oct 25 1992

  +12 volts measured at  12.05(V)
   +5 volts measured at   4.96(V)
  -12 volts measured at -12.05(V)
  +24 volts measured at  23.80(V)
```

```
    Airflow temperature measured at 38(C)
    Inlet temperature measured at 25(C)
```

Table 5-20 describes the **show environment all** display fields.

**Table 5-20 Show Environment All Field Descriptions**

| Field | Description |
|---|---|
| Environmental status as of... | Date and time of last query. |
| Data age and refresh | Environmental measurements are output into a buffer every 60 seconds, unless other higher-priority processes are running. |
| WARNING | If environmental measurements are not within specification, warning messages are displayed. |
| `Lower Power Supply` | Type of power supply installed and its status (on or off). |
| `Upper Power Supply` | Type of power supply installed and its status (on or off). |
| `Intermittent Powerfails` | Number of power hits (not resulting in shutdown) since system was last booted. |
| `Voltage Specifications` | System voltage measurements. |
| `Airflow and Inlet temperature` | Temperature of air coming in and going out. |

The following example shows typical output of the **show environment all** command on the Cisco 7010. The output shows the status of the single 600W power supply. The following example from a Cisco 7010 shows that a single 600W power supply is installed:

```
7010# show environment all

Environmental Statistics
  Environmental status as of Fri 11-5-1993 19:10:41
  Data is 31 second(s) old, refresh in 29 second(s)

 All Environmental Measurements are within specifications

  Power Supply: 600W AC

  No Intermittent Powerfails

  +12 volts measured at  12.00(V)
   +5 volts measured at   5.02(V)
  -12 volts measured at -12.05(V)
  +24 volts measured at  23.70(V)

  Airflow temperature measured at 35(C)
  Inlet   temperature measured at 26(C)
```

Table 5-21 describes the fields shown in the display.

**Table 5-21 Show Environment Field Descriptions for the Cisco 7010**

| Field | Description |
|---|---|
| Environmental status as of... | Current date and time. |

| Field | Description |
|---|---|
| Data age and refresh | Environmental measurements are output into a buffer every 60 seconds, unless other higher-priority processes are running. |
| All Environmental Measurements are within specifications | All environment measurements are within specification. If they are not, warning messages are displayed. |
| Power Supply: | Type of power supply. |
| No Intermittent Powerfails | Indicates whether intermittent power failures are occurring. |
| +12 volts measured at 12.00(V) | Voltage measurement of the +12 volt line. |
| +5 volts measured at 5.02(V) | Voltage measurement of the +5 volt line. |
| –12 volts measured at –12.05(V) | Voltage measurement of the –12 volt line. |
| +24 volts measured at 23.70(V) | Voltage measurement of the +24 volt line. |

# show environment last

If a shutdown occurs due to detection of fatal environmental margins, the CSC-ENVM (on the AGS+) or the route processor (RP) (on the Cisco 7000 series) logs the last measured value from each of the six test points to internal nonvolatile memory. Only one set of measurements may be stored at any one time.

Use the **show environment last** EXEC command to display these test points.

**show environment last**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show environment last** command on the AGS+:

```
Router# show environment last

Environmental controller firmware version 2.0
   Serial number is 3232, calibrated on 2-14-92, by technician rma
   Internal temperature measured 24.1(C), shuts down at 43.0(C)
   Air flow appears good.
   +5 volt line measured at 4.988(V)
   +12 volt line measured at 12.044(V)
   -12 volt line measured at -11.787(V)
   -5 volt line measured at -4.939(V)

LAST Environmental Shutdown Measurements:
   Internal temperature was 24.0(C)
   Air flow sensor was good
   +5 volt line was 4.990(V)
   +12 volt line was 9.900(V)*
   -12 volt line was -11.719(V)
   -5 volt line was -4.926(V)
```

As the display shows, the first block of data is equivalent to **show environment**, in that it displays the current measurements. The second block shows all the testpoint values at the time of the LAST environmental shutdown. An asterisk suffixes the testpoint that caused the failure. In this example, the +12 volt testpoint dropped to 9.900(V) to cause the shutdown.

The following example is for the Cisco 7000. The router retrieves the environmental statistics at the time of the last shutdown. In this example, the last shutdown was Tuesday, May 19, 1992 at 12:40p.m., so the environmental statistics at that time are displayed.

```
Router# show environment last

Environmental Statistics
  Environmental status as of 14:47:00 UTC Thu May 21 1992
  Data is 6 second(s) old, refresh in 54 second(s)

  WARNING: Upper Power Supply is NON-OPERATIONAL

LAST Environmental Statistics
```

```
Environmental status as of 12:40:00 UTC Tues May 19 1992
Lower Power Supply: 700W, ON     Upper Power Supply: 700W, OFF

No Intermittent Powerfails

+12 volts measured at  12.05(V)
 +5 volts measured at   4.98(V)
-12 volts measured at -12.00(V)
+24 volts measured at  23.80(V)

Airflow temperature measured at 30(C)
Inlet   temperature measured at 23(C)
```

Table 5-22 describes the **show environment last** display fields.

**Table 5-22 Show Environment Last Field Descriptions**

| Field | Description |
|---|---|
| Environmental status as of... | Current date and time. |
| Data age and refresh | Environmental measurements are output into a buffer every 60 seconds, unless other higher-priority processes are running. |
| WARNING | If environmental measurements are not within specification, warning messages are displayed. |
| `LAST` | Displays test point values at time of the last environmental shutdown. |
| `Lower Power Supply/Upper Power Supply` `Power Supply:` | For the Cisco 7000, indicates the status of the two 700W power supplies. For the Cisco 7010, indicates the status of the single 600W power supply. |

# show environment table

Use the **show environment table** EXEC command to display environmental measurements and a table that lists the ranges of environment measurement that are within specification. This command is available on the Cisco 7000 only.

**show environment table**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following sample output shows the current environmental status in tables that list voltage and temperature parameters. There are three warning messages; one each about the lower power supply, the airflow temperature, and the inlet temperature. In this example, voltage parameters are shown to be in the normal range, airflow temperature is at a critical level, and inlet temperature is at the warning level.

```
Router> show environment table
Environmental Statistics
  Environmental status as of Mon 11-2-1992 17:43:36
  Data is 52 second(s) old, refresh in 8 second(s)

  WARNING: Lower Power Supply is NON-OPERATIONAL
  WARNING: Airflow temperature has reached CRITICAL level at 73(C)
  WARNING: Inlet temperature has reached WARNING level at 41(C)

Voltage Parameters:

 SENSE        CRITICAL               NORMAL               CRITICAL
-------|--------------------|------------------------|--------------------

+12(V)                   10.20      12.05(V)      13.80
 +5(V)                    4.74       4.98(V)       5.26
-12(V)                  -10.20     -12.05(V)     -13.80
+24(V)                   20.00      24.00(V)      28.00

Temperature Parameters:

 SENSE     WARNING      NORMAL      WARNING      CRITICAL      SHUTDOWN
-------|-------------|------------|-------------|--------------|--------------

Airflow           10           60           70    73(C)   88

Inlet             10           39    41(C)  46            64
```

Table 5-23 describes the **show environment table** display fields.

**Table 5-23 Show Environment Table Field Descriptions**

| Field | Description |
|---|---|
| SENSE (Voltage Parameters) | Voltage specification for DC line. |

| Field | Description |
|---|---|
| SENSE (Temperature Parameters) | Air being measured. Inlet measures the air coming in, and Airflow measures the temperature of the air inside the chassis. |
| NORMAL | All monitored conditions meet normal requirements. |
| WARNING | System is approaching an out-of-tolerance condition. |
| CRITICAL | Out-of-tolerance condition exists. |
| PROCESSOR SHUTDOWN | Processor has detected condition that could cause physical damage to the system. |

# show logging

Use the **show logging** EXEC command to display the state of logging (syslog).

**show logging**

This command displays the state of syslog error and event logging, including host addresses, and whether console logging is enabled. This command also displays Simple Network Management Protocol (SNMP) configuration parameters and protocol activity.

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show logging** command:

```
Router# show logging

Syslog logging: enabled
     Console logging: disabled
     Monitor logging: level debugging, 266 messages logged.
     Trap logging: level informational, 266 messages logged.
     Logging to 131.108.2.238

SNMP logging: disabled, retransmission after 30 seconds
    0 messages logged
```

Table 5-24 describes significant fields shown in the display.

**Table 5-24 Show Logging Field Descriptions**

| Field | Description |
|---|---|
| Syslog logging | When enabled, system logging messages are sent to a UNIX host that acts as a syslog server; that is, it captures and saves the messages. |
| Console logging | If enabled, states the level; otherwise, this field displays disabled. |
| Monitor logging | Minimum level of severity required for a log message to be sent to a monitor terminal (not the console). |
| Trap logging | Minimum level of severity required for a log message to be sent to a syslog server. |
| SNMP logging | Shows whether SNMP logging is enabled and the number of messages logged, and the retransmission interval. |

# show memory

Use the **show memory** EXEC command to show statistics about the router's memory, including memory free pool statistics.

**show memory** [*type*] [**free**]

## Syntax Description

| | |
|---|---|
| *type* | (Optional) Memory type to display (**processor**, **multibus**, **io**, **sram**). If *type* is not specified, statistics for all memory types present in the router will be displayed. |
| **free** | (Optional) Displays free memory statistics. |

## Command Mode

EXEC

## Sample Displays

The following is sample output from the **show memory** command:

```
Router# show memory

               Head  FreeList    Total(b)     Used(b)     Free(b)  Largest(b)
Processor    2E0FF8    2AABFC    13758472      847216    12911256    12908036

          Processor memory

Address   Bytes Prev.   Next     Ref  PrevF   NextF   Alloc PC  What
2E0FF8     2128 0       2E1848     1                  84352     *Init*
2E1848     2052 2E0FF8  2E204C     1                  86184     *Init*
2E204C      564 2E1848  2E2280     1                  861B0     *Init*
2E2280     2052 2E204C  2E2A84     1                  1266      *Init*
2E2A84      308 2E2280  2E2BB8     1                  44974     *Init*
2E2BB8      220 2E2A84  2E2C94     1                  3F788     *Init*
2E2C94     2052 2E2BB8  2E3498     1                  3F7A8     *Init*
2E3498     4052 2E2C94  2E446C     1                  46770     *Init*
2E446C      516 2E3498  2E4670     1                  44E4C     *Packet Buffer*
2E4670      516 2E446C  2E4874     1                  44E4C     *Packet Buffer*
2E4874      516 2E4670  2E4A78     1                  44E4C     *Packet Buffer*
2E4A78      516 2E4874  2E4C7C     1                  44E4C     *Packet Buffer*
2E4C7C      516 2E4A78  2E4E80     1                  44E4C     *Packet Buffer*
2E4E80      516 2E4C7C  2E5084     1                  44E4C     *Packet Buffer*
2E5084      516 2E4E80  2E5288     1                  44E4C     *Packet Buffer*
2E5288      516 2E5084  2E548C     1                  44E4C     *Packet Buffer*
2E548C      516 2E5288  2E5690     1                  44E4C     *Packet Buffer*
2E5690      516 2E548C  2E5894     1                  44E4C     *Packet Buffer*
Router#
```

The following is sample output from the **show memory free** command:

```
Router# show memory free

               Head  FreeList    Total(b)     Used(b)     Free(b)  Largest(b)
Processor    2E0FF8    2AABFC    13758472      847120    12911352    12908036

          Processor memory
```

```
Address   Bytes Prev.   Next     Ref  PrevF   NextF   Alloc PC  What

             72    Free list 1

             88    Free list 2

             96    Free list 3
384A04       96 38496C  384A64     0  0       0       1205A4    IGRP Router

            108    Free list 4

            124    Free list 5

                   Final freespace block
3B09FC 12908036 3B0834  0          0  0       0       76162     (coalesced)
```

The display of **show memory free** contains the same types of information as the **show memory** display, except that only free memory is displayed, and the information is displayed in order for each free list.

The first section of the display includes summary statistics about the activities of the system memory allocator. Table 5-25 describes significant fields shown in the first section of the display.

**Table 5-25 Show Memory Field Descriptions—First Section**

| Field | Description |
|---|---|
| Head | Hexadecimal address of the head of the memory allocation chain. |
| FreeList | Hexadecimal address of the base of the free list. |
| Total (b) | Sum of used bytes plus free bytes. |
| Used (b) | Amount of memory in use. |
| Free (b) | Amount of memory not in use. |
| Largest (b) | Size of largest available free block. |

The second section of the display is a block-by-block listing of memory use. Table 5-26 describes significant fields shown in the second section of the display.

**Table 5-26 Characteristics of Each Block of Memory—Second Section**

| Field | Description |
|---|---|
| Address | Hexadecimal address of block. |
| Bytes | Size of block in bytes. |
| Prev. | Address of previous block (should match Address on previous line). |
| Next | Address of next block (should match address on next line). |
| Ref | Reference count for that memory block, indicating how many different processes are using that block of memory. |
| PrevF | Address of previous free block (if free). |
| NextF | Address of next free block (if free). |
| Alloc PC | Address of the system call that allocated the block. |

| Field | Description |
|---|---|
| What | Name of process that owns the block, or "(fragment)" if the block is a fragment, or "(coalesced)" if the block was coalesced from adjacent free blocks. |

The **show memory io** command displays the free IO memory blocks. On the Cisco 4000, this command quickly shows how much unused IO memory is available.

The following is sample output from the **show memory io** command:

```
Router# show memory io

Address   Bytes Prev.   Next     Ref  PrevF   NextF   Alloc PC  What
6132DA0   59264 6132664 6141520  0    0      600DDEC  3FCF0     *Packet Buffer*
600DDEC     500 600DA4C 600DFE0  0   6132DA0 600FE68 0
600FE68     376 600FAC8 600FFE0  0   600DDEC 6011D54 0
6011D54     652 60119B4 6011FEO  0   600FE68 6013D54 0
614FCA0     832 614F564 614FFE0  0   601FD54 6177640 0
6177640 2657056 6172E90 0        0   614FCA0 0       0
Total: 2723244
```

The **show memory sram** command displays the free SRAM memory blocks. For the Cisco 4000, this command supports the high-speed static RAM memory pool to make it easier to debug or diagnose problems with allocation or freeing of such memory.

The following is sample output from the **show memory sram** command:

```
Router# show memory sram

Address   Bytes Prev.   Next     Ref  PrevF   NextF   Alloc PC  What
7AE0      38178 72F0    0        0    0       0       0
Total     38178
```

The **show memory** command on the Cisco 4000 includes information about SRAM memory and IO memory, and appears as follows:

```
Router# show memory

               Head  Free Start Total Bytes     Used Bytes Free Bytes
SRAM           1000      7AE0      65538        27360        38178
Processor    20CFC4    23E178    2043964       282372      1761592
IO memory   6000000   6132DA0    4194656      1471412      2723244

Address   Bytes Prev.   Next     Ref  PrevF   NextF   Alloc PC  What
1000       2032 0       17F0       1                  3E73E     *Init*
17F0       2032 1000    1FE0       1                  3E73E     *Init*
1FE0        544 17F0    2200       1                  3276A     *Init*
2200         52 1FE0    2234       1                  31D68     *Init*
2234         52 2200    2268       1                  31DAA     *Init*
2268         52 2234    229C       1                  31DF2     *Init*
72F0       2032 6E5C    7AE0       1                  3E73E     Init
7AE0      38178 72F0    0          0    0      0      0
Router#
```

# show ntp associations

To show the status of Network Time Protocol (NTP) associations, use the **show ntp associations** EXEC command.

**show ntp associations** [**detail**]

## Syntax Description

| | |
|---|---|
| **detail** | (Optional) Shows detailed information about each NTP association. |

## Command Mode

EXEC

## Sample Displays

Detailed descriptions of the information displayed by this command can be found in the NTP specification (RFC 1305).

The following is sample output from the **show ntp associations** command:

```
Router# show ntp associations
     address         ref clock     st  when  poll reach  delay  offset    disp
 ~160.89.32.2      160.89.32.1       5    29  1024  377     4.2   -8.59     1.6
+~131.108.13.33    131.108.1.111     3    69   128  377     4.1    3.48     2.3
*~131.108.13.57    131.108.1.111     3    32   128  377     7.9   11.18     3.6
* master (synced), # master (unsynced), + selected, - candidate, ~ configured
Router#
```

Table 5-27 describes significant fields shown in the display.

**Table 5-27 Show NTP Associations Field Descriptions**

| Field | Description |
|---|---|
| address | Address of peer. |
| ref clock | Address of peer's reference clock. |
| st | Peer's stratum. |
| when | Time since last NTP packet received from peer. |
| poll | Polling interval (seconds). |
| reach | Peer reachability (bit string, in octal). |
| delay | Round-trip delay to peer (milliseconds). |
| offset | Relative time of peer’s clock to local clock (milliseconds). |
| disp | Dispersion |
| The first character of the line can be one or more of the following: | |
| * | Synchronized to this peer. |
| # | Almost synchronized to this peer. |
| + | Peer selected for possible synchronization. |
| - | Peer is a candidate for selection. |
| ~ | Peer is statically configured. |

The following is sample output of the **show ntp associations detail** command:

```
Router# show ntp associations detail
160.89.32.2 configured, insane, invalid, stratum 5
ref ID 160.89.32.1, time AFE252C1.6DBDDFF2 (00:12:01.428 PDT Mon Jul 5 1993)
our mode active, peer mode active, our poll intvl 1024, peer poll intvl 64
root delay 137.77 msec, root disp 142.75, reach 376, sync dist 215.363
delay 4.23 msec, offset -8.587 msec, dispersion 1.62
precision 2**19, version 3
org time AFE252E2.3AC0E887 (00:12:34.229 PDT Mon Jul 5 1993)
rcv time AFE252E2.3D7E464D (00:12:34.240 PDT Mon Jul 5 1993)
xmt time AFE25301.6F83E753 (00:13:05.435 PDT Mon Jul 5 1993)
filtdelay =     4.23    4.14    2.41    5.95    2.37    2.33    4.26    4.33
filtoffset =   -8.59   -8.82   -9.91   -8.42  -10.51  -10.77  -10.13  -10.11
filterror =     0.50    1.48    2.46    3.43    4.41    5.39    6.36    7.34

131.108.13.33 configured, selected, sane, valid, stratum 3
ref ID 131.108.1.111, time AFE24F0E.14283000 (23:56:14.078 PDT Sun Jul 4 1993)
our mode client, peer mode server, our poll intvl 128, peer poll intvl 128
root delay 83.72 msec, root disp 217.77, reach 377, sync dist 264.633
delay 4.07 msec, offset 3.483 msec, dispersion 2.33
precision 2**6, version 3
org time AFE252B9.713E9000 (00:11:53.442 PDT Mon Jul 5 1993)
rcv time AFE252B9.7124E14A (00:11:53.441 PDT Mon Jul 5 1993)
xmt time AFE252B9.6F625195 (00:11:53.435 PDT Mon Jul 5 1993)
filtdelay =     6.47    4.07    3.94    3.86    7.31    7.20    9.52    8.71
filtoffset =    3.63    3.48    3.06    2.82    4.51    4.57    4.28    4.59
filterror =     0.00    1.95    3.91    4.88    5.84    6.82    7.80    8.77

131.108.13.57 configured, our_master, sane, valid, stratum 3
ref ID 131.108.1.111, time AFE252DC.1F2B3000 (00:12:28.121 PDT Mon Jul 5 1993)
our mode client, peer mode server, our poll intvl 128, peer poll intvl 128
root delay 125.50 msec, root disp 115.80, reach 377, sync dist 186.157
delay 7.86 msec, offset 11.176 msec, dispersion 3.62
precision 2**6, version 2
org time AFE252DE.77C29000 (00:12:30.467 PDT Mon Jul 5 1993)
rcv time AFE252DE.7B2AE40B (00:12:30.481 PDT Mon Jul 5 1993)
xmt time AFE252DE.6E6D12E4 (00:12:30.431 PDT Mon Jul 5 1993)
filtdelay =    49.21    7.86    8.18    8.80    4.30    4.24    7.58    6.42
filtoffset =   11.30   11.18   11.13   11.28    8.91    9.09    9.27    9.57
filterror =     0.00    1.95    3.91    4.88    5.78    6.76    7.74    8.71
```

Table 5-28 describes significant fields shown in the display.

**Table 5-28 Show NTP Associations Detail Field Descriptions**

| Field | Descriptions |
|---|---|
| configured | Peer was statically configured. |
| dynamic | Peer was dynamically discovered. |
| our_master | Local machine is synchronized to this peer. |
| selected | Peer is selected for possible synchronization. |
| candidate | Peer is a candidate for selection. |
| sane | Peer passes basic sanity checks. |
| insane | Peer fails basic sanity checks. |
| valid | Peer time is believed to be valid. |
| invalid | Peer time is believed to be invalid. |
| leap_add | Peer is signaling that a leap second will be added. |

| Field | Descriptions |
|---|---|
| leap-sub | Peer is signaling that a leap second will be subtracted. |
| unsynced | Peer is not synchronized to any other machine. |
| ref ID | Address of machine peer is synchronized to. |
| time | Last timestamp peer received from its master. |
| our mode | Our mode relative to peer (active / passive / client / server / bdcast / bdcast client). |
| peer mode | Peer’s mode relative to us. |
| our poll ivl | Our poll interval to peer. |
| peer poll ivl | Peer’s poll interval to us. |
| root delay | Delay along path to root (ultimate stratum 1 time source). |
| root disp | Dispersion of path to root. |
| reach | Peer reachability (bit string in octal). |
| sync dist | Peer synchronization distance. |
| delay | Round trip delay to peer. |
| offset | Offset of peer clock relative to our clock. |
| dispersion | Dispersion of peer clock. |
| precision | Precision of peer clock in Hz. |
| version | NTP version number that peer is using. |
| org time | Originate time stamp. |
| rcv time | Receive time stamp. |
| xmt time | Transmit time stamp. |
| filtdelay | Round trip delay in milliseconds of each sample. |
| filtoffset | Clock offset in milliseconds of each sample. |
| filterror | Approximate error of each sample. |

# show ntp status

To show the status of Network Time Protocol (NTP), use the **show ntp status** EXEC command.

**show ntp status**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show ntp status** command:

```
Router# show ntp status

Clock is synchronized, stratum 4, reference is 131.108.13.57
nominal freq is 250.0000 Hz, actual freq is 249.9990 Hz, precision is 2**19
reference time is AFE2525E.70597B34 (00:10:22.438 PDT Mon Jul 5 1993)
clock offset is 7.33 msec, root delay is 133.36 msec
root dispersion is 126.28 msec, peer dispersion is 5.98 msec
```

Table 5-29 shows the significant fields in the display.

**Table 5-29 Show NTP Status Field Descriptions**

| Field | Description |
|---|---|
| synchronized | System is synchronized to an NTP peer. |
| unsynchronized | System is not synchronized to any NTP peer. |
| stratum | NTP stratum of this system. |
| reference | Address of peer we are synchronized to. |
| nominal freq | Nominal frequency of system hardware clock. |
| actual freq | Measured frequency of system hardware clock. |
| precision | Precision of this system’s clock (in Hz). |
| reference time | Reference timestamp. |
| clock offset | Offset of our clock to synchronized peer. |
| root delay | Total delay along path to root clock. |
| root dispersion | Dispersion of root path. |
| peer dispersion | Dispersion of synchronized peer. |

# show privilege

To display your current level of privilege, use the **show privilege** EXEC command.

**show privilege**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show privilege** command. The current privilege level is 15.

```
Router# show privilege

Current privilege level is 15
```

## Related Command

**enable password level**

# show processes

Use the **show processes** EXEC command to display information about the active processes.

**show processes** [**cpu**]

## Syntax Description

| | |
|---|---|
| **cpu** | (Optional) Displays detailed CPU utilization statistics. |

## Command Mode

EXEC

## Sample Displays

The following is sample output from the **show processes** command:

```
Router# show processes

CPU utilization for five seconds: 5%/5%; one minute: 4%; five minutes: 4%
 PID Q T       PC Runtime (ms)    Invoked   uSecs   Stacks  TTY Process
   1 L E     FCAC        28092       1396   20123  928/1000   0 Check heaps
   2 M E    304CE            0      83702       0  918/1000   0 Timers
   3 L E    538EE           92        323     284  778/1000   0 ARP Input
   4 M E    E11D2            0      83701       0  818/1000   0 SMT input
   5 M T    D0B3C           12        560      21  868/1000   0 ENVM Update
   6 L E    78EA0            0          1       0  924/1000   0 Probe Input
   7 M E    78A3E            4         40     100  952/1000   0 RARP Input
   8 H E    6BB88          220       1202     183 1830/2000   0 IP Input
   9 M E    8E962            0      16746       0  964/1000   0 TCP Timer
  10 L E    8FFEC            0          2       0  886/1000   0 TCP Protocols
  11 M E    75E72            4        143      27  820/1000   0 BOOTP Server
  12 M E     7582          140          8   17500  672/1000   0 Net Background
  13 L E    2BDD8           48        250     192  876/1000   0 Logger
  14 M *        0       145744       3307   44071 1420/2000   0 Exec
  15 M T    10816         1784      84843      21  780/1000   0 TTY Background
  16 H E     77EE            8        156      51  396/500    0 Net Input
  17 M T     74B8        11364       1415    8031  872/1000   0 Per-minute Jobs
  18 M E    D1DFE            0          1       0  974/1000   0 Crash writer
  19 H E   1EFF0A         3324      24309     136  602/1000   0 AT Input
  20 M E   1EEA60        41496      32350    1282  572/1000   0 AT RTMP
  21 L E   1F5F82           24        119     201  852/1000   0 AT NBP
  22 L E   201DF0       998592        189 5283555  516/1000   0 AT ZIP
  23 L E   1FED20        34460       8705    3958  574/1000   0 AT Maintenance
  24 M E   1F2C64           12        112     107  774/1000   0 AT ARP
  25 M E   224148           12      18321       0  590/1000   0 AT Domain
```

The following is sample output from the **show processes cpu** command:

```
Router# show processes cpu

CPU utilization for five seconds: 5%/2%; one minute: 3%; five minutes: 2%
  PID  Runtime (ms)    Invoked    uSecs    5Sec 1Min 5Min  Process
    1          1736         58    29931      0%   0%   0%  Check heaps
    2            68        585      116      1%   1%   0%  IP Input
    3             0        744        0      0%   0%   0%  TCP Timer
    4             0          2        0      0%   0%   0%  TCP Protocols
    5             0          1        0      0%   0%   0%  BOOTP Server
    6            16        130      123      0%   0%   0%  ARP Input
    7             0          1        0      0%   0%   0%  Probe Input
```

```
  8         0        7      0     0%   0%   0%  MOP Protocols
  9         0        2      0     0%   0%   0%  Timers
 10       692       64  10812     0%   0%   0%  Net Background
 11         0        5      0     0%   0%   0%  Logger
 12         0       38      0     0%   0%   0%  BGP Open
 13         0        1      0     0%   0%   0%  Net Input
 14       540     3466    155     0%   0%   0%  TTY Background
 15         0        1      0     0%   0%   0%  BGP I/O
 16      5100     1367   3730     0%   0%   0%  IGRP Router
 17        88     4232     20     2%   1%   0%  BGP Router
 18       152    14650     10     0%   0%   0%  BGP Scanner
 19       224       99   2262     0%   0%   1%  Exec
```

Table 5-30 describes significant fields shown in the two displays.

**Table 5-30 Show Processes Field Descriptions**

| Field | Description |
|---|---|
| PID | Process ID. |
| Q | Process queue priority. Possible values: H (high), M (medium), L (low). |
| T | Scheduler test. Possible values: E (event), T (time), S (suspended). |
| PC | Current program counter. |
| Runtime (ms) | CPU time the process has used, in milliseconds. |
| Invoked | Number of times the process has been invoked. |
| uSecs | Microseconds of CPU time for each process invocation. |
| Stacks | Low water mark/Total stack space available. |
| TTY | Terminal that controls the process. |
| Process | Name of process. |
| five seconds | CPU utilization by task in last 5 seconds. |
| one minute | CPU utilization by task in last minute. |
| five minutes | CPU utilization by task in last 5 minutes. |

Description of first line: CPU utilization for the last 5 seconds, 1 minute, and 5 minutes. The second part of the 5-second figure is the percentage of the CPU used by interrupt routines.

**Note** Because the network server has a 4-millisecond clock resolution, run times are considered reliable only after a large number of invocations or a reasonable, measured run time.

# show processes memory

Use the **show processes memory** EXEC command to show memory utilization.

**show processes memory**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show processes memory** command:

```
Router# show processes memory

Total: 2416588, Used: 530908, Free: 1885680
   PID   TTY  Allocated    Freed       Holding Process
   0     0    462708       2048        460660 *Init*
   0     0    76           4328 -      4252 *Sched*
   0     0    82732        33696       49036 *Dead*
   1     0    2616         0           2616 Net Background
   2     0    0            0           0 Logger
   21    0    20156        40          20116 IGRP Router
   4     0    104          0           104 BOOTP Server
   5     0    0            0           0 IP Input
   6     0    0            0           0 TCP Timer
   7     0    360          0           360 TCP Protocols
   8     0    0            0           0 ARP Input
   9     0    0            0           0 Probe Input
   10    0    0            0           0 MOP Protocols
   11    0    0            0           0 Timers
   12    0    0            0           0 Net Input
```

Table 5-31 describes significant fields shown in the display.

**Table 5-31 Show Processes Memory Field Descriptions**

| Field | Description |
|---|---|
| Total | Total amount of memory held. |
| PID | Process ID. |
| TTY | Terminal that controls the process. |
| Allocated | Sum of all memory that process has requested from the system. |
| Freed | How much memory a process has returned to the system. |
| Holding | Allocated memory minus freed memory. A value can be negative when it has freed more than it was allocated. |
| Process | Process name. |
| *Init* | System initialization. |
| *Sched* | The scheduler. |

| Field | Description |
|---|---|
| *Dead* | Processes as a group that are now dead. |

# show protocols

Use the **show protocols** EXEC command to display the configured protocols.

This command shows the global and interface-specific status of any configured Level 3 protocol; for example, IP, DECnet, IPX, AppleTalk, and so forth.

**show protocols**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show protocols** command:

```
Router# show protocols

Global values:
  Internet Protocol routing is enabled
  DECNET routing is enabled
  XNS routing is enabled
  Appletalk routing is enabled
  X.25 routing is enabled
Ethernet 0 is up, line protocol is up
  Internet address is 131.108.1.1, subnet mask is 255.255.255.0
  Decnet cost is 5
  XNS address is 2001.AA00.0400.06CC
  AppleTalk address is 4.129, zone Twilight
Serial 0 is up, line protocol is up
  Internet address is 192.31.7.49, subnet mask is 255.255.255.240
Ethernet 1 is up, line protocol is up
  Internet address is 131.108.2.1, subnet mask is 255.255.255.0
  Decnet cost is 5
  XNS address is 2002.AA00.0400.06CC
  AppleTalk address is 254.132, zone Twilight
Serial 1 is down, line protocol is down
  Internet address is 192.31.7.177, subnet mask is 255.255.255.240
  AppleTalk address is 999.1, zone Magnolia Estates
```

For more information on the parameters or protocols shown in this sample output, see the *Router Products Configuration Guide* publication.

# show queueing

To list the current state of the queue lists, use the **show queueing** privileged EXEC command.

**show queueing** [**custom** | **priority**]

## Syntax Description

| | |
|---|---|
| **custom** | (Optional) Shows status of custom queue lists. |
| **priority** | (Optional) Shows status of priority lists. |

## Command Mode

Privileged EXEC

## Usage Guidelines

If no keyword is entered, this command show the status of both custom and priority queue lists.

## Sample Display

The following is sample output from the **show queueing custom** EXEC command:

```
Router# show queueing custom
Current custom queue configuration:

List   Queue  Args
3      10     default
3      3      interface Tunnel3
3      3      protocol ip
3      3      byte-count 444 limit 3
```

## Related Commands

**custom-queue-listt**
**priority-group**
**priority-list interface**
**priority-list queue-limit**
**priority-list stun**
**queue-list default**
**queue-list interface**
**queue-list protocol**
**queue-list queue byte-count**
**queue-list queue limit**
**queue-list stun**

# show snmp

To check the status of communications between the SNMP agent and SNMP manager, use the **show snmp** EXEC command.

**show snmp**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

This command provides counter information for RFC 1213 SNMP operations. It also displays the chassis ID string defined with the **snmp-server chassis-id** command.

## Sample Display

The following is sample output from the **show snmp** command:

```
Router# show snmp
Chassis: SN#TS02K229
167 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    167 Number of requested variables
    0 Number of altered variables
    0 Get-request PDUs
    167 Get-next PDUs
    0 Set-request PDUs
167 SNMP packets output
    0 Too big errors (Maximum packet size 484)
    0 No such name errors
    0 Bad values errors
    0 General errors
    167 Get-response PDUs
    0 SNMP trap PDUs
```

## Related Command

**snmp-server chassis-id**

# show stacks

Use the **show stacks** EXEC command to monitor the stack utilization of processes and interrupt routines. Its display includes the reason for the last system reboot. If the system was reloaded because of a system failure, a saved system stack trace is displayed. This information is of use only to Cisco engineers analyzing crashes in the field. It is included here in case you need to read the displayed statistics to an engineer over the phone.

**show stacks**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show stacks** command following a system failure:

```
Router# show stacks

Minimum process stacks:
Free/Size  Name
 652/1000  Router Init
 726/1000  Init
 744/1000  BGP Open
 686/1200  Virtual Exec

Interrupt level stacks:
Level    Called Free/Size  Name
  1           0 1000/1000  env-flash
  3         738  900/1000  Multiport Communications Interfaces
  5         178  970/1000  Console UART
System was restarted by bus error at PC 0xAD1F4, address 0xD0D0D1A
GS Software (GS3), Version 9.1(0.16), BETA TEST SOFTWARE
Compiled Tue 11-Aug-92 13:27 by jthomas
Stack trace from system failure:
FP: 0x29C158, RA: 0xACFD4
FP: 0x29C184, RA: 0xAD20C
FP: 0x29C1B0, RA: 0xACFD4
FP: 0x29C1DC, RA: 0xAD304
FP: 0x29C1F8, RA: 0xAF774
FP: 0x29C214, RA: 0xAF83E
FP: 0x29C228, RA: 0x3E0CA
FP: 0x29C244, RA: 0x3BD3C
```

# snmp-server access-policy

To create or update an access policy, use the **snmp-server access-policy** global configuration command. To remove the specified access policy, use the **no** form of this command.

**snmp-server access-policy** *destination-party source-party context privileges*
**no snmp-server access-policy** *destination-party source-party context*

## Syntax Description

| | |
|---|---|
| *destination-party* | Name of a previously defined party identified as the destination party or target for this access policy. This name serves as a label used to reference a record defined for this party through the **snmp-server party** command. |
| *source-party* | Name of a previously defined party identified as the source party or subject for this access policy. This name serves as a label used to reference a record defined for this party through the **snmp-server party** command. |
| *context* | Name of a previously defined context that defines the resources for the access policy. This name serves as a label used to reference a record defined for this context through the **snmp-server contextt** command. |
| *privileges* | Bit mask representing the access privileges that govern the management operations that the source party can ask the destination party to perform. |

## Command Mode

Global configuration

## Usage Guidelines

An access policy defines the management operations the destination party can perform in relation to resources defined by the specified context when requested by the source party. A destination party performs management operations that are requested by a source party. A source party sends communications to a destination party requesting the destination party to perform management operations. A context identifies object resources accessible to a party.

Access policies are defined on the router for communications from the manager to the agent; in this case, the agent is the destination party and the manager is the source party. Access policies can also be defined on the router for Response message and trap message communication from the agent to the manager; in this case, the manager is the destination party and the agent is the source party.

The *privileges* argument specifies the types of SNMP operations that are allowed between the two parties. There are seven types of SNMP operations.You specify the privileges as a bit mask representing the access privileges that govern the management operations that the source party can ask the destination party to perform. In other words, the bit mask identifies the commands that the source party can send to the destination party.

You use decimal or hexadecimal format to specify privileges as a sum of values in which each value specifies an SNMP PDU type that the source party can use to request an operation. The decimal values are defined as follows:

- Get =1
- GetNext = 2
- Response = 4
- Set = 8
- SNMPv1-Trap = 16
- GetBulk = 32
- SNMPv2-Trap = 128

To remove an access-policy entry, all three arguments specified as command arguments must match exactly the values of the entry to be deleted. A difference of one value constitutes a different access policy.

The first **snmp-server** command that you enter enables both versions of SNMP.

## Examples

The following example configures an access policy providing the manager with read-only access to the agent:

```
snmp-server access-policy agt1 mgr1 ctx1 0x23
```

The following example configures an access policy providing the manager with read-write access to the agent:

```
snmp-server access-policy agt2 mgr2 ctx2 43
```

The following example configures an access policy that allows responses and SNMP v.2 traps to be sent from the agent to a management station:

```
snmp-server access-policy mgr1 agt1 ctx1 132
```

The following example removes the access policy configured for the destination party named *agt1*, the source party named *mgr1*, and with a context named *ctx1*.

```
no snmp-server access-policy agt1 mgr1 ctx1
```

## Related Commands

**snmp-server context**
**snmp-server party**

# snmp-server chassis-id

To provide a message line identifying the SNMP server serial number, use the **snmp-server chassis-id** global configuration command. Use the **no** form of this command to restore the default value, if any.

**snmp-server chassis-id** *text*
**no snmp-server chassis-id**

## Syntax Description

| | |
|---|---|
| *text* | Message you want to enter to identify the chassis serial number. |

## Default

On hardware platforms where the serial number can be machine read, the default is the serial number. For example, an AGS+ does not have a default value; a Cisco 7000 has a default value of its serial number.

## Command Mode

Global configuration

## Usage Guidelines

The Cisco MIB provides a chassis MIB variable that enables the SNMP manager to gather data on system card descriptions, chassis type, chassis hardware version, chassis ID string, software version of ROM monitor, software version of system image in ROM, bytes of processor RAM installed, bytes of NVRAM installed, bytes of NVRAM in use, current configuration register setting, and the value of the configuration register at the next reload. The following installed card information is provided: type of card, serial number, hardware version, software version, and chassis slot number.

The chassis ID message can be seen with **show snmp** command.

## Example

In the following example, the chassis serial number specified is 1234456:

```
snmp-server chassis-id 1234456
```

## Related Command

**show snmp**

# snmp-server community

To set up the community access string to permit access to the SNMPv1 protocol, use the **snmp-server community** global configuration command. The **no** form of this command removes the specified community string.

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

## Syntax Description

| | |
|---|---|
| *string* | Community string that acts like a password and permits access to the SNMP protocol. |
| **RO** | (Optional) Specifies read-only access. Authorized management stations are only able to retrieve MIB objects. |
| **RW** | (Optional) Specifies read-write access. Authorized management stations are able to both retrieve and modify MIB objects. |
| *number* | (Optional) Integer from 1 to 99 that specifies an access list of IP addresses that are allowed to use the community string to gain access to the SNMP v.1 agent. |

## Default

By default, an SNMP community string permits read-only access.

## Command Mode

Global configuration

## Usage Guidelines

For the previous version of this command, the *string* argument was optional. The *string* argument is now required. However, to prevent errors and provide backward-compatibility, if the string option is omitted, a default value of public is assumed.

The **no snmp-server** command disables both versions of SNMP (SNMPv1 and SNMPv2).

The first **snmp-server** command that you enter enables both versions of SNMP.

## Examples

The following example assigns the string *comaccess* to SNMPv1 allowing read-only access and specifies that IP access list 4 can use the community string:

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

The following example disables both versions of SNMP:

```
no snmp-server
```

## Related Command

**snmp-server party**

# snmp-server contact

To set the system contact (syscontact) string, use the **snmp-server contact** global configuration command. Use the **no** form to remove the system contact information.

**snmp-server contact** *text*
**no snmp-server contact**

## Syntax Description

| | |
|---|---|
| *text* | String that describes the system contact information. |

## Default

No syscontact string is set.

## Command Mode

Global configuration

## Example

The following is an example of a syscontact string:

```
snmp-server contact Dial System Operator at beeper # 27345
```

# snmp-server context

To create or update a context record, use the **snmp-server context** global configuration command. To remove a specific context entry, use the **no** form of this command.

**snmp-server context** *context-name context-oid view-name*
**no snmp-server context** *context-name*

## Syntax Description

| | |
|---|---|
| *context-name* | Name of the context to be created or updated. This name serves as a label used to reference a record for this context. |
| *context-oid* | Object identifier to assign to the context. Specify this value in dotted decimal notation, with an optional text identifier; for example, 1.3.6.1.6.3.3.1.4.131.108.45.11.1(== initialContextId.131.108.45.11.1). |
| *view-name* | Name of a previously defined view. The view defines the objects available to the context. |

## Command Mode

Global configuration

## Usage Guidelines

A context record identifies object resources accessible to a party. A context record is one of the components that make up an access policy. Therefore, you must configure a context record before you can create an access policy that includes the context. Context records and party records further codify MIB views.

To remove a context entry, specify only the name of the context. The name identifies the context to be deleted.

The first **snmp-server** command that you enter enables both versions of SNMP.

## Example

The following example shows how to create a context that includes all objects in the MIB-II subtree using a previously defined view named *mib2*:

```
snmp-server context mycontext initialContextid.131.108.24.56.3 mib2
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**snmp-server view**
**write memory** †
**write terminal** †

# snmp-server host

To specify the recipient of an SNMP trap operation, use the **snmp-server host** global configuration command. The **no** form of this command removes the specified host.

**snmp-server host** *host community-string* [**envmon**] [**framerelay**] [**sdlc**] [**snmp**] [**tty**] [**x25**]
**no snmp-server host** *host community-string* [**envmon**] [**framerelay**] [**sdlc**] [**snmp**] [**tty**] [**x25**]

## Syntax Description

| | |
|---|---|
| *host* | Name or Internet address of the host. |
| *community-string* | Password-like community string to send with the trap operation. |
| **envmon** | (Optional) Enables Cisco enterprise-specific environmental monitor traps to be sent to the trap receiver *host* when an environmental threshold has been exceeded. |
| **framerelay** | (Optional) Enables Frame Relay traps to be sent to the trap receiver *host*. |
| **sdlc** | (Optional) Enables SDLC traps to be sent to the trap receiver *host*. |
| **snmp** | (Optional) Enables the SNMP traps defined in RFC 1157. |
| **tty** | (Optional) Enables Cisco enterprise-specific traps when a TCP connection closes. |
| **x25** | (Optional) Enable X.25 event traps to be sent to *host*. |

## Default

No traps are sent.

If you enter the command with no keywords, the default is to enable all trap types.

## Command Mode

Global configuration

## Usage Guidelines

The **snmp-server host** command specifies which host or hosts should receive SNMP traps. You need to issue the **snmp-server host** command once for each host acting as a trap recipient. When multiple **snmp-server host** commands are given, the community string in the last command is used, and in general, the trap types set in the last command will be used for all SNMP trap operations.

## Examples

The following example sends the SNMP traps defined in RFC 1157 to the host specified by the name cisco.com. The community string is defined as the string *comaccess*.

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

The following example sends the SNMP and Cisco enterprise-specific traps to address 131.108.2.160:

```
snmp-server host 131.108.2.160
```

## Related Command

**snmp-server trap-timeout**

# snmp-server location

To set the system location string, use the **snmp-server location** global configuration command. Use the **no** form of this command to remove the location string.

**snmp-server location** *text*
**no snmp-server location**

## Syntax Description

| | |
|---|---|
| *text* | String that describes the system location information. |

## Default

No system location string is set.

## Command Mode

Global configuration

## Example

The following example illustrates a system location string:

```
snmp-server location Building 3/Room 214
```

# snmp-server packetsize

To establish control over the largest SNMP packet size permitted when the SNMP server is receiving a request or generating a reply, use the **snmp-server packetsize** global configuration command. Use the **no** form of this command to restore the default value.

**snmp-server packetsize** *byte-count*
**no snmp-server packetsize**

## Syntax Description

| | |
|---|---|
| *byte-count* | Integer byte count from 484 to 8192. |

## Default

484 bytes

## Command Mode

Global configuration

## Example

The following example establishes a packet filtering of a maximum size of 1024 bytes:

```
snmp-server packetsize 1024
```

# snmp-server party

To create or update a party record, use the **snmp-server party** global configuration command. To remove a specific party entry, use the **no** form of this command.

**snmp-server party** *party-name party-oid* [*protocol-address*] [**packetsize** *size*]
[**local** | **remote**] [**authentication** {**md5** *key* [**clock** *clock*]
[**lifetime** *lifetime*] | **snmpv1** *string*}]
**no snmp-server party** *party-name*

## Syntax Description

| | |
|---|---|
| *party-name* | Name of the party characterized by the contents of the record. This name serves as a label used to reference the party record that you are creating or modifying. |
| *party-oid* | Object identifier to assign to the party. Specify this value in dotted decimal notation, with an optional text identifier; for example, 1.3.6.1.6.3.3.1.3.131.108.34.54.1 (= initialPartyId.131.108.34.54.1) |
| *protocol-address* | (Optional) Address of the protocol that the party record pertains to. Currently the only supported protocol is UDP, so this value specifies a UDP address in the format *a.b.c.d port*.<br>In future releases, additional protocols will be supported.<br>This value is used to specify the destination of trap messages. |
| **packetsize** *size* | (Optional) Maximum size in bytes of a message that this party is able to receive. By default, the packet size set through the **snmp-server packetsize** command is used. |
| **local** \| **remote** | (Optional) Indicates that the party is local or remote. If neither **local** nor **remote** is specified, a default value of local is assumed. |
| **authentication** | (Optional) Indicates that the party uses an authentication protocol. If specified, either **md5** or **snmpv1** is required. |
| **md5** *key* | (Optional) Indicates that the party uses the Message Digest algorithm MD5 for message authentication. If **md5** is specified, you must also specify a 16-byte hexadecimal ASCII string representing the MD5 authentication key for the party. All messages sent to this party will be authenticated using the SNMP v2 MD5 authentication method with the key specified by *key*. |
| **clock** *clock* | (Optional) Initial value of the authentication clock. |
| **lifetime** *lifetime* | (Optional) Lifetime, in seconds, that represents the upper bound on acceptable delivery delay for messages generated by the party. |

| | |
|---|---|
| **snmpv1** *string* | (Optional) Community string. The keyword **snmpv1** indicates that the party uses community-based authentication. All messages sent to this party will be authenticated using the SNMP v1community string specified by *string* instead of MD5. |

## Defaults

If neither **local** nor **remote** is specified to indicate the location of the party, the party is assumed to be local.

If you do not specify a packet size, the packet size set through the**snmp-server packetsize** command is used.

## Command Mode

Global configuration

## Usage Guidelines

You define parties to identify managers and agents. An SNMP v2 party identity is unique; it includes the logical network location of the party, characterized by the transport protocol domain and transport addressing information, and, optionally, an authentication method and its arguments.The authentication protocol reliably identifies the origin of all messages sent by the party. The authentication protocol also ensures the integrity of the messages; in other words, it ensures that the message received is the message that was sent.

Specifying **md5** as the authentication method implies that this party record pertains to an SNMPv2 party.

Specifying **snmpv1** as the authentication method implies that this party record pertains to an SNMPv1 party. This allows a management station that supports only SNMPv1 to use SNMPv2 MIB views. Instead of using the **snmp-server community** command, you can use the **snmp-server party** command with the **snmpv1** keyword to define an SNMP v.1 party to be used to communicate with an SNMP v.1 management station. The **snmp-server community** command does not allow you to create MIB views for an SNMP v.1 management station.

If authentication is not specified, the party record pertains to an SNMPv2 party, and no authentication will be performed for messages sent to this party.

To remove a party record, specify only the name of the party. The name identifies the party to be deleted.

The first **snmp-server** command that you enter enables both versions of SNMP.

## Examples

The following example configures a remote unauthenticated party:

```
snmp-server party mgr1 initialPartyId.131.108.45.32.3 udp 131.108.45.76 162
```

The following example configures a local MD5-authenticated party with a large maximum packet size. You enter this command as a single line:

```
snmp-server party agt1 initialPartyId.131.108.45.32.4 packetsize 1500 local
 authentication md5 23de457623900ac3ef568fcb236589 lifetime 400
```

The following example configures an SNMP v.1 proxy party for the community *public*:

```
snmp-server party proxyv1 initialPartyId.131.108.45.32.100 authentication snmpv1 public
```

The following example removes the party named *mgr1*:

```
no snmp-server party mgr1
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**snmp-server community**
**write memory** †
**write terminal** †

# snmp-server queue-length

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

**snmp-server queue-length** *length*

## Syntax Description

| | |
|---|---|
| *length* | Integer that specifies the number of trap events that can be held before the queue must be emptied. |

## Default

10 events

## Command Mode

Global configuration

## Usage Guidelines

This command defines the length of the message queue for each trap host. Once a trap message is successfully transmitted, software will continue to empty the queue, but never faster than at a rate of four trap messages per second.

## Example

The following example establishes a message queue that traps four events before it must be emptied:

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

# snmp-server system-shutdown

To use the SNMP message reload feature, the device configuration must include the **snmp-server system-shutdown** global configuration command. The **no** form of this command prevents an SNMP system-shutdown request (from an SNMP manager) from resetting the Cisco agent.

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

## Syntax Description

This command has no arguments or keywords.

## Default

This command is not included in the configuration file.

## Command Mode

Global configuration

## Example

The following example illustrates how to include the SNMP message reload feature in the device configuration:

```
snmp-server system-shutdown
```

# snmp-server trap-authentication

To establish trap message authentication, use the **snmp-server trap-authentication** global configuration command. To remove message authentication, use the **no** form of this command.

**snmp-server trap-authentication** [**snmpv1** | **snmpv2**]
**no snmp-server trap-authentication** [**snmp1** | **snmp2**]

## Syntax Description

| | |
|---|---|
| **snmpv1** | (Optional) Indicates that SNMP authentication traps will be sent to SNMPv1 management stations only. |
| **snmpv2** | (Optional) Indicates that SNMP authentication traps will be sent to SNMPv2 management stations only. |

## Defaults

Specifying the **snmp-server trap-authentication** command without a keyword turns on trap message authentication. In this case, messages are sent to the host that is specified though the **snmp-server host** command and to any SNMP stations configured through access policies to receive trap messages.

## Command Mode

Global configuration

## Usage Guidelines

Specify the **snmpv1** or **snmpv2** keyword to indicate the type of management stations to send the trap messages to.

This command enables the router as an agent to send a trap message when it receives an SNMPv1 packet with an incorrect community string or an SNMPv2 packet with an incorrect MD5 authentication key.

The SNMP specification requires that a trap message be generated for each packet with an incorrect community string or authentication key; however, because this action can result in a security breach, the router (as an agent) by default does not send a trap message when it receives an incorrect community string or authentication key.

The community string or key is checked before any access list that may be set, so it is possible to get spurious trap messages. In other words, if you have issued an snmp-server community command with a specified access list, you might receive messages that come from someone that is not on the access list; in this case, an authentication trap is issued.The only workarounds are to disable trap authentication or to configure an access list on a router between the SNMP agent and the SNMP manager to prevent packets from getting to the SNMP agent.

To turn off all message authentication traps, use the **no snmp-server trap-authentication** without a keyword. To turn off message authentication traps only for SNMPv1 stations or only for SNMPv2 stations, give the negative form of the command with the appropriate keyword.

The first **snmp-server** command that you enter enables both versions of SNMP.

## Example

The following example illustrates how to enter the command that establishes trap message authentication:

```
snmp-server trap-authentication
```

## Related Command

**snmp-server host**

# snmp-server trap-source

To specify the interface (and hence the corresponding IP address) that an SNMP trap should originate from, use the **snmp-server trap-source** global configuration command. Use the **no** form of the command to remove the source designation.

**snmp-server trap-source** *interface*
**no snmp-server trap-source**

## Syntax Description

| | |
|---|---|
| *interface* | Interface from which the SNMP trap originates. The argument includes the interface type and number in platform-specific syntax. |

## Default

No interface is specified.

## Command Mode

Global configuration

## Usage Guidelines

When an SNMP trap is sent from a Cisco SNMP server, it has a trap address of whatever interface it happened to go out of at that time. Use this command if you want to use the trap address to trace particular needs.

## Examples

The following example specifies that the IP address for interface Ethernet 0 is the source for all traps on the router:

```
snmp-server trap-source ethernet 0
```

The following example specifies that the IP address for interface Ethernet 2/1 on a Cisco 7000 is the source for all traps on the router:

```
snmp-server trap-source ethernet 2/1
```

# snmp-server trap-timeout

To define how often to try resending trap messages on the retransmission queue, use the **snmp-server trap-timeout** global configuration command.

**snmp-server trap-timeout** *seconds*

## Syntax Description

| | |
|---|---|
| *seconds* | Integer that sets the interval, in seconds, for resending the messages |

## Default

30 seconds

## Command Mode

Global configuration

## Usage Guidelines

Before the router tries to send a trap, it looks for a route to the destination address. If there is no known route, the trap is saved in a retransmission queue. The **server trap-timeout** command determines the number of seconds between retransmission attempts.

## Example

The following example sets an interval of 20 seconds to try resending trap messages on the retransmission queue:

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

## Related Command

**snmp-server host**

# snmp-server userid

To create or update an SNMP v.2 security context using the simplified security conventions method, use the **snmp-server userid** global configuration command. The **no** form of this command removes the specified security context.

**snmp-server userid** *user-id* [**view** *view-name*] [**RO** | **RW**] [**password** *password*]
**no snmp-server userid** *user-id*

## Syntax Description

| | |
|---|---|
| *user-id* | User ID name that identifies an approved SNMP v.2 user. The user ID represents a set of security information for this user. This value can identify a particular user of the system or a background process. |
| **view** *view-name* | (Optional) View to be used for this security context. The argument *view-name* must be the name of a predefined view. For authenticated users, defaults to the predefined view *everything*. For users who are not authenticated, defaults to the predefined view *restricted*. |
| **RO** | (Optional) Specifies read-only access.This is the default for unauthenticated users. |
| **RW** | (Optional) Specifies read-write access. This is the default for authenticated users. |
| **password** *password* | (Optional) Indicates that this is an authenticated user, and defines the password used to authenticate the user. The password must be at least eight characters long. |

## Defaults

For the **snmp-server userid** command, the default value for the *view-name* argument depends on whether the security context is password protected. Depending on whether the security context is password protected, one of the following default values applies:

- If the security context is password protected (meaning the user is authenticated), the default value for *view-name* is *everything*. *Everything* is a predefined value indicating that the user can see all objects.
- If the security context is not password protected (meaning that the user is not authenticated), the default value for *view-name* is *restricted*. *Restricted* is a predefined value indicating that the user can see three groups: system, snmpStats, and snmpParties.

These predefined views are described in RFC 1447.

Read-only access is the default for unauthenticated users.

Read-write access is the default for authenticated users.

## Command Mode

Global configuration

## Usage Guidelines

The **snmp-server userid** command implements the *simplified security conventions* method of configuring the relationship between an agent and a manager. It provides a single-step method that offers an alternative to the access policy configuration method of defining this relationship. The simplified method offers ease-of-use at the cost of forfeiting control over certain values that can be configured if you create an access policy. The simplified security conventions method applies to a configuration in which the agent is the destination or recipient of messages and the manager is the source or sender of messages. You cannot use this command to define a relationship in which the agent is the source and the manager is the destination. The security context created does not apply to trap messages.

**Caution** Use the simplified security conventions method only if the management station participating in the manager-agent relationship also supports this method.

If you provide a password, the password is encrypted on write operations for which encryption is enabled.

If you use the **snmp-server userid** command, the SNMPv2 implementation assumes default values that it determines internally for required information that you cannot provide through the command interface. SNMPv2 uses the following methods to determine these values:

- To create the context, it constructs the object identifier assigned to the context from the agent's IP address and the user ID name supplied as an argument to the **snmp-server userid** command.
- To create a party record for the agent, it constructs the object identifier assigned to the party from the agent's IP address and the *user-id* supplied as an argument to the **snmp-server userid** command. It assumes that the agent is **local**. If the user is authenticated—indicated by a password argument supplied on the **snmp-server userid** command—it constructs an MD5 key from the password.
- To create a party record for the manager, it constructs the object identifier from the agent's address and the *user-id* supplied as an argument to the **snmp-server userid** command. It assumes that the agent is **remote**. If the user is authenticated—indicated by a password argument supplied on the **snmp-server userid** command—it constructs an MD5 key from the password.
- To define the privileges, it sets a bit-mask value based on whether the user has read-only (**RO**) or read-write (**RW**) access, as specified on the **snmp-server userid** command. The SNMP v.2 implementation assumes the following default values:
    - — For read-only access, it sets the bit mask to 0x23; this means that the source party can send the Get, GetNext, and GetBulk commands to the destination party.
    - — For read-write access, it sets the bit mask to 0x2B; this means that the source party can send the Get, GetNext, GetBulk, and Set commands to the destination party.

The first **snmp-server** command that you enter enables both versions of SNMP.

## Example

The following example configures a security context for the user *harold*, who is unauthenticated, uses the view *default*, and has read-only access:

```
snmp-server userid harold
```

## Related Commands

**snmp-server access-policy**
**snmp-server contextt**
**snmp-server party**
**snmp-server view**

# snmp-server view

To create or update a view entry, use the **snmp-server view** global configuration command. To remove the specified SNMP server view entry, use the **no** form of this command.

**snmp-server view** *view-name oid-tree* {**included** | **excluded**}
**no snmp-server view** *view-name*

## Syntax Description

| | |
|---|---|
| *view-name* | Label for the view record that you are updating or creating. The name is used to reference the record. |
| *oid-tree* | Object identifier of the ASN.1 subtree to be included or excluded from the view. To identify the subtree, specify a text string consisting of numbers, such as *1.3.6.2.4*, or a word, such as *system*. Replace a single subidentifier with the asterisk (*) wildcard to specify a subtree family; for example 1.3.*.4. |
| **included** \| **excluded** | Type of view. You must specify either **included** or **excluded**. |

## Command Mode

Global configuration

## Usage Guidelines

Other SNMPv2 commands require a view as an argument. You use this command to create a view to be used as arguments for other commands that create records including a view.

Two standard predefined views can be used when a view is required, instead of defining a view. One is *everything*, which indicates that the user can see all objects. The other is *restricted*, which indicates that the user can see three groups: system, snmpStats, and snmpParties. The predefined views are described in RFC 1447.

The first **snmp-server** command that you enter enables both versions of SNMP.

## Examples

The following example creates a view that includes all objects in the MIB-II subtree:

```
snmp-server view mib2 mib-2 included
```

The following example creates a view that includes all objects in the MIB-II system group and all objects in the Cisco enterprise MIB:

```
snmp-server phred system included
snmp-server view phred cisco included
```

The following example creates a view that includes all objects in the MIB-II system group except for sysServices (System 7) and all objects for interface 1 in the MIB-II interfaces group:

```
snmp-server view agon system included
snmp-server view agon system.7 excluded
snmp-server view agon ifEntry.*.1 included
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**snmp-server context**
**snmp-server userid**
**write memory** †
**write terminall** †

# tacacs-server attempts

To control the number of login attempts that can be made on a line set up for TACACS verification, use the **tacacs-server attempts** global configuration command. Use the **no tacacs-server attempts** command to remove this feature and restore the default.

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

## Syntax Description

| | |
|---|---|
| *count* | Integer that sets the number of attempts. |

## Default

Three attempts

## Command Mode

Global configuration

## Example

The following example changes the login attempt to just one try:

```
tacacs-server attempts 1
```

# tacacs-server authenticate

To specify that the network or router must indicate whether the user may perform an action when the user attempts to perform the action, use the **tacacs-server authenticate** global configuration command.

**tacacs-server authenticate** {**connection** [**always**] | **enable** | **slip** [**always**] [**access-lists**]}

## Syntax Description

| | |
|---|---|
| **connection** | Configures a required response when a user makes a TCP connection. |
| **enable** | Configures a required response when a user enters the **enable** command. |
| **slip** | Configures a required response when a user starts a SLIP or PPP session. |
| **always** | (Optional) Performs authentication even when a user is not logged in. This option only applies to the **connection** or **slip** keywords. |
| **access-lists** | (Optional) Requests and installs access lists. This option only applies to the **slip** keyword. |

## Command Mode

Global configuration

## Usage Guidelines

Enter one of the keywords to specify the action (when a user makes a TCP connection, for example).

**Note** Before you use the **tacacs-server authenticate** command, you must enable the **tacacs-server extended** command.

**Note** This command is not used in AAA/TACACS+ and has been replaced by the **aaa authorization** command.

## Example

The following example configures TACACS logins that authenticate user TCP connections:

```
tacacs-server authenticate connect
```

## Related Command

**enable use-tacacss**

# tacacs-server extended

To enable an extended TACACS mode, use the **tacacs-server extended** global configuration command. Use the **no** form of this command to disable the mode.

**tacacs-server extended**
**no tacacs-server extended**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Global configuration

## Usage Guidelines

---

**Note** This command initializes extended TACACS. To initialize AAA/TACACS+, use the **aaa new-model** command.

---

## Example

The following example enables extended TACACS mode:

```
tacacs-server extended
```

# tacacs-server host

To specify a TACACS host, use the **tacacs-server host** global configuration command. You can use multiple **tacacs-server host** commands to specify multiple hosts. The software searches for the hosts in the order you specify them. The **no** form of this command deletes the specified name or address.

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

## Syntax Description

| | |
|---|---|
| *name* | Name or IP address of the host. |

## Default

No TACACS host is specified.

## Command Mode

Global configuration

## Example

The following example specifies a TACACS host named SCACAT:

```
tacacs-server host SCACAT
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**login tacacs** †
**ppp** †
**slip** †

# tacacs-server key

Use the **tacacs-server key** command to set the authentication/encryption key used for all TACACS+ communications between the access server and the TACACS+ daemon. To disable the key, use the **no** form of the command.

**tacacs-server key** *key*
**no tacacs-server key** [*key*]

## Syntax Description

| | |
|---|---|
| *key* | The key used to set authentication and encryption. This key must match the key used on the TACACS+ daemon. |

## Command Mode

Global Configuration

## Usage Guidelines

After enabling AAA with the **aaa new-model** command, you must set the authentication and encryption key using the **tacas-server key** command.

The key entered must match the key used on the TACACS+ daemon. All leading spaces are ignored, spaces within and at the end of the key are not. If you use spaces in your key, do not enclose the key in double quotes unless the quotes themselves are part of the key.

## Example

The following example illustrates how to set the authentication and encryption key to 'dare to go':

```
tacacs-server key dare to go
```

## Related Command

**aaa new-model**

# tacacs-server last-resort

To cause the network server to request the privileged password as verification, or to force successful login without further input from the user, use the **tacacs-server last-resort** global configuration command. The **no tacacs-server last-resort** command restores the system to the default behavior.

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

## Syntax Description

| | |
|---|---|
| **password** | Allows the user to access the EXEC command mode by entering the password set by the **enable** command. |
| **succeed** | Allows the user to access the EXEC command mode without further question. |

## Default

If, when running the TACACS server, the TACACS server does not respond, the default action is to deny the request.

## Command Mode

Global configuration

## Usage Guidelines

Use the **tacacs-server last-resort** command to be sure that login can occur; for example, when a systems administrator needs to log in to troubleshoot TACACS servers that might be down.

**Note** This command is not used in AAA/TACACS+.

## Example

The following example forces successful login:

```
tacacs-server last-resort succeed
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**enable password**
**login** (EXEC) †

# tacacs-server notify

Use the **tacacs-server notify** global configuration command to cause a message to be transmitted to the TACACS server, with retransmission being performed by a background process for up to 5 minutes.

**tacacs-server notify** {**connection** [**always**] | **enable** | **logout** [**always**] | **slip** [**always**]}

## Syntax Description

| | |
|---|---|
| **connection** | Specifies that a message be transmitted when a user makes a TCP connection. |
| **always** | (Optional) Sends a message even when a user is not logged in. This option applies only to SLIP or PPP sessions and can be used with the **connection**, **logout**, or **slip** keywords. |
| **enable** | Specifies that a message be transmitted when a user enters the **enable** command. |
| **logout** | Specifies that a message be transmitted when a user logs out. |
| **slip** | Specifies that a message be transmitted when a user starts a SLIP or PPP session. |

## Default

No message is transmitted to the TACACS server.

## Command Mode

Global configuration

## Usage Guidelines

The terminal user receives an immediate response allowing access to the feature specified. Enter one of the keywords to specify notification of the TACACS server upon the corresponding action (when user logs out, for example).

**Note** This command is not used in AAA/TACACS+ and has been replaced by the **aaa accounting** suite of commands.

## Example

The following example sets up notification of the TACACS server when a user logs out:

```
tacacs-server notify logout
```

# tacacs-server optional-passwords

To specify that the first TACACS request to a TACACS server be made *without* password verification, use the **tacacs-server optional-passwords** global configuration command. Use the **no** form of this command to restore the default.

**tacacs-server optional-passwords**
**no tacacs-server optional-passwords**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Global configuration

## Usage Guidelines

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

**Note** This command is not used by AAA/TACACS+.

## Example

The following example configures the first login to not require TACACS verification:

```
tacacs-server optional-passwords
```

# tacacs-server retransmit

To specify the number of times the router software will search the list of TACACS server hosts before giving up, use the **tacacs-server retransmit** global configuration command. The router software will try all servers, allowing each one to timeout before increasing the retransmit count. The **no** form of this command restores the default.

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

## Syntax Description

| | |
|---|---|
| *retries* | Integer that specifies the retransmit count. |

## Default

Two retries

## Command Mode

Global configuration

## Example

The following example specifies a retransmit counter value of five times:

```
tacacs-server retransmit 5
```

# tacacs-server timeout

To set the interval that the server waits for a server host to reply, use the **tacacs-server timeout** global configuration command. The **no** form of this command restores the default.

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

## Syntax Description

| | |
|---|---|
| *seconds* | Integer that specifies the timeout interval in seconds. |

## Default

5 seconds

## Command Mode

Global configuration

## Example

The following example changes the interval timer to 10 seconds:

```
tacacs-server timeout 10
```

# test flash

To test Flash memory on MCI and envm Flash EPROM interfaces, use the **test flash** EXEC command.

**test flash**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Example

The following example illustrates how to begin the interface test:

```
test flash
```

# test interfaces

To test the system interfaces on the modular router, use the **test interfaces** EXEC command.

**test interfaces**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

The **test interfaces** EXEC command is intended for the factory checkout of network interfaces. It is not intended for diagnosing problems with an operational router. The **test interfaces** output does not report correct results if the router is attached to a "live" network. For each network interface that has an IP address that can be tested in loopback (MCI and ciscoBus Ethernet and all serial interfaces), the **test interfaces** command sends a series of ICMP echoes. Error counters are examined to determine the operational status of the interface.

## Example

The following example illustrates how to begin the interface test:

```
test interfaces
```

# test memory

To perform a test of Multibus memory (including nonvolatile memory) on the modular router, use the **test memory** EXEC command.

**test memory**

**Caution** The memory test overwrites memory. If you use the **test memory** command, you will need to rewrite nonvolatile memory. For example, if you test Multibus memory, which is the memory used by the CSC-R 4-Mbps Token Ring interfaces, you will need to reload the system before the network interfaces will operate properly. The **test memory** command is intended primarily for use by Cisco personnel.

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

## Example

The following example illustrates how to begin the memory test:

```
test memory
```

# trace (privileged)

Use the **trace** EXEC command to discover the routes the router's packets will actually take when traveling to their destination.

**trace** [*protocol*] [*destination*]

## Syntax Description

| | |
|---|---|
| **protocol** | (Optional) Protocols that can be used are **appletalk**, **clns**, **ip** and **vines**. |
| *destination* | (Optional) Destination address or host name on the command line. The default parameters for the appropriate protocol are assumed and the tracing action begins. |

## Default

*protocol* is based on the router's examination of the format of *destination*. For example, if the router finds a *destination* in IP format, the *protocol* defaults to **ip**.

## Command Mode

Privileged EXEC

## Usage Guidelines

The **trace** command works by taking advantage of the error messages generated by routers when a datagram exceeds its time-to-live (TTL) value.

The **trace** command starts by sending probe datagrams with a TTL value of one. This causes the first router to discard the probe datagram and send back an error message. The **trace** command sends several probes at each TTL level and displays the round-trip time for each.

The **trace** command sends out one probe at a time. Each outgoing packet may result in one or two error messages. A *time exceeded* error message indicates that an intermediate router has seen and discarded the probe. A *destination unreachable* error message indicates that the destination node has received the probe and discarded it because it could not deliver the packet. If the timer goes off before a response comes in, **trace** prints an asterisk (*).

The **trace** command terminates when the destination responds, when the maximum TTL is exceeded, or when the user interrupts the trace with the escape sequence. By default, to invoke the escape sequence, press Ctrl-^ X—which is done by simultaneously pressing the Ctrl, Shift, and 6 keys, letting go, then pressing the X key.

To use nondefault parameters and invoke an extended **trace** test, enter the command without a *destination* argument. You will be stepped through a dialog to select the desired parameters.

## Common Trace Problems

Due to bugs in the IP implementation of various hosts and routers, the IP **trace** command may behave in odd ways.

Not all destinations will respond correctly to a probe message by sending back an *ICMP port unreachable* message. A long sequence of TTL levels with only asterisks, terminating only when the maximum TTL has been reached, may indicate this problem.

There is a known problem with the way some hosts handle an *ICMP TTL exceeded* message. Some hosts generate an *ICMP* message but they reuse the TTL of the incoming packet. Since this is zero, the ICMP packets do not make it back. When you trace the path to such a host, you may see a set of TTL values with asterisks (*). Eventually the TTL gets high enough that the *ICMP* message can get back. For example, if the host is six hops away, **trace** will time out on responses 6 through 11.

## Sample Display Showing Trace IP Routes

The following display shows sample IP **trace** output when a destination host name has been specified:

```
Router# trace ABA.NYC.mil

Type escape sequence to abort.
Tracing the route to ABA.NYC.mil (26.0.0.73)
  1 DEBRIS.CISCO.COM (131.108.1.6) 1000 msec 8 msec 4 msec
  2 BARRNET-GW.CISCO.COM (131.108.16.2) 8 msec 8 msec 8 msec
  3 EXTERNAL-A-GATEWAY.STANFORD.EDU (192.42.110.225) 8 msec 4 msec 4 msec
  4 BB2.SU.BARRNET.NET (131.119.254.6) 8 msec 8 msec 8 msec
  5 SU.ARC.BARRNET.NET (131.119.3.8) 12 msec 12 msec 8 msec
  6 MOFFETT-FLD-MB.in.MIL (192.52.195.1) 216 msec 120 msec 132 msec
  7 ABA.NYC.mil (26.0.0.73) 412 msec 628 msec 664 msec
```

Table 5-32 describes the fields shown in the display.

**Table 5-32 Trace Field Descriptions**

| Field | Description |
|---|---|
| 1 | Indicates the sequence number of the router in the path to the host. |
| DEBRIS.CISCO.COM | Host name of this router. |
| 131.108.1.6 | Internet address of this router. |
| 1000 msec 8 msec 4 msec | Round-trip time for each of the three probes that are sent. |

## Sample Display Showing Extended IP Trace Dialog

The following display shows a sample **trace** session involving the extended dialog of the **trace** command.

```
Router# trace

Protocol [ip]:
Target IP address: mit.edu
Source address:
Numeric display [n]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Type escape sequence to abort.
Tracing the route to MIT.EDU (18.72.2.1)
```

```
 1 ICM-DC-2-V1.ICP.NET (192.108.209.17) 72 msec 72 msec 88 msec
 2 ICM-FIX-E-H0-T3.ICP.NET (192.157.65.122) 80 msec 128 msec 80 msec
 3 192.203.229.246 540 msec 88 msec 84 msec
 4 T3-2.WASHINGTON-DC-CNSS58.T3.ANS.NET (140.222.58.3) 84 msec 116 msec 88 msec
 5 T3-3.WASHINGTON-DC-CNSS56.T3.ANS.NET (140.222.56.4) 80 msec 132 msec 88 msec
 6 T3-0.NEW-YORK-CNSS32.T3.ANS.NET (140.222.32.1) 92 msec 132 msec 88 msec
 7 T3-0.HARTFORD-CNSS48.T3.ANS.NET (140.222.48.1) 88 msec 88 msec 88 msec
 8 T3-0.HARTFORD-CNSS49.T3.ANS.NET (140.222.49.1) 96 msec 104 msec 96 msec
 9 T3-0.ENSS134.T3.ANS.NET (140.222.134.1) 92 msec 128 msec 92 msec
10 W91-CISCO-EXTERNAL-FDDI.MIT.EDU (192.233.33.1) 92 msec 92 msec 112 msec
11 E40-RTR-FDDI.MIT.EDU (18.168.0.2) 92 msec 120 msec 96 msec
12 MIT.EDU (18.72.2.1) 96 msec 92 msec 96 msec
```

Table 5-33 describes the fields that are unique to the extended trace sequence, as shown in the display.

**Table 5-33 Trace Field Descriptions**

| Field | Description |
|---|---|
| Target IP address | You must enter a host name or an IP address. There is no default. |
| Source address | One of the interface addresses of the router to use as a source address for the probes. The router will normally pick what it feels is the best source address to use. |
| Numeric display | The default is to have both a symbolic and numeric display; however, you can suppress the symbolic display. |
| Timeout in seconds | The number of seconds to wait for a response to a probe packet. The default is 3 seconds. |
| Probe count | The number of probes to be sent at each TTL level. The default count is 3. |
| Minimum Time to Live [1] | The TTL value for the first probes. The default is 1, but it can be set to a higher value to suppress the display of known hops. |
| Maximum Time to Live [30] | The largest TTL value that can be used. The default is 30. The **trace** command terminates when the destination is reached or when this value is reached. |
| Port Number | The destination port used by the UDP probe messages. The default is 33434. |
| Loose, Strict, Record, Timestamp, Verbose | IP header options. You can specify any combination. The **trace** command issues prompts for the required fields. Note that **trace** will place the requested options in each probe; however, there is no guarantee that all routers (or end nodes) will process the options. |
| Loose | Allows you to specify a list of nodes that must be traversed when going to the destination. |
| Strict | Allows you to specify a list of nodes that must be the only nodes traversed when going to the destination. |
| Record | Allows you to specify the number of hops to leave room for. |
| Timestamp | Allows you to specify the number of time stamps to leave room for. |
| Verbose | If you select any option, the verbose mode is automatically selected and **trace** prints the contents of the option field in any incoming packets. You can prevent verbose mode by selecting it again, toggling its current setting. |

Table 5-34 describes the characters that can appear in **trace** output.

**Table 5-34 IP Trace Text Characters**

| Char | Description |
|---|---|
| *nn* msec | For each node, the round-trip time in milliseconds for the specified number of probes. |
| * | The probe timed out. |
| ? | Unknown packet type. |
| Q | Source quench. |
| P | Protocol unreachable. |
| N | Network unreachable. |
| U | Port unreachable. |
| H | Host unreachable. |

## Related Command

**trace (user)**

# trace (user)

Use the **trace** EXEC command to discover the IP routes the router's packets will actually take when traveling to their destination.

**trace** [*protocol*] [*destination*]

## Syntax Description

| | |
|---|---|
| **protocol** | (Optional) Protocols that can be used are **appletalk**, **clns**, **ip** and **vines**. |
| *destination* | (Optional) Destination address or host name on the command line. The default parameters for the appropriate protocol are assumed and the tracing action begins. |

## Default

The *protocol* argument is based on the router's examination of the format of the *destination* argument. For example, if the router finds a *destination* in IP format, the *protocol* defaults to **ip**.

## Command Mode

EXEC

## Usage Guidelines

The **trace** command works by taking advantage of the error messages generated by routers when a datagram exceeds its time-to-live (TTL) value.

The **trace** command starts by sending probe datagrams with a TTL value of one. This causes the first router to discard the probe datagram and send back an error message. The **trace** command sends several probes at each TTL level and displays the round-trip time for each.

The **trace** command sends out one probe at a time. Each outgoing packet may result in one or two error messages. A *time exceeded* error message indicates that an intermediate router has seen and discarded the probe. A *destination unreachable* error message indicates that the destination node has received the probe and discarded it because it could not deliver the packet. If the timer goes off before a response comes in, **trace** prints an asterisk (*).

The **trace** command terminates when the destination responds, when the maximum TTL is exceeded, or when the user interrupts the trace with the escape sequence. By default, to invoke the escape sequence, press Ctrl-^ X—which is done by simultaneously pressing the Ctrl, Shift, and 6 keys, letting go, then pressing the X key.

## Common Trace Problems

Due to bugs in the IP implementation of various hosts and routers, the IP **trace** command may behave in odd ways.

Not all destinations will respond correctly to a probe message by sending back an *ICMP port unreachable* message. A long sequence of TTL levels with only asterisks, terminating only when the maximum TTL has been reached, may indicate this problem.

There is a known problem with the way some hosts handle an *ICMP TTL exceeded* message. Some hosts generate an *ICMP* message but they reuse the TTL of the incoming packet. Since this is zero, the ICMP packets do not make it back. When you trace the path to such a host, you may see a set of TTL values with asterisks (*). Eventually the TTL gets high enough that the *ICMP* message can get back. For example, if the host is six hops away, **trace** will time out on responses 6 through 11.

## Sample Display Showing Trace IP Routes

The following display shows sample IP **trace** output when a destination host name has been specified:

```
Router# trace ip ABA.NYC.mil

Type escape sequence to abort.
Tracing the route to ABA.NYC.mil (26.0.0.73)
  1 DEBRIS.CISCO.COM (131.108.1.6) 1000 msec 8 msec 4 msec
  2 BARRNET-GW.CISCO.COM (131.108.16.2) 8 msec 8 msec 8 msec
  3 EXTERNAL-A-GATEWAY.STANFORD.EDU (192.42.110.225) 8 msec 4 msec 4 msec
  4 BB2.SU.BARRNET.NET (131.119.254.6) 8 msec 8 msec 8 msec
  5 SU.ARC.BARRNET.NET (131.119.3.8) 12 msec 12 msec 8 msec
  6 MOFFETT-FLD-MB.in.MIL (192.52.195.1) 216 msec 120 msec 132 msec
  7 ABA.NYC.mil (26.0.0.73) 412 msec 628 msec 664 msec
```

Table 5-35 describes the fields shown in the display.

**Table 5-35 Trace Field Descriptions**

| Field | Description |
|---|---|
| 1 | Indicates the sequence number of the router in the path to the host. |
| DEBRIS.CISCO.COM | Host name of this router. |
| 131.108.1.61 | Internet address of this router. |
| 1000 msec 8 msec 4 msec | Round-trip time for each of the three probes that are sent. |

Table 5-36 describes the characters that can appear in **trace** output.

**Table 5-36 IP Trace Text Characters**

| Char | Description |
|---|---|
| *nn* msec | For each node, the round-trip time in milliseconds for the specified number of probes. |
| * | The probe timed out. |
| ? | Unknown packet type. |
| Q | Source quench. |
| P | Protocol unreachable. |
| N | Network unreachable. |
| U | Port unreachable. |
| H | Host unreachable. |

## Related Command

trace (privileged)

# username

To establish a username-based authentication system at login, even though your network cannot support a TACACS service, use the **username** global configuration command.

**username** *name* [**nopassword** | **password** *encryption-type* **password** *password*]
**username** *name* **password** *secret*
**username** *name* [**access-class** *number*]
**username** *name* [**autocommand** *command*]
**username** *name* [**noescape**] [**nohangup**]

## Syntax Description

| | |
|---|---|
| *name* | Host name, server name, user ID, or command name. The *name* argument can only be one word. White spaces and quotation marks are not allowed. |
| **nopassword** | (Optional) No password is required for this user to log in. This is usually most useful in combination with the **autocommand** keyword. |
| **password** | (Optional) Specifies a possibly encrypted password for this username. |
| *encryption-type* | (Optional) A single-digit number that defines whether the text immediately following is encrypted, and, if so, what type of encryption is used. Currently defined encryption types are 0, which means that the text immediately following is not encrypted, and 7, which means that the text is encrypted using a Cisco-defined encryption algorithm. |
| *password* | (Optional) A password can contain embedded spaces and must be the last option specified in the **username** command. |
| *secret* | For CHAP authentication: specifies the secret for the local router or the remote device. The secret is encrypted when it is stored on the local router. This prevents the secret from being stolen. The secret can consist of any string of up to 11 printable ASCII characters. There is no limit to the number of username/password combinations that can be specified, allowing any number of remote devices to be authenticated. |
| **access-class** | (Optional) Specifies an outgoing access list that overrides the access list specified in the **access-class** line configuration command. It is used for the duration of the user’s session. |
| *number* | (Optional) The access list number. |
| **autocommand** | (Optional) Causes the specified command to be issued automatically after the user logs in. When the command is complete, the session is terminated. As the command can be any length and contain imbedded spaces, commands using the **autocommand** keyword must be the last option on the line. |
| *command* | (Optional) The command string. |
| **noescape** | (Optional) Prevents a user from using an escape character on the host to which that user is connected. |

| | |
|---|---|
| **nohangup** | (Optional) Prevents the communication server from disconnecting the user after an automatic command (set up with the **autocommand** keyword) has completed. Instead, the user gets another login prompt. |

## Default

None

## Command Mode

Global configuration

## Usage Guidelines

The **username** command provides username/password authentication for login purposes only. (Note that it does not provide username/password authentication for enable mode when the **enable use-tacacs** command is also used.)

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

Add a **username** entry for each remote system that the local router communicates with and requires authentication from. The remote device must have a **username** entry for the local router. This entry must have the same password as the local router's entry for that remote device.

This command can be useful for defining usernames that get special treatment, for example, an "info" username that does not require a password, but connects the user to a general purpose information service.

The **username** command is also required as part of the configuration for the Challenge Handshake Authentication Protocol (CHAP). For each remote system that the local router communicates with from which it requires authentication, add a **username** entry.

---

**Note** To enable the local router to respond to remote CHAP challenges, one **username** *name* entry must be the same as the **hostname** *name* entry that has already been assigned to your router.

---

If there is no *secret* specified and **debug serial-interface** is enabled, an error is displayed when a link is established and the CHAP challenge is not implemented. Debugging information on CHAP is available using the **debug serial-interface** and **debug serial-packet** commands. For more information about **debug** commands, refer to the *Debug Command Reference* publication.

## Examples

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

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

To implement an information service that does not require a password to be used, the command takes the following form:

```
username info nopassword noescape autocommand telnet nic.ddn.mil
```

To implement an ID that will work even if the TACACS servers all break, the command takes the following form:

```
username superuser password superpassword
```

The following example configuration enables CHAP on interface serial 0. It also defines a password for the local server, Adam, and a remote server, Eve.

```
hostname Adam
interface serial 0
encapsulation ppp
ppp authentication chap
username Adam password oursystem
username Eve password theirsystem
```

When you look at your configuration file, the passwords will be encrypted and the display will look similar to the following:

```
hostname Adam
interface serial 0
encapsulation ppp
ppp authentication chap
username Adam password 7 1514040356
username Eve password 7 121F0A18
```

## Related Command

**hostname**

CHAPTER 6

# Interface Commands

This chapter contains the commands used to configure nonprotocol-specific interface features. The commands are in alphabetical order. For hardware technical descriptions, and for information about installing the router interfaces, refer to the hardware installation and maintenance publication for your particular product.

For interface configuration tasks and examples, refer to the chapter entitled “Configuring Interfaces” in the *Router Products Configuration Guide*.

For a conversion table of the modular products and Cisco 7000 series processors, see the appendix entitled “Cisco 7000 Processors.”

**Note** For information about the Channel Interface Processor (CIP), see the chapter entitled “IBM Channel Attach Commands.” The CIP is described in a separate chapter because of the interrelationship of host system configuration values and router configuration values.

# async default ip address

To assign the interface address that is used by the device connecting to the router via PPP or SLIP, unless you override the address at the command line, use the **async default ip address** interface configuration command. Use the **no** form of this command to remove the address from your configuration.

**async default ip address** *ip-address*
**no async default ip address**

## Syntax Description

| | |
|---|---|
| *ip-address* | Address of the client interface. |

## Default

No interface address is assigned.

## Command Mode

Interface configuration

## Example

The following example specifies address 182.32.7.51 on asynchronous interface 1:

```
interface async 1
async default ip address 182.32.7.51
```

## Related Command

**async dynamic address**

# async dynamic address

To specify an address on an asynchronous interface (rather than using the default address), use the **async dynamic address** interface configuration command. Use the **no** form of this command to disable dynamic addressing.

**async dynamic address**
**no async dynamic address**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Example

The following example shows dynamic addressing assigned to an interface:

```
interface async 1
async dynamic address
```

## Related Commands

**ppp**
**slip**

# async dynamic routing

To implement asynchronous routing on an interface, use the **async dynamic routing** interface configuration command. The **no** form of this command disables use of routing protocols; static routing will still be used.

**async dynamic routing**
**no async dynamic routing**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Example

The following example shows how to enable asynchronous routing on asynchronous interface 1. The **ip tcp header-compression passive** command enables Van Jacobson TCP header compression and prevents transmission of compressed packets until a compressed packet arrives from the asynchronous link.

```
interface async 1
async dynamic routing
async dynamic address
async default ip address 1.1.1.2
ip tcp header-compression passive
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**async dynamic address**
**ip tcp header-compression** †

# async mode dedicated

To place a line into network mode using SLIP or PPP encapsulation, use the **async mode dedicated** interface configuration command. The **no** form of this command returns the line to interactive mode.

**async mode dedicated**
**no async mode**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

With dedicated asynchronous mode, the interface will use either SLIP or PPP encapsulation, depending on which **encapsulation** method is configured for the interface. An EXEC prompt does not appear, and the line is not available for normal interactive use.

If you configure a line for dedicated mode, you will not be able to use **async dynamic address**, because there is no user prompt. You must configure either **async default ip address** and **ip unnumbered** or **ip address**.

## Example

The following example assigns an Internet address to an asynchronous line and places the line into network mode. Setting the stop bits to 1 enhances performance.

```
interface async 1
async default ip address 182.32.7.51
async mode dedicated
encapsulation slip
```

## Related Command

**async mode interactive**

# async mode interactive

To enable the **slip** and **ppp** EXEC commands, use the **async mode interactive** line configuration command. Use the **no** form of this command to prevent users from implementing SLIP and PPP at the EXEC level.

**async mode interactive**
**no async mode**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Example

The following example enables the **ppp** and **slip** EXEC commands:

```
interface async 1
async mode interactive
```

## Related Commands

**async mode dedicated**
**ppp**
**slip**

# auto-polarity

To enable automatic receiver polarity reversal on a hub port connected to an Ethernet interface of a Cisco 2505 or Cisco 2507, use the **auto-polarity** hub configuration command. To disable this feature, use the **no** form of this command.

**auto-polarity**
**no auto-polarity**

## Syntax Description

This command has no arguments or keywords.

## Default

Enabled

## Command Mode

Hub configuration

## Usage Guidelines

This command applies to a port on an Ethernet hub only.

## Example

The following example enables automatic receiver polarity reversal on hub 0, ports 1 through 3:

```
hub ethernet 0 1 3
auto-polarity
```

## Related Command

**hub**

# backup delay

To define how much time should elapse before a secondary line is set up or taken down after a primary line transition, use the **backup delay** interface configuration command. Use the **no backup delay** command to remove the definition.

**backup delay** {*enable-delay* | **never**} {*disable-delay* | **never**}
**no backup delay** {*enable-delay* | **never**} {*disable-delay* | **never**}

## Syntax Description

| | |
|---|---|
| *enable-delay* | Integer that specifies the delay in seconds after the primary line goes down before the secondary line is activated. |
| *disable-delay* | Integer that specifies the delay in seconds after the primary line goes up before the secondary line is deactivated. |
| **never** | Prevents the secondary line from being activated or deactivated. |

## Default

**never**

## Command Mode

Interface configuration

## Usage Guidelines

When a primary line goes down, the router delays the amount of seconds defined by the *enable-delay* argument before enabling the secondary line. If, after the delay period, the primary line is still down, the secondary line is activated.

When a primary line comes back up, the router will delay the amount of seconds defined by the *disable-delay* argument.

---

**Note** In cases where there are spurious signal disruptions that may appear as intermittent lost carrier signals, it is recommended that some delay be enabled before activating and deactivating a secondary.

---

**Note** The interval configured with the **backup delay** command does not affect the operation of the **backup load** command.

---

## Examples

The following example sets a 10-second delay on deactivating the secondary line; however, the line is activated immediately:

```
interface serial 0
backup delay 0 10
```

The same example on the Cisco 7000 requires the following commands:

```
interface serial 1/1
backup delay 0 10
```

# backup interface

To configure the serial interface as a secondary, or dial backup line, use the **backup interface** interface configuration command. Use the **no backup** command with the appropriate serial port designation to turn disable this feature.

**backup interface** *interface-name*
**backup interface** *interface-name slot*/*port* (for the Cisco 7000 series)
**no backup interface** *interface-name*

## Syntax Description

| | |
|---|---|
| *interface-name* | Serial port to be set as the secondary interface line. |
| *slot* | On the Cisco 7000 series, specifies the slot number. |
| *port* | On the Cisco 7000 series, specifies the port number. |

## Default

Disabled

## Command Mode

Interface configuration

## Examples

The following example sets serial 1 as the backup line to serial 0:

```
interface serial 0
backup interface serial 1
```

The following example on the Cisco 7000 sets serial 2 as the backup line to serial 1:

```
interface serial 1/1
backup interface serial 2/2
```

## Related Command

**down-when-looped**

# backup load

To set the traffic load thresholds for dial backup service, use the **backup load** interface configuration command. Use the **no backup load** command to remove the setting.

**backup load** {*enable-threshold* | **never**} {*disable-load* | **never**}
**no backup load** {*enable-threshold* | **never**} {*disable-load* | **never**}

## Syntax Description

| | |
|---|---|
| *enable-threshold* | Integer that specifies a percentage of the primary line's available bandwidth. |
| **never** | Specifies that the secondary line never be activated due to load. |
| *disable-load* | Integer that specifies a percentage of the primary line's available bandwidth. |
| **never** | Specifies that the secondary line never be deactivated due to load. |

## Default

Both arguments default to **never**.

## Command Mode

Interface configuration

## Usage Guidelines

When the transmitted or received load on the primary line is greater than the value assigned to the *enable-threshold* argument, the secondary line is enabled.

When the transmitted load on the primary line plus the transmitted load on the secondary line is less than the value entered for the *disable-load* argument, and the received load on the primary line plus the received load on the secondary line is less than the value entered for the *disable-load* argument, the secondary line is disabled.

If the **never** keyword is used instead of an *enable-threshold* value, the secondary line is never activated because of load. If the **never** keyword is used instead of a *disable-load* value, the secondary line is never deactivated because of load.

## Examples

The following example sets the traffic load threshold to 60 percent on the primary line. When that load is exceeded, the secondary line is activated, and will not be deactivated until the combined load is less than 5 percent of the primary bandwidth.

```
interface serial 0
backup load 60 5
```

The same example on the Cisco 7000 requires the following commands:

```
interface serial 1/1
backup load 60 5
```

# bandwidth

To set a bandwidth value for an interface, use the **bandwidth** interface configuration command. Use the **no bandwidth** command to restore the default values.

**bandwidth** *kilobits*
**no bandwidth**

## Syntax Description

| | |
|---|---|
| *kilobits* | Intended bandwidth in kilobits per second. For a full bandwidth DS3, enter the value **44736**. |

## Default

Default bandwidth values are set during startup and can be displayed with the EXEC command **show interfaces**.

## Command Mode

Interface configuration

## Usage Guidelines

The **bandwidth** command sets an informational parameter only; you cannot adjust the actual bandwidth of an interface with this command. For some media, such as Ethernet, the bandwidth is fixed; for other media, such as serial lines, you can change the actual bandwidth by adjusting hardware. For both classes of media, you can use the **bandwidth** configuration command to communicate the current bandwidth to the higher-level protocols.

Additionally, IGRP uses the minimum path bandwidth to determine a routing metric. The TCP protocol adjusts initial retransmission parameters based on the apparent bandwidth of the outgoing interface.

At higher bandwidths, the value you configure with the **bandwidth** command is not what is displayed by the **show interface** command. The value shown is that used in IGRP updates and also used in computing load.

**Note** This is a routing parameter only; it does not affect the physical interface.

## Example

The following example sets the full bandwidth for DS3 transmissions:

```
interface serial 0
bandwidth 44736
```

## Related Command

A dagger (†) indicates that the command is documented in another chapter.

**vines metric** †

# channel-group

Use the **channel-group** controller configuration command to define the timeslots that belong to each T1 or E1 circuit.

**channel-group** *number* **timeslots** *range* [**speed** {**48** | **56** | **64**}]

## Syntax Description

| | |
|---|---|
| *number* | Channel-group number. When configuring a T1 data line, channel-group numbers can be a value from 0 to 23. When configuring an E1 data line, channel-group numbers can be a value from 0 to 30. |
| **timeslots** *range* | Timeslot or range of timeslots belonging to the channel group. The first timeslot is numbered 1. For a T1 controller, the timeslot range is from 1 to 24. For an E1 controller, the timeslot range is from 1 to 31. |
| **speed** {**48** \| **56** \| **64**} | (Optional) Specifies the line speed (in kilobits per second) of the T1 or E1 link. |

## Default

The default line speed when configuring a T1 controller is 56 kbps.

The default line speed when configuring an E1 controller is 64 kbps.

## Command Mode

Controller configuration

## Usage Guidelines

Use this command in configurations where the router is intended to communicate with a T1 or E1 fractional data line. The channel-group number may be arbitrarily assigned and must be unique for the controller. The timeslot range must match the timeslots assigned to the channel group. The service provider defines the timeslots that comprise a channel group.

## Example

In the following example, three channel groups are defined. Channel-group 0 consists of a single timeslot, channel-group 8 consists of 7 timeslots and runs at a speed of 64 kbps per timeslot, and channel-group 12 consists of a single timeslot.

```
channel-group 0 timeslots 1
channel-group 8 timeslots 5,7,12-15,20 speed 64
channel-group 12 timeslots 2
```

## Related Commands

**linecode**
**framing**

# clear controller lex

To reboot the LAN Extender chassis and restart its operating software, use the **clear controller lex** privileged EXEC command.

**clear controller lex** *number* [**prom**]
**clear controller lex** *slot*/*port* [**prom**] (for the Cisco 7000 series)

## Syntax Description

| | |
|---|---|
| *number* | Number of the LAN Extender interface corresponding to the LAN Extender to be rebooted. |
| **prom** | (Optional) Forces a reload of the PROM image, regardless of any Flash image. |
| *slot* | On the Cisco 7000 series, specifies the backplane slot number. On the Cisco 7000, the value can be 0, 1, 2, 3, or 4. On the Cisco 7010, the value can be 0, 1, or 2. |
| *port* | On the Cisco 7000 series, specifies the port number of the interface. The value can be 0, 1, 2, or 3 for the serial interface. |

## Command Mode

Privileged EXEC

## Usage Guidelines

The **clear controller lex** command halts operation of the LAN Extender and performs a cold restart.

Without the **prom** keyword, if an image exists in Flash memory, and that image has a newer software version than the PROM image, and that image has a valid checksum, then this command runs the Flash image. If any one of these three conditions is not met, this command reloads the PROM image.

With the **prom** keyword, this command reloads the PROM image, regardless of any Flash image.

## Examples

The following example halts operation of the LAN Extender bound to LAN Extender interface 2 and causes the LAN Extender to perform a cold restart from Flash memory:

```
Router# clear controller lex 2
reload remote lex controller? [confirm] yes
```

The following example halts operation of the LAN Extender bound to LAN Extender interface 2 and causes the LAN Extender to perform a cold restart from PROM:

```
Router# clear controller lex 2 prom
reload remote lex controller? [confirm] yes
```

# clear controller

Use the **clear controller** EXEC command to reset the T1 or E1 controller interface on the Cisco 7000 series or Cisco 4000 series routers.

**clear controller** {**t1** | **e1**}*slot*/*port* (Cisco 7000)
**clear controller** {**t1** | **e1**} *number* (Cisco 4000)

## Syntax Description

| | |
|---|---|
| *slot* | Backplane slot number; can be 0, 1, 2, 3, or 4. The slots are numbered from left to right. |
| *port* | Port number of the interface. It can be **0** or **1** depending on the type of controller, as follows:<br>• MIP (MultiChannel Interface Processor) **0 or 1**<br>Ports on each interface processor are numbered from the top down. |
| *number* | Network interface module (NIM) number, in the range 0 through 2. |

## Command Mode

EXEC

## Example

The following example resets the T1 controller at slot 4, port 0 on a Cisco 7000 series router:

```
clear controller t1 4/0
```

The following example resets the E1 controller at NIM 0 on a Cisco 4000 series router:

```
clear controller e1 0
```

## Related Command

**controller e1**
**controller t1**

# clear counters

To clear the interface counters, use the **clear counters** EXEC command.

**clear counters** [*type number*] [**ethernet** | **serial**]
**clear counters** [*type slot*/*port*] [**ethernet** | **serial**] (for the Cisco 7000 series)

## Syntax Description

| | |
|---|---|
| *type* | (Optional) Specifies the interface type; it is one of the keywords listed in Table 6-1. |
| *number* | (Optional) Specifies the interface counter displayed with the **show interfaces** command. |
| **ethernet** | (Optional) If the *type* is **lex**, you can clear the interface counters on the Ethernet interface. |
| **serial** | (Optional) If the *type* is **lex**, you can clear the interface counters on the serial interface. |
| *slot* | (Optional) On the Cisco 7000 series, specifies the backplane slot number. On the Cisco 7000, the value can be 0, 1, 2, 3, or 4. On the Cisco 7010, the value can be 0, 1, or 2. |
| *port* | (Optional) On the Cisco 7000 series, specifies the port number of the interface. The value can be 0, 1, 2, or 3 for the serial interface. |

**Table 6-1 Clear Counters Interface Type Keywords**

| Keyword | Interface Type |
|---|---|
| **async** | Asynchronous interface |
| **bri** | Integrated Services Digital Network (ISDN) Basic Rate Interface (BRI) |
| **dialer** | Dialer interface |
| **ethernet** | Ethernet interface |
| **fddi** | Fiber Distributed Data Interface (FDDI) |
| **hssi** | High-Speed Serial Interface (HSSI) |
| **lex** | LAN Extender interface |
| **loopback** | Loopback interface |
| **null** | Null interface |
| **serial** | Synchronous serial interface |
| **tokenring** | Token Ring interface |
| **tunnel** | Tunnel interface |

## Command Mode

EXEC

## Usage Guidelines

This command clears all the current interface counters from the interface unless the optional arguments *type* and *number* are specified to clear only a specific interface type (serial, Ethernet, Token Ring, and so on).

---

**Note** This command will not clear counters retrieved using SNMP, but only those seen with the EXEC **show interface** command.

---

## Examples

The following example illustrates how to clear all interface counters:

```
clear counters
```

The following example illustrates how to clear interface counters on the serial interface residing on a Cisco 1000 series LAN Extender:

```
clear counters lex 0 serial
```

## Related Command

**show interfaces**

# clear hub

To reset and reinitialize the hub hardware connected to an interface of a Cisco 2505 or Cisco 2507, use the **clear hub** EXEC command.

**clear hub ethernet** *number*

## Syntax Description

| | |
|---|---|
| **ethernet** | Indicates the hub in front of an Ethernet interface. |
| *number* | Hub number to clear, starting with 0. Since there is currently only one hub, this number is 0. |

## Command Mode

EXEC

## Example

The following example clears hub 0:

```
clear hub ethernet 0
```

## Related Command

**hub**

# clear hub counters

To set to zero the hub counters on an interface of a Cisco 2505 or Cisco 2507, use the **clear hub counters** EXEC command.

**clear hub counters** [**ether** *number* [*port* [*end-port*]]]

## Syntax Description

| | |
|---|---|
| **ether** | (Optional) Indicates the hub in front of an Ethernet interface. |
| *number* | (Optional) Hub number for which to clear counters. Since there is currently only one hub, this number is 0. If you specify the keyword **ether**, you must specify the *number* . |
| *port* | (Optional) Port number on the hub. On the Cisco 2505, port numbers range from 1 through 8. On the Cisco 2507, port numbers range from 1 through 16. If a second port number follows, then this port number indicates the beginning of a port range. If you do not specify a port number, counters for all ports are cleared. |
| *end-port* | (Optional) Ending port number of a range. |

## Command Mode

EXEC

## Example

The following example clears the counters displayed in a **show hub** command for all ports on hub 0:

```
clear hub counters ether 0
```

## Related Command

**show hub**

# clear interface

To reset the hardware logic on an interface, use the **clear interface** EXEC command.

**clear interface** *type number*
**clear interface** *type slot*/*port* (on a Cisco 7000 series)
**clear interface** *type slot*/*port* [**:***channel-group*] (on a Cisco 7000 MIP T1 interface)

## Syntax Description

| | |
|---|---|
| *type* | Specifies the interface type; it is one of the keywords listed in Table 6-2. |
| *number* | Specifies the port, connector, or interface card number. |
| *slot* | On the Cisco 7000 series, specifies the backplane slot number. On the 7000, value can be 0, 1, 2, 3, or 4. On the 7010, value can be 0, 1, or 2. |
| *port* | On the Cisco 7000 series, specifies the port number of the interface and can be 0, 1, 2, 3, 4 or 5 depending on the type of interface, as follows:<br>• AIP (ATM Interface Processor) 0<br>• EIP (Ethernet Interface Processor) 0, 1, 2, 3, 4, or 5<br>• FIP (FDDI Interface Processor) 0<br>• HIP (HSSI Interface Processor) 0<br>• MIP (Multichannel Interface Processor) 0 or 1<br>• TRIP (Token Ring Interface Processor) 0, 1, 2, or 3 |
| *channel-group* | (Optional) On the Cisco 7000 series supporting channelized T1, specifies the channel and can be between 0 and 23. |

**Table 6-2 Clear Interface Type Keywords**

| Keyword | Interface Type |
|---|---|
| **async** | Async interface |
| **atm** | Asynchronous Transfer Mode (ATM) interface |
| **bri** | Integrated Services Digital Network (ISDN) Basic Rate Interface (BRI) |
| **ethernet** | Ethernet interface |
| **fddi** | Fiber Distributed Data Interface (FDDI) |
| **hssi** | High-Speed Serial Interface (HSSI) |
| **loopback** | Loopback interface |
| **null** | Null interface |
| **serial** | Synchronous serial interface |
| **tokenring** | Token Ring interface |
| **tunnel** | Tunnel interface |

## Command Mode

EXEC

**Note** Under normal circumstances, you do not need to clear the hardware logic on interfaces.

## Example

The following example resets the interface logic on HSSI interface 1:

```
clear interface hssi 1
```

# clear rif-cache

To clear entries from the Routing Information Field (RIF) cache, use the **clear rif-cache** EXEC command.

**clear rif-cache**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Example

The following example illustrates how to clear the RIF cache:

```
clear rif-cache
```

## Related Command

A dagger (†) indicates that the command is documented in another chapter.

**multiring** †

# clock source (controller)

Use the **clock source** controller configuration command to set the T1-line clock-source for the MIP in the Cisco 7000 or for the NIM in the Cisco 4000.

**clock source** {**line** | **internal**}

## Syntax Description

| | |
|---|---|
| **line** | Specifies the T1 line as the clock source. |
| **internal** | Specifies the MIP (Cisco 7000) or the NIM (Cisco 4000) as the clock source. |

## Default

T1 line

## Command Mode

Controller configuration

## Usage Guidelines

This command is used in configurations where the interfaces are connected back-to-back, rather than to a T1 line, and one of the interfaces must provide a clocking signal. When the interface is connected to a channelized T1 line, this command need never be used.

## Example

The following example enables internal clocking:

```
clock source internal
```

## Related Commands

**framing**
**linecode**

# clock source (interface)

To control which clock a G.703-E1 interface will use to clock its transmitted data from, use the **clock source** interface configuration command. The **no** form of this command restores the default value.

**clock source** {**line** | **internal**}
**no clock source**

## Syntax Description

| | |
|---|---|
| **line** | Specifies that the interface will clock its transmitted data from a clock recovered from the line's receive data stream (default). |
| **internal** | Specifies that the interface will clock its transmitted data from its internal clock. |

## Default

By default, the applique uses the line's receive data stream.

## Command Mode

Interface configuration

## Usage Guidelines

This command applies to a Cisco 4000 router or Cisco 7000 series router. A G.703-E1 interface can clock its transmitted data from either its internal clock or from a clock recovered from the line's receive data stream.

## Example

The following example specifies the G.703-E1 interface to clock its transmitted data from its internal clock:

```
clock source internal
```

# clock rate

To configure the clock rate for the hardware connections on the serial interface appliques, network interface modules (NIMs), and interface processors (IPs) to an acceptable bit rate, use the **clock rate** interface configuration command. Use the **no clock rate** command to remove the clock rate if you change the interface from a DCE to a DTE device.

**clock rate** *bps*
**no clock rate**

## Syntax Description

| | |
|---|---|
| *bps* | Desired clock rate in bits per second: 1200, 2400, 4800, 9600, 19200, 38400, 56000, 64000, 72000, 125000, 148000, 500000, 800000, 1000000, 1300000, 2000000, or 4000000. |

## Default

No clock rate is configured.

## Command Mode

Interface configuration

## Usage Guidelines

Be aware that the fastest speeds might not work if your cable is too long, and that speeds faster than 148,000 bits per second are too fast for RS-232 signaling. It is recommended that you only use the synchronous serial RS-232 signal at speeds up to 64,000 bits per second. To permit a faster speed, use an RS-449 or V.35 applique.

## Example

The following example sets the clock rate on the first serial interface to 64,000 bits per second:

```
interface serial 0
clock rate 64000
```

# cmt connect

To start the processes that perform the connection management (CMT) function and allow the ring on one fiber to be started, use the **cmt connect** EXEC command.

**cmt connect** [*interface-name* [**phy-a** | **phy-b**]]

## Syntax Description

| | |
|---|---|
| *interface-name* | (Optional) Specifies the FDDI interface. |
| **phy-a** | (Optional) Selects Physical Sublayer A. |
| **phy-b** | (Optional) Selects Physical Sublayer B. |

## Command Mode

EXEC

## Usage Guidelines

In normal operation, the FDDI interface is operational once the interface is connected and configured. The **cmt connect** command allows the operator to start the processes that perform the CMT function.

The **cmt connect** command is not needed in the normal operation of FDDI; this command is used mainly in interoperability tests.

## Examples

The following examples demonstrate use of the **cmt connect** command for starting the CMT processes on the FDDI ring.

The following command starts all FDDI interfaces:

```
cmt connect
```

The following command starts both fibers on the FDDI interface unit zero:

```
cmt connect fddi 0
```

The following command on the Cisco 7000 starts both fibers on the FDDI interface unit zero:

```
cmt connect fddi 1/0
```

The following command starts only Physical Sublayer A on the FDDI interface unit 0 (zero):

```
cmt connect fddi 0 phy-a
```

The following command on the Cisco 7000 starts only Physical Sublayer A on the FDDI interface unit 0 (zero):

```
cmt connect fddi 1/0 phy-a
```

# cmt disconnect

To stop the processes that perform the connection management (CMT) function and allow the ring on one fiber to be stopped, use the **cmt disconnect** EXEC command.

**cmt disconnect** [*interface-name* [**phy-a** | **phy-b**]]

## Syntax Description

| | |
|---|---|
| *interface-name* | (Optional) Specifies the FDDI interface. |
| **phy-a** | (Optional) Selects Physical Sublayer A. |
| **phy-b** | (Optional) Selects Physical Sublayer B. |

## Command Mode

EXEC

## Usage Guidelines

In normal operation, the FDDI interface is operational once the interface is connected and configured, and is turned off using the **shutdown** interface configuration command. The **cmt disconnect** command allows the operator to stop the processes that perform the CMT function and allow the ring on one fiber to be stopped.

The **cmt disconnect** command is not needed in the normal operation of FDDI; this command is used mainly in interoperability tests.

## Examples

The following examples demonstrate use of the **cmt disconnect** command for stopping the CMT processes on the FDDI ring.

The following command stops all FDDI interfaces:

```
cmt disconnect
```

The following command stops both fibers on the FDDI interface unit zero:

```
cmt disconnect fddi 0
```

The following command on the Cisco 7000 stops both fibers on the FDDI interface unit zero:

```
cmt disconnect fddi 1/0
```

The following command stops only Physical Sublayer A on the FDDI interface unit 0 (zero). This command causes the FDDI media to go into a wrapped state so that the ring will be broken.

```
cmt disconnect fddi 0 phy-a
```

The following command on the Cisco 7000 stops only Physical Sublayer A on the FDDI interface unit 0 (zero). This command causes the FDDI media to go into a wrapped state so that the ring will be broken.

```
cmt disconnect fddi 1/0 phy-a
```

# compress

To configure software compression for Link Access Procedure, Balanced (LAPB), Point-to-Point Protocol (PPP), and High-Level Data Link Control (HDLC) encapsulations, use the **compress** interface configuration command. To disable compression, use the **no** form of this command.

**compress** [**predictor** | **stac**]
**no compress** [**predictor** | **stac**]

## Syntax Description

| | |
|---|---|
| **predictor** | (Optional) Specifies that a predictor compression algorithm will be used on LAPB and PPP encapsulation. |
| **stac** | (Optional) Specifies that a Stacker (LZS) compression algorithm will be used on HDLC and PPP encapsulation. |

## Default

Compression is disabled.

## Command Mode

Interface configuration

## Usage Guidelines

You can configure point-to-point software compression for all LAPB, PPP, and HDLC encapsulations. Compression reduces the size of frames via lossless data compression. The compression algorithm used is a predictor algorithm (the RAND compression algorithm), which uses a compression dictionary to predict what the next character in the frame will be.

For HDLC encapsulations, you can specify a Stacker compression algorithm by using the **stac** keyword. PPP and LAPB encapsulations support both predictor and Stacker compression algorithms.

Compression is performed in software and may significantly affect system performance. We recommend that you disable compression if CPU load exceeds 65 percent. To display the CPU load, use the **show process cpu** EXEC command.

Compression requires that both ends of the serial link be configured to use compression. You should never enable compression for connections to a public data network.

---

**Note** The best performance data compression algorithms adjust their compression methodology as they identify patterns in the data. To prevent data loss and support this adjustment process, the compression algorithm is run over LAPB to ensure that everything is sent in order, with no missing data and no duplicate data.

---

If the majority of your traffic is already compressed files, we recommend that you not use compression. If the files are already compressed, the additional processing time spent in attempting unsuccessfully to compress them again will slow system performance.

Table 6-3 provides general guidelines for deciding which compression type to select for LAPB encapsulations.

**Table 6-3 Compression Guidelines for LAPB Encapsulations**

| Compression Type to Use | Situation |
|---|---|
| Predictor | The bottleneck is the load on the router. |
| Stacker | The bottleneck is line bandwidth. |
| None | Most files are already compressed. |

Stacker compression for LAPB encapsulations reaches its performance ceiling on T1 lines; it is not recommended for faster lines because the added processing slows their performance. Stacker compression processing might be slower on other systems than on the Cisco 4500 routers.

When using predictor compression, you can adjust the MTU for the serial interface and the LAPB maximim bits per frame (N1) parameter, as shown in the first example, to avoid informational diagnostics regarding excessive MTU or N1 sizes. However, you should not change those parameters when you use Stacker compression.

## Examples

The following example enables predictor compression on serial interface 0 for a LAPB link:

```
interface serial 0
encapsulation lapb
compress predictor
mtu 1509
lapb n1 12072
```

The following example enables Stacker compression on serial interface 0 for a LAPB link. This example does not set the MTU size and the maximum bits per frame (N1); we recommend that you do not change those LAPB parameters for Stacker compression:

```
interface serial 0
encapsulation lapb
compress predictor
```

## Related Commands

A dagger (**†**) indicates that the command is documented in another chapter.

**encapsulation lapb**
**encapsulation x25**
**show compress**
**show processes** †

# controller

To configure a T1 or E1 controller and enter controller configuration mode, use the **controller** global configuration command.

**controller** [**t1** | **e1**] *slot*/*port* (on the Cisco 7000)

**controller [t1 | e1]** *number* (on the Cisco 4000)

## Syntax Description

| | |
|---|---|
| **t1** | T1 controller. |
| **e1** | E1 controller. |
| *slot* | Backplane slot number; can be 0, 1, 2, 3, or 4. On the Cisco 7010, the slot number can be 0, 1, or 2. The slots are numbered from left to right. |
| *port* | Port number of the interface. It can be **0** or **1** for the MIP (MultiChannel Interface Processor). Ports on each interface processor are numbered from the top down. |
| *number* | Network interface module (NIM) number, in the range 0 through 2. |

## Default

No T1 or E1 controller is configured.

## Command Mode

Global configuration

## Usage Guidelines

This command is used in configurations where the router is intended to communicate with a T1 or E1 fractional data line. Additional parameters for the T1 or E1 line must be configured for the controller before the T1 or E1 circuits can be configured by means of the **interface** global configuration command.

This command is used only on a Cisco 7000 or Cisco 4000 series router.

## Example

In the following example, the MIP in slot 4, port 0 of a Cisco 7000 is configured as a T1 controller:

```
controller t1 4/0
```

In the following example, NIM 0 of a Cisco 4000 is configured as a T1 controller:

```
controller t1 0
```

## Related Commands

**channel-group**
**clear controller lex**
**clear controller t1**
**clock source (controller)**
**framing**
**linecode**
**show controllers e1**
**show controllers t1**

# copy flash lex

To download an executable image from Flash memory on the core router to the LAN Extender chassis, use the **copy flash lex** privileged EXEC command.

**copy flash lex** *number*

## Syntax Description

| | |
|---|---|
| *number* | Number of the LAN Extender interface to which to download an image from Flash memory. |

## Command Mode

Privileged EXEC

## Usage Guidelines

If you attempt to download a version of the software older than what is currently running on the LAN Extender, a warning message is displayed.

## Example

The following example illustrates how to copy the executable image *namexx* to the LAN Extender interface 0:

```
Router# copy flash lex 0
Name of file to copy? namexx
Address of remote host [255.255.255.255] <cr>
writing namexx !!!!!!!!!!!!!!!!!!!!!!!!!copy complete
```

## Related Command

**copy tftp lex**

# copy tftp lex

To download an executable image from a TFTP server to the LAN Extender, use the **copy tftp lex** privileged EXEC command.

**copy tftp lex** *number*

## Syntax Description

| | |
|---|---|
| *number* | Number of the LAN Extender interface to which to download an image. |

## Command Mode

Privileged EXEC

## Usage Guidelines

If you attempt to download a version of the software older than what is currently running on the LAN Extender, a warning message is displayed.

## Example

The following example illustrates how to copy the file *namexx* from the TFTP server:

```
Router# copy tftp lex 0
Address or name of remote host (255.255.255.255]? 131.108.1.111
Name of file to copy? namexx
OK to overwrite software version 1.0 with 1.1 ?[confirm]
Loading namexx from 131.108.13.111!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 127825/131072 bytes]

Successful download to LAN Extender
```

# crc

To set the length of the cyclic redundancy check (CRC) on a Fast Serial Interface Processor (FSIP) or HSSI Interface Processor (HIP) of the Cisco 7000 series only, use the **crc** interface configuration command. To set the CRC length to 16 bits, use the **no** form of this command.

**crc** *size*
**no crc**

## Syntax Description

| | |
|---|---|
| *size* | CRC size (16 or 32 bits). |

## Default

16 bits

## Command Mode

Interface configuration

## Usage Guidelines

All interfaces use a 16-bit cyclic redundancy check (CRC) by default, but also support a 32-bit CRC. CRC is an error-checking technique that uses a calculated numeric value to detect errors in transmitted data. The designators 16 and 32 indicate the length (in bits) of the frame check sequence (FCS). A CRC of 32 bits provides more powerful error detection, but adds overhead. Both the sender and receiver must use the same setting.

CRC-16, the most widely used throughout the United States and Europe, is used extensively with wide-area networks (WANs). CRC-32 is specified by IEEE 802 and as an option by some point-to-point transmission standards. It is often used on SMDS networks and LANs.

## Example

In the following example, the 32-bit CRC is enabled on serial interface 3/0:

```
interface serial 3/0
crc 32
```

# crc4

To enable generation of the G.703-E1 CRC4, use the **crc4** interface configuration command. To disable this feature, use the **no** form of this command.

**crc4**
**no crc4**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

This command applies to a Cisco 4000 router or Cisco 7000 series router. It is useful for checking data integrity while operating in framed mode. CRC4 provides additional protection for a frame alignment signal under noisy conditions. Refer to CCITT Recommendation G.704 for a definition of CRC4.

## Example

The following example enables CRC4 generation on the G.703-E1 interface:

```
crc4
```

# dce-terminal-timing enable

When running the line at high speeds and long distances, use the **dce-terminal-timing enable** interface configuration command to prevent phase shifting of the data with respect to the clock. If SCTE is not available from the DTE, use **no dce-terminal-timing-enable**, which causes the DCE to use its own clock instead of SCTE from the DTE.

**dce-terminal-timing enable**
**no dce-terminal-timing enable**

## Syntax Description

This command has no keywords or arguments.

## Default

DCE uses its own clock.

## Command Mode

Interface configuration

## Usage Guidelines

On the Cisco 4000 platform, you can specify the serial Network Interface Module timing signal configuration. When the board is operating as a DCE and the DTE provides terminal timing (SCTE or TT), the **dce-terminal-timing enable** command causes the DCE to use SCTE from the DTE.

## Example

The following example prevents phase shifting of the data with respect to the clock:

```
interface serial 0
dce-terminal-timing enable
```

# delay

To set a delay value for an interface, use the **delay** interface configuration command. Use the **no delay** command to restore the default delay value.

**delay** *tens-of-microseconds*
**no delay**

## Syntax Description

| | |
|---|---|
| *tens-of-microseconds* | Integer that specifies the delay in tens of microseconds for an interface or network segment. |

## Default

Default delay values may be displayed with the EXEC command **show interfaces**.

## Command Mode

Interface configuration

## Example

The following example sets a 30,000-microsecond delay on serial interface 3:

```
interface serial 3
delay 30000
```

## Related Command

**show interfaces**

# description (controller)

To add a description to a T1 controller interface on a Cisco 7000 series router, use the **description** controller configuration command. Use the **no description** command to remove the description.

**description** *string*
**no description**

## Syntax Description

| | |
|---|---|
| *string* | Comment or a description to help you remember what is attached to the interface |

## Default

No description is added.

## Command Mode

Controller configuration

## Usage Guidelines

The **description** command is meant solely as a comment to be put in the configuration to help you remember what certain T1 controllers are used for. The description affects the MIP interfaces only and appears in the output of the **show controllers t1** and **write terminal** EXEC commands.

## Example

The following example shows how to add a description for a T1 controller on slot 4, port 1, channel group 0:

```
interface serial 4/1:0
description Fractional T1 line to Mountain View -- 128 Kb/s
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**show controllers t1**
**write terminal** †

# description (interface)

To add a description to an interface configuration, use the **description** interface configuration command. Use the **no description** command to remove the description.

**description** *string*
**no description**

## Syntax Description

| | |
|---|---|
| *string* | Comment or a description to help you remember what is attached to this interface. |

## Default

No description is added.

## Command Mode

Interface configuration

## Usage Guidelines

The **description** command is meant solely as a comment to be put in the configuration to help you remember what certain interfaces are used for. The description appears in the output of the following EXEC commands: **show configuration**, **show interfaces**, and **write terminal**.

## Example

The following example describes a 3174 controller on serial interface 0:

```
interface serial 0
description 3174 Controller for test lab
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**show configuration** †
**show interfaces**
**write terminal** †

# down-when-looped

To configure an interface to inform the system it is down when loopback is detected, use the **down-when-looped** interface configuration command.

**down-when-looped**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

This command is valid for HDLC or PPP encapsulation on serial and HSSI interfaces.

When an interface has a backup interface configured, it is often desirable that the backup interface be enabled when the primary interface is either down or in loopback. By default, the backup is only enabled if the primary interface is down. By using the **down-when-looped** command, the backup interface will also be enabled if the primary interface is in loopback.

If testing an interface with the loopback command, or by placing the DCE into loopback, **down-when-looped** should not be configured; otherwise, packets will not be transmitted out the interface that is being tested.

## Example

In the following example, interface serial 0 is configured for HDLC encapsulation. It is then configured to let the system know that it is down when in loopback mode.

```
interface serial0
encapsulation hdlc
down-when-looped
```

## Related Commands

**backup interface**
**loopback (interface)**

# dte-invert-txc

On the Cisco 4000 platform, you can specify the serial Network Interface Module timing signal configuration. When the board is operating as a DTE, the **dte-invert-txc** command inverts the TXC clock signal it gets from the DCE that the DTE uses to transmit data. Use the **no** form of this command if the DCE accepts SCTE from the DTE.

**dte-invert-txc**
**no dte-invert-txc**

## Syntax Description

This command has no arguments or keywords.

## Default

Off

## Command Mode

Interface configuration

## Usage Guidelines

Use this command if the DCE cannot receive SCTE from the DTE, the data is running at high speeds, and the transmission line is long. This prevents phase shifting of the data with respect to the clock.

If the DCE accepts SCTE from the DTE, use **no dte-invert-txc**.

## Example

The following example inverts the TXC on serial interface 0:

```
interface serial 0
dte-invert-txc
```

# early-token-release

To enable *early token release*, use the **early-token-release** interface configuration command. Once enabled, use the no form of this command to disable this feature.

**early-token-release**
**no early-token-release**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

Early token release is a method whereby the Token Ring interfaces can release the token back onto the ring immediately after transmitting, rather than waiting for the frame to return. This feature helps increase the total bandwidth of the Token Ring.

The CSC-C2CTR, CSC-R16 (or CSC-R16M), CSC-2R, and CSC-1R cards and the Token Ring Interface Processor (TRIP) on the Cisco 7000 all support early token release.

## Examples

The following example enables the use of early token release on Token Ring interface 1:

```
interface tokenring 1
early-token-release
```

On the Cisco 7000 series, to enable the use of early token release on your Token Ring interface processor in slot 4 on port 1, issue the following configuration commands:

```
interface tokenring 4/1
early-token-release
```

# encapsulation

To set the encapsulation method used by the interface, use the **encapsulation** interface configuration command.

**encapsulation** *encapsulation-type*

## Syntax Description

| | |
|---|---|
| *encapsulation-type* | Encapsulation type. See Table 6-4 for a list of supported encapsulation types. |

## Default

The default depends on the type of interface. For example, a synchronous serial interface defaults to HDLC.

## Command Mode

Interface configuration

## Usage Guidelines

In order to use SLIP or PPP, the router must be configured with an IP routing protocol or with the **ip host-routing** command. This configuration is done automatically if you are using old-style **slip address** commands. However, you must configure it manually if you configure SLIP or PPP via the **interface async** command.

**Table 6-4 Encapsulation Types**

| Keyword | Encapsulation Type |
|---|---|
| **atm-dxi** | Asynchronous Transfer Mode-Data Exchange Interface. |
| **frame-relay** | Frame Relay (for serial interface). |
| **hdlc** | High-Level Data Link Control (HDLC) protocol for serial interface. This encapsulation method provides the synchronous framing and error detection functions of HDLC without windowing or retransmission. |
| **lapb** | X.25 LAPB DTE operation (for serial interface). |
| **ppp** | Point-to-Point Protocol (PPP) (for serial interface). |
| **sdlc** | IBM serial SNA. |
| **sdlc-primary** | IBM serial SNA (for primary serial interface). |
| **sdlc-secondary** | IBM serial SNA (for secondary serial interface). |
| **smds** | Switched Multimegabit Data Services (SMDS) (for serial interface). |
| **snap** | IEEE 802.2 Ethernet media. This encapsulation is specified in RFC 1042 and allows Ethernet protocols to run on IEEE 802.2 media. |
| **stun** | Cisco Serial Tunnel (STUN) protocol functions (for serial interface). |
| **x25** | X.25 DTE operation (for serial interface). |

## Examples

The following example resets HDLC serial encapsulation on serial interface 1:

```
interface serial 1
encapsulation hdlc
```

The following example enables PPP encapsulation on serial interface 0:

```
interface serial 0
encapsulation ppp
```

## Related Commands

**keepalive**
**ppp**
**ppp authentication chap**
**slip**

# encapsulation atm-dxi

Use the **encapsulation atm-dxi** interface configuration command to enable ATM-DXI encapsulation. The **no encapsulation atm-dxi** command disables ATM-DXI.

**encapsulation atm-dxi**
**no encapsulation atm-dxi**

## Syntax Description

This command has no arguments or keywords.

## Default

HDLC

## Command Mode

Interface configuration

## Example

The following example configures ATM-DXI encapsulation on serial interface 1:

```
interface serial 1
encapsulation atm-dxi
```

## Related Command

**atm-dxi map**

# encapsulation lapb

To set the LAPB encapsulation method used by the interface, use the **encapsulation lapb** interface configuration command.

**encapsulation lapb** [**dte** | **dce**] [**multi** | *protocol*]

## Syntax Description

| | |
|---|---|
| **dte** | (Optional) DDN X.25 DTE operation (for serial interface). |
| **dce** | (Optional) DDN X.25 DCE operation (for serial interface). |
| **multi** | (Optional) Multi-protocol support. |
| *protocol* | (Optional) Protocol type. See Table 6-5 for a list of supported protocol types. |

## Default

DTE is the default operational type.

IP is the default protocol type.

## Command Mode

Interface configuration

## Usage Guidelines

In order to use a particular encapsulation, you must configure the router with that protocol type.

**Table 6-5 Encapsulation LAPB Protocol Types**

| Keyword | Protocol Type |
|---|---|
| **apollo** | Apollo domain. |
| **appletalk** | AppleTalk. |
| **clns** | ISO CLNS. |
| **decnet** | DECnet. |
| **ip** | IP. |
| **ipx** | Novell IPX. |
| **multi** | Multiprotocol operation. |
| **qllc** | QLLC protocol. |
| **snapshot** | Snapshot routing support. |
| **vines** | Banyan VINES. |
| **xns** | Xerox Network Services. |

## Example

The following example enables LAPB encapsulation on serial interface 0, using a default IP routing protocol:

```
interface serial 0
encapsulation lapb
```

# encapsulation x25

To set the X.25 encapsulation method used by the interface, use the **encapsulation x25** interface configuration command.

**encapsulation x25** [**bfe** | **ddn** | **ietf**]
**encapsulation x25 dce** [**ddn** | **ietf**]
**encapsulation x25 dte** [**bfe** | **ddn** | **ietf**]

## Syntax Description

| | |
|---|---|
| **dce** | (Optional) DDN X.25 DCE operation (for serial interface). |
| **dte** | (Optional) DDN X.25 DTE operation (for serial interface). |
| **bfe** | (Optional) Blacker Front End attachment encapsulation. |
| **ddn** | (Optional) Defense Data Network attachment encapsulation. |
| **ietf** | (Optional) IETF RFC-1356 encapsulation. |

## Default

IETF RFC-1356 is the default encapsulation.

## Command Mode

Interface configuration

## Usage Guidelines

In order to use a particular encapsulation, you must configure the router with that protocol type.

## Examples

The following example enables X.25 encapsulation on serial interface 0, using a default IETF encapsulation:

```
interface serial 0
encapsulation x25
```

The following example enables X.25 encapsulation on serial interface 0, using BFE encapsulation:

```
interface serial 0
encapsulation x25 dte bfe
```

# fddi burst-count

To allow the FCI card to preallocate buffers to handle bursty FDDI traffic (for example, NFS bursty traffic), use the **fddi burst-count** interface configuration command. Use the **no** form of this command to revert to the default value.

**fddi burst-count** *number*
**no fddi burst-count**

## Syntax Description

| | |
|---|---|
| *number* | Number of preallocated buffers in the range from 1 to 10. |

## Default

3 buffers

## Command Mode

Interface configuration

## Usage Guidelines

This command applies to the FCI card only. The microcode software version should *not* be 128.45 or 128.43.

## Example

The following example sets the number of buffers to 5:

```
interface fddi 0
fddi burst-count 5
```

# fddi c-min

To set the C-Min timer on the PCM, use the **fddi c-min** interface configuration command. Use the **no** form of this command to revert to the default value.

**fddi c-min** *microseconds*
**no fddi c-min**

## Syntax Description

| | |
|---|---|
| *microseconds* | Sets the timer value in microseconds. |

## Default

1600 microseconds

## Command Mode

Interface configuration

## Usage Guidelines

This command applies to the processor CMT only. You need extensive knowledge of the PCM state machine to tune this timer. Use this command when you run into PCM interoperability problems.

## Example

The following example sets the C-Min timer to 2000 microseconds:

```
interface fddi 0
fddi c-min 2000
```

## Related Commands

**fddi tb-min**
**fddi tl-min-time**
**fddi t-out**

# fddi cmt-signal-bits

To control the information transmitted during the connection management (CMT) signaling phase, use the **fddi cmt-signal-bits** interface configuration command.

**fddi cmt-signal-bits** *signal-bits* [**phy-a** | **phy-b**]

## Syntax Description

| | |
|---|---|
| *signal-bits* | A hexadecimal number preceded by 0x; for example, 0x208. The FDDI standard defines ten bits of signaling information that must be transmitted, as follows:<br><br>**bit 0—**Escape bit. Reserved for future assignment by the FDDI standards committee.<br>**bits 1 and 2**—Physical type, as defined in Table 6-6.<br>**bit 3**—Physical compatibility. Set if topology rules include the connection of a physical-to-physical type at the end of the connection.<br>**bits 4 and 5**—Link Confidence test duration; set as defined in Table 6-7.<br>**bit 6**—Media Access Control (MAC) available for link confidence test.<br>**bit 7**—Link confidence test failed. The setting of bit 7 indicates that the link confidence was failed by the Cisco end of the connection.<br>**bit 8—**MAC for local loop.<br>**bit 9**—MAC on physical output. |
| **phy-a** | (Optional) Selects Physical Sublayer A. |
| **phy-b** | (Optional) Selects Physical Sublayer B. |

## Default

The default signal bits for the **phy-a** and **phy-b** keywords are as follows:

- **phy-a** is set to 0x008 (hexadecimal) or 00 0000 1000 (binary). Bits 1 and 2 are set to 00 to select Physical A. Bit 3 is set to 1 to indicate “accept any connection.”
- **phy-b** is set to 0x20c (hexadecimal) or 10 0000 1100 (binary). Bits 1 and 2 are set to 10 to select Physical B. Bit 3 is set to 1 to indicate “accept any connection.” Bit 9 is set to 1 to select MAC on output. The normal data flow on FDDI is input on Physical A and output on Physical B.

## Command Mode

Interface configuration

## Usage Guidelines

If neither the **phy-a** nor **phy-b** keyword is specified, the signal bits apply to both physical connections.

**Note** Use of the **fddi cmt-signal-bits** configuration command is *not* recommended under normal operations. This command is used when debugging specific CMT implementation issues.

Use Table 6-6 and Table 6-7 to set the physical type and duration bits.

**Table 6-6 FDDI Physical Type Bit Specifications**

| Bit 2 | Bit 1 | Physical Type |
|---|---|---|
| 0 | 0 | Physical A |
| 1 | 0 | Physical B |
| 0 | 1 | Physical S |
| 1 | 1 | Physical M |

**Table 6-7 FDDI Link Confidence Test Duration Bit Specification**

| Bit 5 | Bit 4 | Test Duration |
|---|---|---|
| 0 | 0 | Short test (default 50 milliseconds) |
| 1 | 0 | Medium test (default 500 milliseconds) |
| 0 | 1 | Long test (default 5 seconds) |
| 1 | 1 | Extended test (default 50 seconds) |

## Example

The following example sets the CMT signaling phase to signal bits 0x208 on both physical connections:

```
interface fddi 0
fddi cmt-signal-bits 208
```

# fddi duplicate-address-check

To turn on the duplicate address detection capability on the FDDI, use the **fddi duplicate-address-check** interface configuration command. Use the **no** form of this command to disable this feature.

**fddi duplicate-address-check**
**no fddi duplicate-address-check**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

If you use this command, the router will detect a duplicate address if multiple stations are sharing the same MAC address. If the router finds a duplicate address, it will shut down the interface.

## Example

The following example enables duplicate address checking on the FDDI:

```
interface fddi 0
fddi duplicate-address-check
```

# fddi encapsulate

To specify encapsulating bridge mode on the CSC-C2/FCIT interface card, use the **fddi encapsulate** interface configuration command. Use the **no fddi encapsulate** command to turn off encapsulation bridging and return the FCIT interface to its translational, nonencapsulating mode.

**fddi encapsulate**
**no fddi encapsulate**

## Syntax Description

This command has no arguments or keywords.

## Default

The FDDI interface by default uses the SNAP encapsulation format defined in RFC 1042. It is not necessary to define an encapsulation method for this interface when using the CSC-FCI interface card.

## Command Mode

Interface configuration

## Usage Guidelines

The **no fddi encapsulate** command applies only to CSC-C2/FCIT interfaces, because the CSC-FCI interfaces are always in encapsulating bridge mode.The CSC-C2/FCIT interface card fully supports transparent and translational bridging for the following configurations:

- FDDI to FDDI
- FDDI to Ethernet
- FDDI to Token Ring

The command **fddi encapsulate** puts the CSC-C2/FCIT interface into encapsulation mode when doing bridging. In transparent mode, the FCIT interface interoperates with earlier versions of the CSC-FCI encapsulating interfaces when performing bridging functions on the same ring.

**Caution** Bridging between dissimilar media presents several problems that can prevent communications from occurring. These problems include b it-order translation (or usage of MAC addresses as data), maximum transfer unit (MTU) differences, frame status differences, and multicast address usage. Some or all of these problems may be present in a multimedia bridged LAN and preventing communication from taking place. These problems are most prevalent when bridging between Token Rings and Ethernets or between Token Rings and FDDI nets. This is because of the different way Token Ring is implemented by the end nodes.

The following protocols have problems when bridged between Token Ring and other media: Novell IPX, DECnet Phase IV, AppleTalk, VINES, XNS, and IP. Further, the following protocols may have problems when bridged between FDDI and other media: Novell IPX and XNS. We recommend that these protocols be routed whenever possible.

## Example

The following example sets FDDI interface 1 on the CSC-C2/FCIT interface card to encapsulating bridge mode:

```
interface fddi 1
fddi encapsulate
```

# fddi smt-frames

To enable the SMT frame processing capability on the FDDI, use the **fddi smt-frames** interface configuration command. Use the **no** form of this command to disable this feature, in which case the router will not generate or respond to SMT frames.

**fddi smt-frames**
**no fddi smt-frames**

## Syntax Description

This command has no arguments or keywords.

## Default

Enabled

## Command Mode

Interface configuration

## Usage Guidelines

Use the **no** form of this command to turn off SMT frame processing for diagnosing purposes. Use the **fddi smt-frames** command to reenable the feature.

## Example

The following example disables SMT frame processing:

```
interface fddi 0
no fddi smt-frames
```

# fddi t-out

To set the t-out timer in the physical connection management (PCM), use the **fddi t-out** interface configuration command. Use the **no** form of this command to revert to the default value.

**fddi t-out** *milliseconds*
**no fddi t-out**

## Syntax Description

| | |
|---|---|
| *milliseconds* | Sets the timeout timer. |

## Default

100 milliseconds

## Command Mode

Interface configuration

## Usage Guidelines

This command applies to the processor CMT only. You need extensive knowledge of the PCM state machine to tune this timer. Use this command when you run into PCM interoperability problems.

## Example

The following example sets the timeout timer to 200 milliseconds:

```
interface fddi 0
fddi t-out 200
```

## Related Commands

**fddi c-min**
**fddi tb-min**
**fddi tl-min-time**

# fddi tb-min

To set the TB-Min timer in the physical connection management (PCM), use the **fddi tb-min** interface configuration command. Use the **no** form of this command to revert to the default value.

**fddi tb-min** *milliseconds*
**no fddi tb-min**

## Syntax Description

| | |
|---|---|
| *milliseconds* | Sets the TB-Min timer value in milliseconds. |

## Default

100 milliseconds

## Command Mode

Interface configuration

## Usage Guidelines

This command applies to the processor CMT only. You need extensive knowledge of the PCM state machine to tune this timer. Use this command when you run into PCM interoperability problems.

## Example

The following example sets the TB-Min timer to 200 milliseconds:

```
interface fddi 0
fddi tb-min 200
```

## Related Commands

**fddi c-min**
**fddi tl-min-time**
**fddi t-out**

# fddi tl-min-time

To control the TL-Min time (the minimum time to transmit a Physical Sublayer, or PHY line state, before advancing to the next physical connection management (PCM) state, as defined by the X3T9.5 specification), use the **fddi tl-min-time** interface configuration command.

**fddi tl-min-time** *microseconds*

## Syntax Description

| | |
|---|---|
| *microseconds* | Integer that specifies the time used during the connection management (CMT) phase to ensure that signals are maintained for at least the value of TL-Min so the remote station can acquire the signal. |

## Default

30 microseconds

## Command Mode

Interface configuration

## Usage Guidelines

Interoperability tests have shown that some implementations of the FDDI standard need more than 30 microseconds to sense a signal.

## Examples

The following example changes the TL-Min time from 30 microseconds to 100 microseconds:

```
interface fddi 0
fddi tl-min-time 100
```

The following example changes the TL-Min time from 30 microseconds to 100 microseconds on a Cisco 7000:

```
interface fddi 3/0
fddi tl-min-time 100
```

## Related Commands

**fddi c-min**
**fddi tl-min-time**
**fddi t-out**

# fddi token-rotation-time

To control ring scheduling during normal operation and to detect and recover from serious ring error situations, use the **fddi token-rotation-time** interface configuration command.

**fddi token-rotation-time** *microseconds*

## Syntax Description

| | |
|---|---|
| *microseconds* | Integer that specifies the token rotation time (TRT). |

## Default

5000 microseconds

## Command Mode

Interface configuration

## Usage Guidelines

The FDDI standard restricts the allowed time to be greater than 4000 microseconds and less than 165,000 microseconds. As defined in the X3T9.5 specification, the value remaining in the TRT is loaded into the token holding timer (THT). Combining the values of these two timers provides the means to determine the amount of bandwidth available for subsequent transmissions.

## Examples

The following example sets the rotation time to 24,000 microseconds:

```
interface fddi 0
fddi token-rotation-time 24000
```

The following example sets the rotation time to 24,000 microseconds on a Cisco 7000:

```
interface fddi 3/0
fddi token-rotation-time 24000
```

# fddi valid-transmission-time

To recover from a transient ring error, use the **fddi valid-transmission-time** interface configuration command.

**fddi valid-transmission-time** *microseconds*

## Syntax Description

| | |
|---|---|
| *microseconds* | Integer that specifies the transmission valid timer (TVX) interval. |

## Default

2500 microseconds

## Command Mode

Interface configuration

## Examples

The following example changes the transmission timer interval to 3000 microseconds:

```
interface fddi 0
fddi valid-transmission-time 3000
```

The following example changes the transmission timer interval to 3000 microseconds on a Cisco 7000:

```
interface fddi 3/0
fddi valid-transmission-time 3000
```

# framing

Use the **framing** controller configuration command to select the frame type for the T1 or E1 data line.

**framing** {**sf** | **esf** | **crc4** | **no-crc4**}

## Syntax Description

| | |
|---|---|
| **sf** | Specifies super frame as the T1 frame type. |
| **esf** | Specifies extended super frame as the T1 frame type. |
| **crc4** | Specifies CRC4 frame as the E1 frame type. |
| **no-crc4** | Specifies no CRC4 frame as the E1 frame type. |

## Default

Super frame is the default on a T1 line.

CRC4 frame is the defalut on an E1 line.

## Command Mode

Controller configuration

## Usage Guidelines

Use this command in configurations where the router is intended to communicate with T1 or E1 fractional data line. The service provider determines which framing type, either **sf**, **esf**, or **crc4** is required for your T1/E1 circuit.

## Example

The following example selects extended super frame as the T1 frame type:

```
framing esf
```

## Related Commands

**channel-group**
**linecode**

# hold-queue

To specify the hold-queue limit of an interface, use the **hold-queue** interface configuration command. Use the **no hold-queue** command with the appropriate keyword to restore the default values for an interface.

**hold-queue** *length* {**in** | **out**}
**no hold-queue** {**in** | **out**}

## Syntax Description

| | |
|---|---|
| *length* | Integer that specifies the maximum number of packets in the queue. |
| **in** | Specifies the input queue. |
| **out** | Specifies the output queue. |

## Default

The default input hold-queue limit is 75 packets. The default output hold-queue limit is 40 packets. These limits prevent a malfunctioning interface from consuming an excessive amount of memory. There is no fixed upper limit to a queue size.

## Command Mode

Interface configuration

## Usage Guidelines

The input hold queue prevents a single interface from flooding the network server with too many input packets. Further input packets are discarded if the interface has too many input packets outstanding in the system.

If priority output queueing is being used, the length of the four output queues is set using the **priority-list** global configuration command. The **hold-queue** command cannot be used to set an output hold queue length in this situation.

For slow links, use a small output hold-queue limit. This approach prevents storing packets at a rate that exceeds the transmission capability of the link. For fast links, use a large output hold-queue limit. A fast link may be busy for a short time (and thus require the hold queue), but can empty the output hold queue quickly when capacity returns.

To display the current hold queue setting and the number of packets discarded because of hold queue overflows, use the EXEC command **show interfaces**.

---

**Note** Increasing the hold queue can have detrimental effects on network routing and response times. For protocols that use seq/ack packets to determine round trip times, do not increase the output queue. Dropping packets instead informs hosts to slow down transmissions to match available bandwidth. This is generally better than having duplicate copies of the same packet within the network (which can happen with large hold queues).

---

## Example

The following example illustrates how to set a small input queue on a slow serial line:

```
interface serial 0
hold-queue 30 in
```

## Related Command

**show interfaces**

# hssi external-loop-request

To allow the router to support a CSU/DSU that uses the LC signal to request a loopback from the router, use the **hssi external-loop-request** interface configuration command. Use the **no** form of this command to disable the feature.

**hssi external-loop-request**
**no hssi external-loop-request**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

The HSA applique (on the HSSI) contains an LED that indicates the LA, LB, and LC signals transiting through the devices. The CSU/DSU uses the LC signal to request a loopback from the router. The CSU/DSU may want to do this so that its own network management diagnostics can independently check the integrity of the connection between the CSU/DSU and the router.

Use this command to enable a two-way, internal, and external loopback request on HSSI from the CSU/DSU.

---

**Note** If your CSU/DSU does not support this feature, it should not be enabled in the router. Not enabling this feature prevents spurious line noise from accidentally tripping the external loopback request line, which would interrupt the normal data flow.

---

## Example

The following example enables a CSU/DSU to use the LC signal to request a loopback from the router:

```
hssi external-loop-request
```

# hssi internal-clock

To convert the HSSI interface into a 45 MHz clock master, use the **hssi internal-clock** interface configuration command. Use the **no** form of this command to disable the clock master mode.

**hssi internal-clock**
**no hssi internal-clock**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

Use this command in conjunction with the HSSI null-modem cable to connect two Cisco routers together with HSSI. You must configure this command at both ends of the link, not just one.

## Example

The following example converts the HSSI interface into a 45 MHz clock master:

```
hssi internal-clock
```

# hub

To enable and configure a port on an Ethernet hub of a Cisco 2505 or Cisco 2507, use the **hub** global configuration command.

**hub ethernet** *number port* [*end-port*]

## Syntax Description

| | |
|---|---|
| **ethernet** | Indicates that the hub is in front of an Ethernet interface. |
| *number* | Hub number, starting with 0. Since there is currently only one hub, this number is 0. |
| *port* | Port number on the hub. On the Cisco 2505, port numbers range from 1 through 8. On the Cisco 2507, port numbers range from 1 through 16. If a second port number follows, then the first port number indicates the beginning of a port range. |
| *end-port* | (Optional) Last port number of a range. |

## Default

No hub ports are configured.

## Command Mode

Global configuration

## Examples

The following example enables port 1 on hub 0:

```
hub ethernet 0 1
no shutdown
```

The following example enables ports 1 through 8 on hub 0:

```
hub ethernet 0 1 8
no shutdown
```

## Related Command

**shutdown**

# ignore-dcd

Use the **ignore-dcd** interface configuration command to configure the serial interface to monitor the DSR signal (instead of the DCD signal) as the line up/down indicator. Use the **no** form of this command to restore the default behavior.

**ignore-dcd**
**no ignore-dcd**

## Syntax Description

This command has no arguments or keywords.

## Default

The serial interface, operating in DTE mode, monitors the DCD signal as the line up/down indicator.

## Command Mode

Interface configuration

## Usage Guidelines

This command applies to Quad Serial NIM interfaces on the Cisco 4000 series and Hitachi-based serial interfaces on the Cisco 2500 series and Cisco 3000 series.

When the serial interface is operating in DTE mode, it monitors the Data Carrier Detect (DCD) signal as the line up/down indicator. By default, the attached DCE device sends the DCD signal. When the DTE interface detects the DCD signal, it changes the state of the interface to up.

In some configurations, such as an SDLC multidrop environment, the DCE device sends the Data Set Ready (DSR) signal instead of the DCD signal, which prevents the interface from coming up. Use this command to tell the interface to monitor the DSR signal instead of the DCD signal as the line up/down indicator.

## Example

The following example configures serial interface 0 to monitor the DSR signal as the line up/down indicator:

```
interface serial 0
ignore-dcd
```

# interface

To configure an interface type and enter interface configuration mode, use the **interface** global configuration command.

**interface** *type number*
**interface** *type slot*/*port* (for the Cisco 7000 series)
**interface serial** *slot*/*port*:*channel-group* (for channelized T1 or E1 on the Cisco 7000)
**interface serial** *number*:*channel-group* (for channelized T1 or E1 on the Cisco 4000)

To configure a subinterface, use the **interface** global configuration command.

**interface** *type number***.***subinterface-number* [**multipoint | point-to-point**]
**interface** *type slot*/*port***.***subinterface-number* [**multipoint | point-to-point**] (for the Cisco 7000 series)

## Syntax Description

| | |
|---|---|
| *type* | Type of interface to be configured. See Table 6-8. |
| *number* | Port, connector, or interface card number. On a Cisco 4000 series router, specifies the NIM number. The numbers are assigned at the factory at the time of installation or when added to a system, and can be displayed with the **show interfaces** command. |
| *slot* | On the Cisco 7000 series, specifies the backplane slot number. On the 7000, value can be **0**, **1**, **2**, **3**, or **4**. On the 7010, value can be **0**, **1**, or **2**. The slots are numbered from left to right. |
| /*port* | On the Cisco 7000 series, specifies the port number of the interface. It can be **0**, **1**, **2**, **3**, **4**, **5**, **6**, or **7** depending on the type of interface, as follows:<br>• AIP (ATM Interface Processor) **0**<br>• EIP (Ethernet Interface Processor) **0**, **1**, **2**, **3**, **4**, or **5**<br>• FIP (FDDI Interface Processor) **0**<br>• FSIP (Fast Serial Interface Processor) **0**, **1**, **2**, **3, 4, 5, 6, or 7**<br>• HIP (HSSI Interface Processor) **0**<br>• MIP (MultiChannel Interface Processor) **0** or **1**<br>• TRIP (Token Ring Interface Processor) **0**, **1**, **2**, or **3**<br>• Ports on each interface processor are numbered from the top down. |
| **:***channel-group* | On the Cisco 7000 series on a MIP/CxCT1 card, specifies the T1 channel group number in the range of 0 to 23 defined with the **channel-group** controller configuration command. |
| **.***subinterface-number* | Subinterface number in the range 1 to 4294967293. The number that precedes the period (.) must match the number this subinterface belongs to. |
| **multipoint \| point-to-point** | (Optional) Specifies a multipoint or point-to-point subinterface. The default is **multipoint**. |

**Table 6-8 Interface Type Keywords**

| Keyword | Interface Type |
|---|---|
| **async** | Auxiliary port line used as an asynchronous interface. |
| **atm** | ATM interface. |
| **bri** | Integrated Services Digital Network (ISDN) Basic Rate Interface (BRI). This interface configuration is propagated to each of the B channels. B channels cannot be individually configured. The interface must be configured with dial-on-demand commands in order for calls to be placed on that interface. |
| **dialer** | Dialer interface. |
| **ethernet** | Ethernet IEEE 802.3 interface. |
| **fddi** | Fiber Distributed Data Interface (FDDI). |
| **hssi** | High-Speed Serial Interface (HSSI). |
| **lex** | LAN Extender (LEX) interface. |
| **loopback** | Software-only loopback interface that emulates an interface that is always up. It is a virtual interface supported on all platforms. The *interface-number* is the number of the loopback interface that you want to create or configure.There is no limit on the number of loopback interfaces you can create. |
| **null** | Null interface. |
| **serial** | Serial interface. |
| **tokenring** | Token Ring interface. |
| **tunnel** | Tunnel interface; a virtual interface. The *number* is the number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create. |

## Default

The default mode for subinterfaces is **multipoint**.

## Command Mode

Global configuration

## Usage Guidelines

Subinterfaces can be configured to support partially meshed Frame Relay networks (refer to the chapter entitled "Configuring Interfaces" in the *Router Products Configuration Guide*).

There is no correlation between the number of the physical serial interface and the number of the logical LAN Extender interface. These interfaces can have the same or different numbers.

## Examples

In the following example, serial interface 0 is configured with PPP encapsulation:

```
interface serial 0
encapsulation ppp
```

The following example enables loopback mode and assigns an IP network address and network mask to the interface. The loopback interface established here will always appear to be up:

```
interface loopback 0
ip address 131.108.1.1 255.255.255.0
```

The following example for the Cisco 7000 shows the interface configuration command for Ethernet port 4 on the EIP that is installed in (or recently removed from) slot 2:

```
interface ethernet 2/4
```

The following example begins configuration on the Token Ring interface processor in slot 1 on port 0 of a Cisco 7000:

```
interface tokenring 1/0
```

The following example shows how a partially meshed Frame Relay network can be configured. In this example, subinterface serial 0.1 is configured as a multipoint subinterface with three frame relay PVCs associated, and subinterface serial 0.2 is configured as a point-to-point subinterface.

```
interface serial 0
encapsulation frame-relay
interface serial 0.1 multipoint
ip address 131.108.10.1 255.255.255.0
frame-relay interface-dlci 42 broadcast
frame-relay interface-dlci 53 broadcast
interface serial 0.2 point-to-point
ip address 131.108.11.1 255.255.0
frame-relay interface-dlci 59 broadcast
```

The following example configures circuit 0 of a T1 link for Point-to-Point Protocol (PPP) encapsulation:

```
controller t1 4/1
circuit 0 1
interface serial 4/1:0
ip address 131.108.13.1 255.255.255.0
encapsulation ppp
```

The following example configures LAN Extender interface 0:

```
interface lex 0
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**circuit**
**controller**
**mac-address** †
**ppp**
**show interfaces**
**slip**

# invert-transmit-clock

Delays between the SCTE clock and data transmission indicate that the transmit clock signal might not be appropriate for the interface rate and length of cable being used. Different ends of the wire may have variances that differ slightly. To invert the clock signal to compensate for these factors, use the **invert-transmit-clock** interface configuration command. This command applies only to the Cisco 7000 series.

**invert-transmit-clock**
**no invert-transmit-clock**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Interface configuration

## Example

In the following example, the clock signal on serial interface 3/0 is inverted.

```
interface serial 3/0
invert-transmit-clock
```

# ip address-pool

To make temporary IP addresses available for dial-in asynchronous clients using Serial Line Internet Protocol (SLIP)/PPP, use the **ip address-pool** global configuration command. Use the **no** form of the command to disable IP address pooling on all interfaces.

**ip address-pool dhcp-proxy-client**
**no ip address-pool dhcp-proxy-client**

## Syntax Description

This command has no arguments or keywords.

## Default

IP address pooling is not enabled.

## Command Mode

Global configuration

## Usage Guidelines

The **ip address-pool** command allows you to use a router as the intermediary (a proxy-client) between a third-party Dynamic Host Configuration Protocol (DHCP) server and clients dialing in to the router on asynchronous interfaces. If this command is issued and no DHCP servers have been defined using the **ip dhcp server** command, the router will use the limited address of 255.255.255.255 to communicate with available DHCP servers on the network.

A DHCP server temporarily allocates network addresses to clients through the router on an as-needed basis. While the client is active, the address is automatically renewed in a minimum of 20-minute increments. When the user terminates the session, the interface connection is terminated so that network resources can be quickly reused.

In normal situations, if a SLIP/PPP session fails (for example if a modem line disconnects), the allocated address is temporarily reserved so that client can receive the same IP address when it dials back into the rotuer. This way, a session that was accidentally terminated can be resumed.

The **ip address-pool** command initializes proxy-client status to all interfaces on the router defined as asynchronous. To selectively disable proxy-client status on a single asynchronous interface, use the **no peer default ip address pool** interface command.

## Example

The following example enables DHCP proxy-client status on all asynchronous interfaces on the router:

```
ip address-pool dhcp-proxy-client
```

## Related Commands

**ip dhcp-server**
**peer default ip address pool**
**show dhcp**
**interface async**
**encapsulation**
**ppp**

# ip dhcp-server

To specify which Dynamic Host Configuration Protocol (DHCP) servers to use on your network, specify the IP address of one or more DHCP servers available on the network by using the **ip dhcp-server** global configuration command. Use the **no** form of the command to remove a DHCP server's IP address.

**ip dhcp-server** [*ip-address* | *name*]
**no ip dhcp-server** [*ip-address* | *name*]

## Syntax Description

| | |
|---|---|
| *ip-address* | (Optional) IP address of a DHCP server. You can specify up to 10 servers on the network. |
| *name* | (Optional) Name of a DHCP server. You can specify up to 10 servers on the network. |

## Default

The IP limited broadcast address of 255.255.255.255 is used for transactions if no DHCP server is specified. Use of this address provides allows automatic detection of DHCP servers.

## Command Mode

Global configuration

## Usage Guidelines

By default, the DHCP proxy-client feature uses the IP address of 255.255.255.255 to discover and interact with DHCP servers. If you wish to specify which DHCP servers are used on your network, use the **ip dhcp-server** command to define up to four specific DHCP servers. To use the DHCP proxy-client feature, enable your router to be a proxy-client on asynchronous interfaces by using the **ip address-pool dhcp-proxy-client** command.

---

**Note** To facilitate transmission, configure intermediary routers to use an ip helper address whenever the DHCP server is not on the local LAN and the router is using broadcasts to interact with the DHCP server. See "Configuring IP" in the *Router Products Configuration Guide Addendum*..

---

## Example

The following command specifies a DHCP server with the IP address of 129.12.13.81:

```
ip dhcp-server 129.12.13.81
```

## Related Commands

**ip address-pool dhcp-proxy-client**
**ip helper address**
**peer default ip address pool**
**show dhcp**

# keepalive

Use the **keepalive** interface configuration command to set the keepalive timer for a specific interface. The **no keepalive** command turns off keepalives entirely.

**keepalive** [*seconds*]
**no keepalive** [*seconds*]

## Syntax Description

| | |
|---|---|
| *seconds* | (Optional) Unsigned integer value greater than 0. The default is 10 seconds. |

## Default

10 seconds

## Command Mode

Interface configuration

## Usage Guidelines

You can configure the keepalive interval, which is the frequency at which the router sends messages to itself (Ethernet and Token Ring) or to the other end (serial), to ensure a network interface is alive. The interval in previous software versions was 10 seconds; it is now adjustable in 1-second increments down to 1 second. An interface is declared down after three update intervals have passed without receiving a keepalive packet.

Setting the keepalive timer to a low value is very useful for rapidly detecting Ethernet interface failures (transceiver cable disconnecting, cable unterminated, and so on).

A typical serial line failure involves losing Carrier Detect (CD). Since this sort of failure is typically noticed within a few milliseconds, adjusting the keepalive timer for quicker routing recovery is generally not useful.

---

**Note** When adjusting the keepalive timer for a very low bandwidth serial interface, large datagrams can delay the smaller keepalive packets long enough to cause the line protocol to go down. You may need to experiment to determine the best value.

---

## Example

The following example sets the keepalive interval to 3 seconds:

```
interface ethernet 0
keepalive 3
```

# lex burned-in-address

To set the burned-in MAC address for a LAN Extender interface, use the **lex burned-in-address** interface configuration command. To clear the burned-in MAC address, use the **no** form of this command.

**lex burned-in-address** *ieee-address*
**no lex burned-in-address**

## Syntax Description

| | |
|---|---|
| *ieee-address* | 48-bit IEEE MAC address written as a dotted triplet of four-digit hexadecimal numbers. |

## Default

No burned-in MAC address is set

## Command Mode

Interface configuration

## Usage Guidelines

Use this command only on a LAN Extender interface that is not currently active (not bound to a serial interface).

## Example

The following example sets the burned-in MAC address on LAN Extender interface 0:

```
interface serial 4
encapsulation ppp
interface lex 0
lex burned-in-address 0000.0c00.0001
ip address 131.108.172.21 255.255.255.0
```

# lex input-address-list

To assign an access list that filters on MAC addresses, use the **lex input-address-list** interface configuration command. To remove an access list from the interface, use the **no** form of this command.

**lex input-address-list** *access-list-number*
**no lex input-address-list**

## Syntax Description

| | |
|---|---|
| *access-list-number* | Number of the access list you assigned with the **access-list** global configuration command. It can be a number from 700 to 799. |

## Default

No access lists are preassigned to a LAN Extender interface.

## Command Mode

Interface configuration

## Usage Guidelines

Use the **lex input-address-list** command to filter the packets that are allowed to pass from the LAN Extender to the core router. The access list filters packets based on the source MAC address.

The LAN Extender interface does not process MAC-address masks. Therefore, you should omit the mask from the **access-list** commands.

For LAN Extender interfaces, an implicit permit everything entry is automatically defined at the end of an access list. Note that this behavior differs from other router access lists, which have an implicit deny everything entry at the end of each access list.

## Example

The following example applies access list 710 to LAN Extender interface 0. This access list denies all packets from MAC address 0800.0214.2776 and permits all other packets.

```
access-list 710 deny 0800.0214.2776
interface lex 0
lex input-address-list 710
```

## Related Command

A dagger (†) indicates that the command is documented in another chapter.

**access-list** †

# lex input-type-list

To assign an access list that filters Ethernet packets by type code, use the **lex input-type-list** interface configuration command. To remove an access list from the interface, use the **no** form of this command.

**lex input-type-list** *access-list-number*
**no lex input-type-list**

## Syntax Description

| | |
|---|---|
| *access-list-number* | Number of the access list you assigned with the **access-list** global configuration command. It can be a number in the range 200 to 299. |

## Default

No access lists are preassigned to a LAN Extender interface.

## Command Mode

Interface configuration

## Usage Guidelines

Filtering is done on the LAN Extender chassis.

The LAN Extender interface does not process masks. Therefore, you should omit the mask from the **access-list** commands.

For LAN Extender interfaces, an implicit permit everything entry is automatically defined at the end of an access list. Note that this behavior differs from other router access lists, which have an implicit deny everything entry at the end of each access list.

## Example

The following example applies access list 220 to LAN Extender interface 0. This access list denies all AppleTalk packets (packets with a type field of 0x809B) and permits all other packets.

```
access-list 220 deny 0x809B 0x0000
interface lex 0
lex input-type-list 220
```

## Related Command

A dagger (†) indicates that the command is documented in another chapter.

**access-list** †

# lex priority-group

To activate priority output queuing on the LAN Extender, use the **lex priority-group** interface configuration command. To disable priority output queuing, use the **no** form of this command.

**lex priority-group** *group*
**no lex priority-group**

## Syntax Description

| | |
|---|---|
| *group* | Number of the priority group. It can be a number in the range 1 to 10. |

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

To define queuing priorities, use the **priority-list protocol** global configuration command. Note that you can use only the following forms of this command:

**priority-list** *list* **protocol** *protocol* {**high** | **medium** | **normal** | **low**}

**priority-list** *list* **protocol bridge** {**high** | **medium** | **normal** | **low**} **list** *list-number*

If you specify a protocol that does not have an assigned Ethernet type code, such as **x25**, **stun**, or **pad**, it is ignored and will not participate in priority output queuing.

## Example

The following example activates priority output queuing on LAN Extender interface 0:

```
priority-list 5 protocol bridge medium list 701
lex interface 0
lex priority-group 5
```

## Related Command

A dagger (†) indicates that the command is documented in another chapter.

**priority-list protocol** †

# lex retry-count

To define the number of times to resend commands to the LAN Extender chassis, use the **lex retry-count** interface configuration command. To return to the default value, use the **no** form of this command.

**lex retry-count** *number*
**no lex retry-count** [*number*]

## Syntax Description

| | |
|---|---|
| *number* | Number of times to retry sending commands to the LAN Extender. It can be a number in the range 0 to 100. The default is 10 times. |

## Default

10

## Command Mode

Interface configuration

## Usage Guidelines

After the core router has sent a command the specified number of times without receiving an acknowledgment from the LAN Extender, it stops sending the command altogether.

## Example

The following example resends commands 20 times to the LAN Extender:

```
lex interface 0
lex retry-count 20
```

## Related Command

**lex timeout**

# lex timeout

To define the amount of time to wait for a response from the LAN Extender, use the **lex timeout** interface configuration command. To return to the default time, use the **no** form of this command.

**lex timeout** *milliseconds*
**no lex timeout** [*milliseconds*]

## Syntax Description

| | |
|---|---|
| *milliseconds* | Time, in milliseconds, to wait for a response from the LAN Extender before resending the command. It can be a number in the range 500 to 60000. The default is 2000 milliseconds (2 seconds). |

## Default

2000 milliseconds (2 seconds)

## Command Mode

Interface configuration

## Usage Guidelines

The **lex timeout** command defines the amount of time that the core router will wait to receive an acknowledgment after having sent a command to the LAN Extender.

## Example

The following example causes unacknowledged packets to be resent at 4-second intervals:

```
lex interface 0
lex timeout 4000
```

## Related Command

**lex retry-count**

# linecode

Use the **linecode** controller configuration command to select the line-code type for the T1 or E1 line.

**linecode** {**ami** | **b8zs** | **hdb3**}

## Syntax Description

| | |
|---|---|
| **ami** | Specifies alternate mark inversion (AMI) as the line-code type. Valid for T1 or E1 controllers. |
| **b8zs** | Specifies B8ZS as the line-code type. Valid for T1 controller only. |
| **hdb3** | Specifies high-density bipolar 3 (hdb3) as the line-code type. Valid for E1 controller only. |

## Default

AMI is the default for T1 lines.

High-density bipolar 3 is the default for E1 lines.

## Command Mode

Controller configuration

## Usage Guidelines

Use this command in configurations where the router is intended to communicate with T1 fractional data line. The T1 service provider determines which line-code type, either **ami** or **b8zs**, is required for your T1 circuit. Likewise, the E1 service provider determines which line-code type, either **ami** or **hdb3**, is required for your E1 circuit

## Example

The following example specifies B8ZS as the line-code type:

```
linecode b8zs
```

# link-test

To re-enable the link-test function on a port on an Ethernet hub of a Cisco 2505 or Cisco 2507, use the **link-test** hub configuration command. Disable this feature if a pre-10BaseT twisted-pair device not implementing link test is connected to the hub port with the **no** form of this command.

**link-test**
**no link-test**

## Syntax Description

This command has no arguments or keywords.

## Default

Enabled

## Command Mode

Hub configuration

## Usage Guidelines

This command applies to a port on an Ethernet hub only. Disable this feature if a 10BaseT twisted-pair device at the other end of the hub does not implement the link test function.

## Example

The following example disables the link test function on hub 0, ports 1 through 3:

```
hub ethernet 0 1 3
no link-test
```

## Related Command

**hub**

# local-lnm

To enable Lanoptics Hub Networking Management of a PCbus Token Ring interface, use the **local-lnm** interface configuration command. Use the **no** form of this command to disable Lanoptics Hub Networking Management.

**local-lnm**
**no local-lnm**

## Syntax Description

This command has no arguments or keywords.

## Default

Management is not enabled.

## Command Mode

Interface configuration

## Usage Guidelines

The Token Ring interface on the AccessPro PC card can be managed by a remote LAN manager over the PCbus interface. At present, the Lanoptics Hub Networking Management software running on an IBM compatible PC is supported.

## Example

The following example enables Lanoptics Hub Networking Management:

```
local-lnm
```

# loopback (controller)

To loop an entire E1 line (including all channel-groups defined on the controller) toward the line and back toward the router, use the **loopback** controller configuration command. To remove the loop, use the **no** form of this command.

**loopback**
**no loopback**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Controller configuration

## Usage Guidelines

This command is useful for testing the DCE device (CSU/DSU) itself.

To show interfaces currently in loopback operation, use the **show interfaces loopback** EXEC command.

## Example

The following example configures the loopback test on the E1 line:

```
controller e1 0
loopback
```

# loopback (interface)

To diagnose equipment malfunctions between interface and device, use the **loopback** interface configuration command. The **no loopback** command disables the test.

**loopback**
**no loopback**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

On HSSI serial interface cards, the loopback function configures a two-way internal and external loop on the HSA applique of the specific interface.

On MCI and SCI serial interface cards, the loopback functions when a CSU/DSU or equivalent device is attached to the router. The **loopback** command loops the packets through the CSU/DSU to configure a CSU loop, when the device supports this feature.

On the MCI and MEC Ethernet cards, the interface receives back every packet it sends when the **loopback** command is enabled. Loopback operation has the additional effect of disconnecting network server functionality from the network.

On the CSC-FCI FDDI card, the interface receives back every packet it sends when the **loopback** command is enabled. Loopback operation has the additional effect of disconnecting network server functionality from the network.

On all Token Ring interface cards (except the 4-megabit CSC-R card), the interface receives back every packet it sends when the **loopback** command is enabled. Loopback operation has the additional effect of disconnecting network server functionality from the network.

---

**Note** Loopback does not work on an X.21 DTE because the X.21 interface definition does not include a loopback definition.

---

To show interfaces currently in loopback operation, use the **show interfaces loopback** EXEC command.

## Example

The following example configures the loopback test on Ethernet interface 4:

```
interface ethernet 4
loopback
```

## Related Commands

**down-when-looped**
**show interfaces loopback**

# loopback applique

To configure an internal loop on the HSSI applique, use the **loopback** interface configuration command. To remove the loop, use the **no** form of this command.

**loopback applique**
**no loopback applique**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

This command loops the packets within the applique, thus providing a way to test for communication within the router. It is useful for sending pings to yourself to check functionality of the applique.

To show interfaces currently in loopback operation, use the **show interfaces loopback** EXEC command.

## Example

The following example configures the loopback test on the HSSI applique:

```
interface serial 1
loopback applique
```

## Related Command

**show interfaces loopback**

# loopback dte

To loop packets to DTE internally within the CSU/DSU at the DTE interface, when the device supports this feature, use the **loopback** interface configuration command. To remove the loop, use the **no** form of this command.

**loopback dte**
**no loopback dte**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

This command is useful for testing the DTE-to-DCE cable.

To show interfaces currently in loopback operation, use the **show interfaces loopback** EXEC command.

## Example

The following example configures the loopback test on the DTE interface:

```
interface serial 1
loopback dte
```

## Related Command

**show interfaces loopback**

# loopback line

To loop packets completely through the CSU/DSU to configure the CSU loop, when the device supports this feature, use the **loopback line** interface configuration command. To remove the loop, use the **no** form of this command.

**loopback line**
**no loopback line**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

This command is useful for testing the DCE device (CSU/DSU) itself.

To show interfaces currently in loopback operation, use the **show interfaces loopback** EXEC command.

## Example

The following example configures the loopback test on the DCE device:

```
interface serial 1
loopback line
```

## Related Command

**show interfaces loopback**

# loopback local (controller)

To loop an entire T1 line (including all channel-groups defined on the controller) toward the line and back toward the router, use the **loopback local** controller configuration command. To remove the loop, use the **no** form of this command.

**loopback local**
**no loopback local**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Controller configuration

## Usage Guidelines

This command is useful for testing the DCE device (CSU/DSU) itself.

To show interfaces currently in loopback operation, use the **show interfaces loopback** EXEC command.

## Example

The following example configures the loopback test on the T1 line:

```
controller t1 0
loopback local
```

# loopback local (interface)

To loop a channelized T1 or channelized E1 channel-group, use the **loopback local** interface configuration command. To remove the loop, use the **no** form of this command.

**loopback local**
**no loopback local**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

This command is useful to loop a single channel-group in a channelized environment without disrupting the other channel-groups.

To show interfaces currently in loopback operation, use the **show interfaces loopback** EXEC command.

## Example

The following example configures the loopback test on the T1 line:

```
interface serial 1/0:22
loopback local
```

## Related Command

**show interfaces loopback**

# loopback remote (controller)

To loop packets from a MIP through the CSU/DSU, over a dedicated T1 link, to the remote CSU at the single destination for this T1 link and back, use the **loopback remote** controller configuration command. To remove the loop, use the **no** form of this command.

**loopback remote**
**no loopback remote**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Controller configuration

## Usage Guidelines

This command applies only when the device supports the remote function. It is used for testing the data communication channels.

For MIP cards, this controller configuration command applies if *only one* destination exists at the remote end of the cloud, the entire T1 line is dedicated to it, and the device at the remote end is a CSU (not a CSU/DSU). This is an uncommon case; MIPs are not usually used in this way.

To show interfaces currently in loopback operation, use the **show interfaces loopback** EXEC command.

## Example

The following example configures a remote loopback test:

```
interface serial 0
loopback remote
```

## Related Command

**show interfaces loopback**

# loopback remote (interface)

To loop packets through a CSU/DSU, over a DS-3 link or a channelized T1 link, to the remote CSU/DSU and back, use the **loopback remote** interface configuration command. To remove the loop, use the **no** form of this command.

**loopback remote**
**no loopback remote**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

This command applies only when the remote CSU/DSU device supports the function. It is used for testing the data communication channels. The loopback usually is performed at the line port, rather than the DTE port, of the remote CSU/DSU.

For a multiport interface processor (MIP) connected to a network via a channelized T1 link, the loopback remote interface configuration command applies if the remote interface is served by a DDS line (56 Kbps or 64 Kbps), and the device at the remote end is a CSU/DSU. In addition, the CSU/DSU at the remote end *must* react to latched DDS CSU loopback codes. Destinations that are served by other types of lines or that have CSU/DSUs that do not react to latched DDS CSU codes cannot participate in an interface remote loopback. Latched DDS CSU loopback code requirements are described in AT&T specification TR-TSY-000476, “OTGR Network Maintenance Access and Testing.”

To show interfaces currently in loopback operation, use the **show interfaces loopback** EXEC command.

## Example

The following example configures a remote loopback test:

```
interface serial 0
loopback remote
```

## Related Command

**show interfaces loopback**

# media-type

To specify the Ethernet Network Interface Module configuration on the Cisco 4000 series, use the **media-type** interface configuration command.

**media-type** [**aui** | **10baset**]
**no media-type** [**aui** | **10baset**]

## Syntax Description

| | |
|---|---|
| **aui** | (Optional) Selects a 15-pin physical connection. |
| **10baset** | (Optional) Selects an RJ45 10BaseT physical connection. |

## Default

AUI 15-pin physical connection

## Command Mode

Interface configuration

## Example

The following example selects an RJ45 10BaseT physical connection on Ethernet interface 1:

```
interface ethernet 1
media-type 10baset
```

# mop enabled

To enable an interface to support the Maintenance Operation Protocol (MOP), use the **mop enabled** interface configuration command. To disable MOP on an interface, use the **no mop enabled** command.

**mop enabled**
**no mop enabled**

## Syntax Description

This command has no arguments or keywords.

## Default

Enabled on Ethernet interfaces and disabled on all other interfaces.

## Command Mode

Interface configuration

## Example

In the following example, MOP is enabled for serial interface 0:

```
interface serial 0
mop enabled
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**mop sysid**
**mop retransmit-timer** †
**mop retries** †

# mop sysid

To enable an interface to send out periodic Maintenance Operation Protocol (MOP) system identification messages, use the **mop sysid** interface configuration command. To disable MOP message support on an interface, use the **no mop sysid** command.

**mop sysid**
**no mop sysid**

## Syntax Description

This command has no arguments or keywords.

## Default

Enabled

## Command Mode

Interface configuration

## Usage Guidelines

You can still run MOP without having the background system ID messages sent. This lets you use the MOP remote console, but does not generate messages used by the configurator.

## Example

In the following example, serial interface 0 is enabled to send MOP system identification messages:

```
interface serial 0
mop sysid
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**mop device-code** †
**mop enabled**

# mtu

To adjust the maximum packet size or maximum transmission unit (MTU) size, use the **mtu** interface configuration command. Use the **no mtu** command to restore the MTU value to its original default value.

**mtu** *bytes*
**no mtu**

## Syntax Description

| | |
|---|---|
| *bytes* | Desired size in bytes. |

## Default

Table 6-9 lists default MTU values according to media type.

**Table 6-9 Default Media MTU Values**

| Media Type | Default MTU |
|---|---|
| Ethernet | 1500 |
| Serial | 1500 |
| Token Ring | 4464 |
| ATM | 4470 |
| FDDI | 4470 |
| HSSI (HSA) | 4470 |

## Command Mode

Interface configuration

## Usage Guidelines

Each interface has a default maximum packet size or maximum transmission unit (MTU) size. This number generally defaults to the largest size possible for that type interface. On serial interfaces, the MTU size varies, but cannot be set smaller than 64 bytes.

**Note** Changing the MTU value with the **mtu** interface configuration command can affect values for the protocol-specific versions of the command (**ip mtu** for example). If the values specified with the **ip mtu** interface configuration command is the same as the value specified with the **mtu** interface configuration command, and you change the value for the **mtu** interface configuration command, the **ip mtu** value automatically matches the new **mtu** interface configuration command value. However, changing the values for the **ip mtu** configuration commands has no effect on the value for the **mtu** interface configuration command.

## Example

The following example specifies an MTU of 1000 bytes:

```
interface serial 1
mtu 1000
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**encapsulation smds** †
**ip mtu** †

# nrzi-encoding

To enable non-return to zero inverted (NRZI) line coding format, use the **nrzi-encoding** interface configuration command. Use the **no** form of this command to disable this capability.

**nrzi-encoding**
**no nrzi-encoding**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

All FSIP interface types support nonreturn to zero (NRZ) and nonreturn to zero inverted (NRZI) format. This is a line coding format that is required for serial connections in some environments. NRZ encoding is most common. NRZI encoding is used primarily with RS-232 connections in IBM environments.

## Example

In the following example, serial interface 1 is configured for NRZI encoding:

```
interface serial 1
nrzi-encoding
```

# peer default ip address pool

You can selectively disable DHCP proxy-client status on an individual asynchronous interface on a router by using the **no peer default ip address pool** interface configuration command. You can turn a single interface back on by issuing the standard command after it is turned off.

**peer default ip address pool**
**no peer default ip address pool**

## Syntax Description

This command has no arguments or keywords.

## Default

DHCP proxy-client status is not enabled until the **ip address-pool** command is issued, at which time the DHCP proxy-client feature is enabled on all asynchronous ports.

## Command Mode

Interface configuration

## Usage Guidelines

The **no peer default ip address pool** command turns off DHCP proxy-client status on individual asynchronous interfaces that are globally turned on with the **ip address-pool dhcp-proxy-client** command. If you have disabled DHCP on a given interface, you can re-enable DHCP on this interface by issuing the standard **peer default ip address pool** command. You cannot enable DHCP on any interface until the **ip address-pool dhcp-proxy-client** command is issued.

## Example

The following command disables DHCP proxy-client status on the current asynchronous interface:

```
no peer default ip address pool
```

## Related Commands

**ip address-pool dhcp-proxy-client**
**ip dhcp-server**
**show dhcp**
**interface async**
**encapsulation**
**ppp**

# ppp authentication chap

To enable Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP), and to enable a TACACS+ authorization method on a serial interface, use the **ppp authentication** interface configuration command. Use the **no** form of the command to disable this authentication.

**ppp authentication** {**chap** | **pap**} [**if-needed**] [*listname*]
**no ppp authentication**

**Caution** If you use a *list-name* that has not been configured with the **aaa authentication ppp** command, you disable PPP on this line.

## Syntax Description

| | |
|---|---|
| **chap** | Enables CHAP on a serial interface. |
| **pap** | Enables PAP on a serial interface. |
| **if-needed** | (Optional) Used with TACACS and XTACACS. Do not perform CHAP or PAP authentication if the user has already provided authentication. This option is available only on asynchronous interfaces. |
| *list-name* | (Optional) Used with AAA/TACACS+. Specify the name of a list of TACACS+ methods of authentication to use. If no listname is specified, the system uses the default. Lists and default are created with the **aaa authentication ppp** command. |

## Default

PPP authentication is not enabled.

## Command Mode

Interface configuration

## Usage Guidelines

Once you have enabled CHAP or PAP, the local communication server requires a password from remote devices. If the remote device does not support CHAP or PAP, no traffic is passed to that device.

If you are using **autoselect** on a TTY line, you will probably want to use the **ppp authentication** command to turn on PPP authentication for the corresponding interface.

When you specify the **if-needed** option, PPP authentication is not required when the user has already provided authentication. This option is useful in conjunction with the **autoselect** command, but cannot be used with AAA/TACACS+.

The *list-name* keyword can be used only when AAA/TACACS+ has been initialized, and cannot be used with the **if-needed** argument.

## Example

The following example enables CHAP on asynchronous interface 4, and uses the authentication list *MIS-access*:

```
interface async 4
encapsulation ppp
ppp authentication chap MIS-access
```

## Related Commands

**autoselect**
**encapsulation ppp**
**ppp use-tacacs**
**username**
**aaa authentication ppp**
**aaa new-model**

# ppp quality

To enable Link Quality Monitoring (LQM) on a serial interface, use the **ppp quality** interface configuration command. Use the **no** form of this command to disable LQM.

**ppp quality** *percentage*
**no ppp quality**

## Syntax Description

| | |
|---|---|
| *percentage* | Specifies the link quality threshold. Range is 1 to 100. |

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

The percentages are calculated for both incoming and outgoing directions. The outgoing quality is calculated by comparing the total number of packets and bytes sent to the total number of packets and bytes received by the peer. The incoming quality is calculated by comparing the total number of packets and bytes received to the total number of packets and bytes sent by the peer.

If the link quality percentage is not maintained, the link is deemed to be of poor quality and is taken down. The policy implements a time lag so that the link does not bounce up and down.

## Example

The following example enables LQM on serial interface 4:

```
interface serial 4
encapsulation ppp
ppp quality 80
```

## Related Commands

**encapsulation ppp**
**keepalive**

# pri-group

To specify ISDN Primary Rate Interface (PRI) on a channelized T1 card on the Cisco 7000 series, use the **pri-group** controller configuration command. Use the **no pri-group** command to remove the ISDN PRI.

**pri-group** [**timeslots** *range*]
**no pri-group**

## Syntax Description

| | |
|---|---|
| **timeslots** *range* | (Optional) Specifies a single range of values from 1 to 23. |

## Default

Disabled

## Command Mode

Controller configuration

## Usage Guidelines

When you configure ISDN PRI, you must first specify an ISDN switch type for PRI and a T1 controller.

## Example

The following example specifies ISDN PRI on T1 slot 1, port 0:

```
isdn switch-type primary-4ess
controllers t1 1/0
framing esf
linecode b8zs
pri-group timeslots 2-6
```

## Related Commands

**controller**
**framing**
**isdn switch-type**
**linecode**

# pulse-time

To enable pulsing DTR signal intervals on the serial interfaces, use the **pulse-time** interface configuration command. Use the **no pulse-time** command to restore the default interval.

**pulse-time** *seconds*
**no pulse-time**

## Syntax Description

| | |
|---|---|
| *seconds* | Integer that specifies the DTR signal interval in seconds. |

## Default

0 seconds

## Command Mode

Interface configuration

## Usage Guidelines

When the serial line protocol goes down (for example, because of loss of synchronization) the interface hardware is reset and the DTR signal is held inactive for at least the specified interval. This function is useful for handling encrypting or other similar devices that use the toggling of the DTR signal to resynchronize.

## Example

The following example enables DTR pulse signals for three seconds on serial interface 2:

```
interface serial 2
pulse-time 3
```

# ring-speed

To set the ring speed for the CSC-1R and CSC-2R Token Ring interfaces, use the **ring-speed** interface configuration command.

**ring-speed** *speed*

## Syntax Description

| | |
|---|---|
| *speed* | Integer that specifies the ring speed, either 4 for 4-Mbps or 16 for 16-Mbps operation. |

## Default

16-Mbps operation

**Caution** Configuring a ring speed that is wrong or incompatible with the connected Token Ring will cause the ring to beacon, which effectively takes the ring down and makes it nonoperational.

## Command Mode

Interface configuration

## Example

The following example sets a Token Ring interface ring speed to 4 Mbps:

```
interface tokenring 0
ring-speed 4
```

# show async status

To list the status of the asynchronous interface 1 associated with the router auxiliary port, use the **show async status** user EXEC command:

**show async status**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

Shows all asynchronous sessions, whether they are using SLIP or PPP encapsulation.

## Sample Display

The following is sample output from the **show async status** command:

```
Router> show async status

Async protocol statistics:
  Rcvd: 5448 packets, 7682760 bytes
        1 format errors, 0 checksum errors, 0 overrun, 0 no buffer
  Sent: 5455 packets, 7682676 bytes, 0 dropped


 Int           Local          Remote Qd InPack OutPac Inerr  Drops  MTU Qsz
   1     192.31.7.84         Dynamic  0      0      0     0      0 1500  10
```

Table 6-10 describes significant fields shown in the display.

**Table 6-10 Show Async Status Field Descriptions**

| Field | Description |
|---|---|
| Rcvd: | Statistics on packets received. |
| 5548 packets | Packets received. |
| 7682760 bytes | Total number of bytes. |
| 1 format errors | Packets with a bad IP header, even before the checksum is calculated. |
| 0 checksum errors | Count of checksum errors. |
| 0 overrun | Number of giants received. |
| 0 no buffer | Number of packets received when no buffer was available. |
| Sent: | Statistics on packets sent. |
| 5455 packets | Packets sent. |
| 7682676 bytes | Total number of bytes. |
| 0 dropped | Number of packets dropped. |
| Int | Interface number. |

| Field | Description |
|---|---|
| * | Line currently in use. |
| Local | Local IP address on the link. |
| Remote | Remote IP address on the link; "Dynamic" indicates that a remote address is allowed but has not been specified; "None" indicates that no remote address is assigned or being used. |
| Qd | Number of packets on hold queue (Qsz is max). |
| InPack | Number of packets received. |
| OutPac | Number of packets sent. |
| Inerr | Number of total input errors; sum of format errors, checksum errors, overruns and no buffers. |
| Drops | Number of packets received that would not fit on the hold queue. |
| MTU | Current maximum transmission unit size. |
| Qsz | Current output hold queue size. |

## Related Commands

**async default ip address**
**async dynamic address**
**async dynamic routing**
**async mode dedicated**
**async mode interactive**
**interface async**

# show compress

To display compression statistics, use the **show compress** EXEC command.

**show compress**

## Syntax Description

This command has no arguments or parameters.

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show compress** command:

```
Router# show compress

Serial0
uncompressed bytes xmt/rcv 10710562/11376835
1  min avg ratio xmt/rcv 2.773/2.474
5  min avg ratio xmt/rcv 4.084/3.793
10 min avg ratio xmt/rcv 4.125/3.873
no bufs xmt 0 no bufs rcv 0
resets 0
```

Table 6-11 describes the fields shown in the display.

**Table 6-11 Show Compress Field Descriptions**

| Field | Description |
|---|---|
| Serial0 | Name and number of the interface. |
| uncompressed bytes xmt/rcv | Total number of uncompressed bytes sent and received. |
| 1 min avg ratio xmt/rcv<br>5 min avg ratio xmt/rcv<br>10 min avg ratio xmt/rcv | Static compression ratio for bytes sent and received, averaged over 1, 5, and 10 minutes. |
| no bufs xmt | Number of times buffers were not available to compress data being sent. |
| no bufs rcv | Number of times buffers were not available to uncompress data being received. |
| resets | Number of resets. |

## Related Command

**compress**

# show controllers cbus

Use the **show controllers cbus** privileged EXEC command on the AGS+ to display all information under the ciscoBus controller card. This command also shows the capabilities of the card and reports controller-related failures.

**show controllers cbus**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Privileged EXEC

## Sample Displays

The following is sample output from the **show controllers cbus** command:

```
Router# show controllers cbus

cBus 1, controller type 3.0, microcode version 2.0
  128 Kbytes of main memory, 32 Kbytes cache memory
  40 1520 byte buffers, 14 4484 byte buffers
  Restarts: 0 line down, 0 hung output, 0 controller error
 --More--
HSCI 1, controller type 10.0, microcode version 129.3
  Interface 6 - Hssi0, electrical interface is Hssi DTE
    5 buffer RX queue threshold, 7 buffer TX queue limit, buffer size 1520
    ift 0004, rql 2, tq 0000 0000, tql 7
    Transmitter delay is 0 microseconds
 MEC 3, controller type 5.1, microcode version 130.6
  Interface 18 - Ethernet2, station address 0000.0c02.a03c (bia 0000.0c02.a03c)
    10 buffer RX queue threshold, 7 buffer TX queue limit, buffer size 1520
    ift 0000, rql 10, tq 0000 0000, tql 7
    Transmitter delay is 0 microseconds
  Interface 19 - Ethernet3, station address 0000.0c02.a03d (bia 0000.0c02.a03d)
    10 buffer RX queue threshold, 7 buffer TX queue limit, buffer size 1520
    ift 0000, rql 10, tq 0000 0000, tql 7
    Transmitter delay is 0 microseconds
```

Table 6-12 describes the fields shown in the following lines of output from the display.

```
cBus 1, controller type 3.0, microcode version 2.0
  128 Kbytes of main memory, 32 Kbytes cache memory
  40 1520 byte buffers, 14 4484 byte buffers
  Restarts: 0 line down, 0 hung output, 0 controller error
```

**Table 6-12 Show Controllers cBus Field Descriptions—Part 1**

| Field | Description |
|---|---|
| cBus 1 | Card type and number (varies depending on card). |
| controller type 3.0 | Version number of the card. |
| microcode version 2.0 | Version number of the card's internal software (in read-only memory). |

| Field | Description |
|---|---|
| 128 Kbytes of main memory | Amount of main memory on the card. |
| 32 Kbytes cache memory | Amount of cache memory on the card. |
| 40 1520 byte buffers | Number of buffers of this size on the card. |
| 14 4484 byte buffers | Number of buffers of this size on the card. |
| Restarts<br>0 line down<br>0 hung output<br>0 controller error | Count of restarts due to the following conditions:<br>Communication line down<br>Output unable to transmit<br>Internal error |

Table 6-13 describes the fields shown in the following lines of output from the display:

```
HSCI 1, controller type 10.0, microcode version 129.3
  Interface 6 - Hssi0, electrical interface is Hssi DTE
    5 buffer RX queue threshold, 7 buffer TX queue limit, buffer size 1520
    ift 0004, rql 2, tq 0000 0000, tql 7
    Transmitter delay is 0 microseconds
```

**Table 6-13 Show Controllers cBus Field Descriptions—Part 2**

| Field | Description |
|---|---|
| HSCI 1 | Card type and number (varies depending on card). |
| controller type 10.0 | Version number of the card. |
| microcode version 129.3 | Version number of the card's internal software (in read-only memory). |
| Interface 6 | Physical interface number. |
| Hssi 0 | Logical name for this interface. |
| electrical interface is Hssi DTE | Self-explanatory. |
| 5 buffer RX queue threshold | Maximum number of buffers allowed in the receive queue. |
| 7 buffer TX queue limit | Maximum number of buffers allowed in the transmit queue. |
| buffer size 1520 | Size of the buffers on this card (in bytes). |
| ift 0004 | Interface type code.<br>0 = EIP<br>1 = FSIP<br>4 = HIP<br>5 = TRIP<br>6 = FIP<br>7 = AIP |
| rql 2 | Receive queue limit. Current number of buffers allowed for the receive queue. It is used to limit the number of buffers used by a particular inbound interface. When equal to 0, all of that interface's receive buffers are in use. |
| tq 0000 0000 | Transmit queue head and tail pointers. |
| tql 7 | Transmit queue limit. Current number of buffers allowed for transmit queue. It limits the maximum cbus buffers allowed to sit on a particular interface's transmit queue. |
| Transmitter delay is 0 microseconds | Transmitter delay between the packets. |

The **show controllers cbus** command displays the internal status of the SP and each cBus interface processor (IP), including the slot location, the card hardware version, and the currently-running microcode version. It also lists each interface (port) on each IP including the logical interface number, interface type, physical (slot/port) address, and hardware (station address) of each interface. The following display shows an AIP installed in IP slot 4, the running AIP microcode is Version 170.30, the PLIM type is 4B/5B, and the available bandwidth is 100 Mbps:

```
Router# show controllers cbus

Switch Processor 5, hardware version 11.1, microcode version 170.46
  Microcode loaded from system
  512 Kbytes of main memory, 128 Kbytes cache memory
  60 1520 byte buffers, 91 4496 byte buffers
  Restarts: 0 line down, 0 hung output, 0 controller error
 AIP 4, hardware version 1.0, microcode version 170.30
  Microcode loaded from system
  Interface 32 - ATM4/0, PLIM is 4B5B(100Mbps)
    15 buffer RX queue threshold, 36 buffer TX queue limit, buffer size 4496
    ift 0007, rql 12, tq 0000 0620, tql 36
    Transmitter delay is 0 microseconds
```

# show controllers cxbus

Use the **show controllers cxbus** privileged EXEC command to display information about the Switch Processor (SP) CxBus controller on the Cisco 7000 series. This command displays information that is specific to the interface hardware. The information displayed is generally useful for diagnostic tasks performed by technical support personnel only.

**show controllers cxbus**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Privileged EXEC

## Sample Display

The following is sample output on the Cisco 7000 from the **show controllers cxbus** command:

```
Router# show controllers cxbus

Switch Processor 5, hardware version 11.1, microcode version 172.6
  Microcode loaded from system
  512 Kbytes of main memory, 128 Kbytes cache memory
  75 1520 byte buffers, 86 4484 byte buffers
  Restarts: 0 line down, 0 hung output, 0 controller error
 CIP 3, hardware version 1.1, microcode version 170.1
  Microcode loaded from system
  CPU utilization 7%, sram 145600/512K, dram 86688/2M
  Interface 24 - Channel 3/0
    43 buffer RX queue threshold, 61 buffer TX queue limit, buffer size 4484
    ift 0007, rql 32, tq 0000 0468, tql 61
    Transmitter delay is 0 microseconds
  Interface 25 - Channel 3/1
    43 buffer RX queue threshold, 61 buffer TX queue limit, buffer size 4484
    ift 0007, rql 34, tq 0000 0000, tql 61
    Transmitter delay is 0 microseconds
```

Table 6-14 describes the fields shown in the display.

**Table 6-14 Show Controllers CxBus Field Descriptions**

| Field | Description |
|---|---|
| IP type, slot number | Unit type and slot number. |
| hardware version | Version number of the controller. |
| microcode version | Version number of the controller’s internal software (in read-only memory). |
| Microcode loaded from | Source of microcode; can be system, ROM, or Flash. |
| main memory<br>cache memory | Amount of main and cache memory on the processor. |
| byte system buffer | An extra buffer left over after carving the normal pools. It is used for host-generated traffic when available. |

| | |
|---|---|
| Restarts<br>line down<br>hung output<br>controller error | Number of restarts due to the following conditions:<br>Communication line down<br>Output unable to transmit<br>Internal error |
| CPU utilization | Measure of how busy the CPU is during a given time interval. |
| sram | The first value is the number of bytes of sram free (that is, not being used by code or data). The second value is the total bytes available of sram, and is expressed in terms of kilobytes or megabytes. The sram is the high-speed static RAM that is used for running the operational code. |
| dram | The first value is the number of bytes of dram free (that is, not being used by code or data). The second value is the total bytes available of dram, and is expressed in terms of kilobytes or megabytes. The dram is normal dynamic RAM that is used for packet buffers, data, and so on. |
| Interface number | Names of interfaces by CxBus interface type, slot, and port number. |
| RX buffers | Number of buffers for received packets. |
| TX queue limit | Maximum number of buffers in transmit queue. |
| ift | Interface type code.<br>0 = EIP<br>1 = FSIP<br>4 = HIP<br>5 = TRIP<br>6 = FIP<br>7 = AIP |
| rql | Receive queue limit. Current number of buffers allowed for the receive queue. It is used to limit the number of buffers used by a particular inbound interface. When equal to 0, all of that interface's receive buffers are in use. |
| tq | Transmit queue head and tail pointers. |
| tql | Transmit queue limit. Current number of buffers allowed for transmit queue. It limits the maximum cbus buffers allowed to sit on a particular interface's transmit queue. |
| Transmitter delay | Delay between outgoing frames. |
| Station address | The hardware address of the interface. |

The following is sample output showing an interface port that has a G.703 cable attached:

```
Router# show controllers cxbus

FSIP 2, hardware version 1.0, microcode version 170.10
  Microcode loaded from flash xyzabc/fsip_q170-10
  Interface 16 - Serial2/0, electrical interface is G.703 Unbalanced
    10 buffer RX queue threshold, 15 buffer TX queue limit, buffer size 1520
    ift 0001, rql 9, tq 0000 0000, tql 15
    Transmitter delay is 0 microseconds
    Interface 17 - Serial2/1, electrical interface is G.703 Unbalanced
    11 buffer RX queue threshold, 14 buffer TX queue limit, buffer size 2104
    ift 0001, rql 10, tq 0000 0000, tql 14
    Transmitter delay is 0 microseconds
  Interface 18 - Serial2/2, electrical interface is G.703 Balanced
    10 buffer RX queue threshold, 15 buffer TX queue limit, buffer size 1520
    ift 0001, rql 9, tq 0000 0000, tql 15
    Transmitter delay is 0 microseconds
  Interface 19 - Serial2/3, electrical interface is G.703 Balanced
    10 buffer RX queue threshold, 15 buffer TX queue limit, buffer size 1520
```

```
ift 0001, rql 8, tq 0000 0428, tql 15
Transmitter delay is 0 microseconds
```

In output, “balanced” and “unbalanced” refer to the electrical signal levels at the connector resulting from different line termination schemes.

# show controllers e1

Use the **show controllers e1** privileged EXEC command on the Cisco 7000 to display information about the E1 links supported by the MultiChannel Interface Processor (MIP).

**show controllers e1** [*slot*/*port*]

## Syntax Description

| | |
|---|---|
| *slot* | Specifies the backplane slot number and can be 0, 1, 2, 3, or 4. |
| *port* | Specifies the port number of the controller and can be 0 or 1. |

## Command Mode

Privileged EXEC

## Usage Guidelines

For the E1 interface on the Cisco 7000, the MIP can query the port adapters to determine their current status. Issue a **show controllers e1** command to display statistics about the E1 link.

If you specify a slot and port number, each 15-minute period will be displayed.

This command displays controller status that is specific to the controller hardware. The information displayed is generally useful for diagnostic tasks performed by technical support personnel only.

## Sample Display

The following is sample output from the **show controllers e1** command on the Cisco 7000 series:

```
Router# show controllers e1

e1 0/0 is up.
  Applique type is Channelized E1 - unbalanced
  Framing is CRC4, Line Code is HDB3
  No alarms detected.
  Data in current interval (725 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
  Total Data (last 24 hours)
     0 Line Code Violations, 0 Path Code Violations,
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins,
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
```

Table 6-15 describes the **show controllers e1** display fields.

**Table 6-15 Show Controllers E1 Field Descriptions**

| Field | Description |
|---|---|
| e1 0/0 is up. | The E1 controller 0 in slot 0 is operating. The controller's state can be up, down, or administratively down. Loopback conditions are shown by (Locally looped) or (Remotely Looped). |
| Applique type | The applique type is shown and will indicate balanced or unbalanced. |

| Field | Description |
|---|---|
| Framing is | Shows the current framing type. |
| Linecode is | Shows the current linecode type. |
| No alarms detected. | Any alarms detected by the controller are displayed here. Possible alarms are as follows:<br>• Transmitter is sending remote alarm.<br>• Transmitter is sending AIS.<br>• Receiver has loss of signal.<br>• Receiver is getting AIS.<br>• Receiver has loss of frame.<br>• Receiver has remote alarm.<br>• Receiver has no alarms. |
| Data in current interval (725 seconds elapsed) | Shows the current accumulation period, which rolls into the 24 hour accumulation every 15 minutes. Accumulation period is from 1 to 900 seconds. The oldest 15-minute period falls off the back of the 24-hour accumulation buffer. |
| Line Code Violations | Indicates the occurrence of either a Bipolar Violation (BPV) or Excessive Zeroes (EXZ) error event. |
| Path Code Violations | Indicates a frame synchronization bit error in the D4 and E1-noCRC formats, or a CRC error in the ESF and E1-CRC formats. |
| Slip Secs | Indicates the replication or deletion of the payload bits of a DS1 frame. A slip might be performed when there is a difference between the timing of a synchronous receiving terminal and the received signal. |
| Fr Loss Secs | Indicates the number of seconds an Out Of Frame (OOF) error is detected. |
| Line Err Secs | Line Errored Seconds (LES) is a second in which one or more Line Code Violation errors are detected. |
| Degraded Mins | A Degraded Minute is one in which the estimated error rate exceeds 1E-6 but does not exceed 1E-3. |
| Errored Secs | In ESF and E1 CRC links, an Errored Second is a second in which one of the following are detected: one or more Path Code Violations; one or more Out of Frame defects; one or more Controlled Slip events; a detected AIS defect.<br>For SF and E1 no-CRC links, the presence of Bipolar Violations also triggers an Errored Second. |
| Bursty Err Secs | A second with fewer than 320 and more than 1 Path Coding Violation error, no Severely Errored Frame defects and no detected incoming AIS defects. Controlled slips are not included in this parameter. |
| Severely Err Secs | For ESF signals, a second with one of the following errors: 320 or more Path Code Violation errors; one or more Out of Frame defects; a detected AIS defect.<br>For E1-CRC signals, a second with one of the following errors: 832 or more Path Code Violation errors; one or more Out of Frame defects.<br>For E1-nonCRC signals, a second with 2048 Line Code Violations or more.<br>For D4 signals, a count of 1-second intervals with Framing Errors, or an Out of Frame defect, or 1544 Line Code Violations. |
| Unavail Secs | A count of the total number of seconds on the interface. |

# show controllers ethernet

Use the **show controllers ethernet** EXEC command to display information on the Cisco 2500, 3000, or 4000.

**show controllers ethernet** *number*

## Syntax Description

| | |
|---|---|
| *number* | Interface number of the Ethernet interface. |

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show controllers ethernet** command on the Cisco 4000:

```
Router# show controllers ethernet 0

LANCE unit 0, NIM slot 1, NIM type code 4, NIM version 1
Media Type is 10BaseT, Link State is Up, Squelch is Normal
idb 0x4060, ds 0x5C80, regaddr = 0x8100000
IB at 0x600D7AC: mode=0x0000, mcfilter 0000/0001/0000/0040
station address 0000.0c03.a14f  default station address 0000.0c03.a14f
buffer size 1524
RX ring with 32 entries at 0xD7E8
Rxhead = 0x600D8A0 (12582935), Rxp = 0x5CF0(23)
00 pak=0x60336D0 ds=0x6033822 status=0x80 max_size=1524 pak_size=98
01 pak=0x60327C0 ds=0x6032912 status=0x80 max_size=1524 pak_size=98
02 pak=0x6036B88 ds=0x6036CDA status=0x80 max_size=1524 pak_size=98
03 pak=0x6041138 ds=0x604128A status=0x80 max_size=1524 pak_size=98
04 pak=0x603FAA0 ds=0x603FBF2 status=0x80 max_size=1524 pak_size=98
05 pak=0x600DC50 ds=0x600DDA2 status=0x80 max_size=1524 pak_size=98
06 pak=0x6023E48 ds=0x6023F9A status=0x80 max_size=1524 pak_size=1506
07 pak=0x600E3D8 ds=0x600E52A status=0x80 max_size=1524 pak_size=1506
08 pak=0x6020990 ds=0x6020AE2 status=0x80 max_size=1524 pak_size=386
09 pak=0x602D4E8 ds=0x602D63A status=0x80 max_size=1524 pak_size=98
10 pak=0x603A7C8 ds=0x603A91A status=0x80 max_size=1524 pak_size=98
11 pak=0x601D4D8 ds=0x601D62A status=0x80 max_size=1524 pak_size=98
12 pak=0x603BE60 ds=0x603BFB2 status=0x80 max_size=1524 pak_size=98
13 pak=0x60318B0 ds=0x6031A02 status=0x80 max_size=1524 pak_size=98
14 pak=0x601CD50 ds=0x601CEA2 status=0x80 max_size=1524 pak_size=98
15 pak=0x602C5D8 ds=0x602C72A status=0x80 max_size=1524 pak_size=98
16 pak=0x60245D0 ds=0x6024722 status=0x80 max_size=1524 pak_size=98
17 pak=0x6008328 ds=0x600847A status=0x80 max_size=1524 pak_size=98
18 pak=0x601EB70 ds=0x601ECC2 status=0x80 max_size=1524 pak_size=98
19 pak=0x602DC70 ds=0x602DDC2 status=0x80 max_size=1524 pak_size=98
20 pak=0x60163E0 ds=0x6016532 status=0x80 max_size=1524 pak_size=98
21 pak=0x602CD60 ds=0x602CEB2 status=0x80 max_size=1524 pak_size=98
22 pak=0x6037A98 ds=0x6037BEA status=0x80 max_size=1524 pak_size=98
23 pak=0x602BE50 ds=0x602BFA2 status=0x80 max_size=1524 pak_size=98
24 pak=0x6018988 ds=0x6018ADA status=0x80 max_size=1524 pak_size=98
25 pak=0x6033E58 ds=0x6033FAA status=0x80 max_size=1524 pak_size=98
26 pak=0x601BE40 ds=0x601BF92 status=0x80 max_size=1524 pak_size=98
27 pak=0x6026B78 ds=0x6026CCA status=0x80 max_size=1524 pak_size=98
28 pak=0x6024D58 ds=0x6024EAA status=0x80 max_size=1524 pak_size=74
29 pak=0x602AF40 ds=0x602B092 status=0x80 max_size=1524 pak_size=98
30 pak=0x601FA80 ds=0x601FBD2 status=0x80 max_size=1524 pak_size=98
```

```
31 pak=0x6038220 ds=0x6038372 status=0x80 max_size=1524 pak_size=98
TX ring with 8 entries at 0xDA20, tx_count = 0
tx_head = 0x600DA58 (12582919), head_txp = 0x5DC4 (7)
tx_tail = 0x600DA58 (12582919), tail_txp = 0x5DC4 (7)
00 pak=0x000000 ds=0x600CF12 status=0x03 status2=0x0000 pak_size=118
01 pak=0x000000 ds=0x602126A status=0x03 status2=0x0000 pak_size=60
02 pak=0x000000 ds=0x600CF12 status=0x03 status2=0x0000 pak_size=118
03 pak=0x000000 ds=0x600CF12 status=0x03 status2=0x0000 pak_size=118
04 pak=0x000000 ds=0x600CF12 status=0x03 status2=0x0000 pak_size=118
05 pak=0x000000 ds=0x600CF12 status=0x03 status2=0x0000 pak_size=118
06 pak=0x000000 ds=0x600CF12 status=0x03 status2=0x0000 pak_size=118
07 pak=0x000000 ds=0x6003ED2 status=0x03 status2=0x0000 pak_size=126
0 missed datagrams, 0 overruns, 2 late collisions, 2 lost carrier events
0 transmitter underruns, 0 excessive collisions,  0 tdr, 0 babbles
0 memory errors, 0 spurious initialization done interrupts
0 no enp status, 0 buffer errors, 0 overflow errors
10 one_col, 10 more_col, 22 deferred, 0 tx_buff
0 throttled, 0 enabled
Lance csr0 = 0x73
```

# show controllers fddi

Use the **show controllers fddi** user EXEC command to display all information under the FDDI controller card on the AGS+ or FDDI Interface Processor (FIP) on the Cisco 7000.

**show controllers fddi**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

This command reflects the internal state of the chips and information the system uses for bridging and routing that is specific to the interface hardware. The information displayed is generally useful for diagnostic tasks performed by technical support personnel only.

## Sample Display

The following is sample output from the **show controllers fddi** command on the Cisco 7000:

```
Router# show controllers fddi

Fddi2/0 - hardware version 2.2, microcode version 1.2
  Phy-A registers:
    cr0 4, cr1 0, cr2 0, status 3, cr3 0
  Phy-B registers:
    cr0 4, cr1 4, cr2 0, status 3, cr3 0
  FORMAC registers:
    irdtlb  71C2, irdtneg F85E, irdthtt F5D5, irdmir  FFFF0BDC
    irdtrth F85F, irdtmax FBC5, irdtvxt 5959, irdstmc 0810
    irdmode 6A20, irdimsk 0000, irdstat 8060, irdtpri 0000
  FIP registers
    ccb:   002C  cmd:   0006  fr:   000F  mdptr: 0000  mema: 0000
    icb:   00C0  arg:   0003  app:  0004  mdpg:  0000  af:   0603
    clm:   E002  bcn:   E016  clbn: 0198  rxoff: 002A  en:   0001
    clmbc: 8011  bcnbc: 8011  robn: 0004  park:  0000  fop:  8004

    txchn: 0000  pend:  0000  act:  0000  tail:  0000  cnt:  0000
    state: 0003  check: 0000  eof:  0000  tail:  0000  cnt:  0000
    rxchn: 0000  buf0:  0534  nxt0: 0570  eof:   0000  tail: 0000
    eofch: 0000  buf1:  051C  nxt1: 0528  pool:  0050  err:  005C

    head:  0984  cur:   0000  t0:   0030  t1:    0027  t2:   000F
    tail:  0984  cnt:   0001  t3:   0000  rxlft: 000B  used: 0000
    txq_s: 0018  txq_f: 0018  Aarm: 0000  Barm:  1388  fint: 8004

 Total LEM: phy-a 6, phy-b 13
```

The last line of output indicates how many LEM events occurred on the specific PHY.

# show controllers lex

To show hardware and software information about the LAN Extender chassis, use the **show controllers lex** EXEC command.

**show controllers lex** [*number*]
**show controllers lex** [*slot*/*port*] (for the Cisco 7000 series)

## Syntax Description

| | |
|---|---|
| *number* | (Optional) Number of the LAN Extender interface about which to display information. |
| *slot* | (Optional) Specifies the backplane slot number on the Cisco 7000 series, and can be 0, 1, 2, 3, or 4. |
| *port* | (Optional) Specifies the port number of the controller and can be 0 or 1. |

## Command Mode

EXEC

## Usage Guidelines

Use the **show controllers lex** command to display information about the hardware revision level, software version number, Flash memory size, serial number, and other information related to the configuration of the LAN Extender.

## Sample Display

The following is sample output from the **show controllers lex** command:

```
Router# show controllers lex 0

Lex0:
FLEX Hardware revision 1
FLEX Software version 255.0
128K bytes of flash memory
Serial number is 123456789
Station address is 0000.4060.1100
```

The following is sample output from the **show controller**s **lex** command when the LAN Extender interface is not bound to a serial interface:

```
Router# show controller lex 1

Lex1 is not bound to a serial interface
```

Table 6-16 describes the fields shown in the output.

**Table 6-16 Show Controllers Lex Field Description**

| Field | Description |
|---|---|
| Lex0: | Number of the LAN Extender interface |

| Field | Description |
|---|---|
| FLEX Hardware revision | Revision number of the Cisco 1000 series LAN Extender chassis |
| FLEX Software version | Revision number of the software running on the LAN Extender chassis |
| 128K bytes of Flash memory | Amount of Flash memory in the LAN Extender |
| Serial number | Serial number of the LAN Extender chassis |
| Station address | MAC address of the LAN Extender chassis |

# show controllers mci

Use the **show controllers mci** privileged EXEC command to display all information under the Multiport Communications Interface card or the SCI. This command displays information the system uses for bridging and routing that is specific to the interface hardware. The information displayed is generally useful for diagnostic tasks performed by technical support personnel only.

**show controllers mci**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Privileged EXEC

## Sample Display

The following is sample output from the **show controllers mci** command:

```
Router# show controllers mci

MCI 0, controller type 1.1, microcode version 1.8
   128 Kbytes of main memory, 4 Kbytes cache memory
22 system TX buffers, largest buffer size 1520
   Restarts: 0 line down, 0 hung output, 0 controller error
Interface 0 is Ethernet0, station address 0000.0c00.d4a6
   15 total RX buffers, 11 buffer TX queue limit, buffer size 1520
   Transmitter delay is 0 microseconds
Interface 1 is Serial0, electrical interface is V.35 DTE
   15 total RX buffers, 11 buffer TX queue limit, buffer size 1520
   Transmitter delay is 0 microseconds
   High speed synchronous serial interface
Interface 2 is Ethernet1, station address aa00.0400.3be4
   15 total RX buffers, 11 buffer TX queue limit, buffer size 1520
   Transmitter delay is 0 microseconds
Interface 3 is Serial1, electrical interface is V.35 DCE
   15 total RX buffers, 11 buffer TX queue limit, buffer size 1520
   Transmitter delay is 0 microseconds
   High speed synchronous serial interface
```

Table 6-17 describes significant fields shown in the display.

**Table 6-17 Show Controllers MCI Field Descriptions**

| Field | Description |
|---|---|
| MCI 0 | Card type and unit number (varies depending on card). |
| controller type 1.1 | Version number of the card. |
| microcode version 1.8 | Version number of the card’s internal software (in read-only memory). |
| 128 Kbytes of main memory | Amount of main memory on the card. |
| 4 Kbytes cache memory | Amount of cache memory on the card. |
| 22 system TX buffers | Number of buffers that hold packets to be transmitted. |

| Field | Description |
|---|---|
| largest buffer size 1520 | Largest size of these buffers (in bytes). |
| Restarts<br>0 line down<br>0 hung output<br>0 controller error | Count of restarts due to the following conditions:<br>Communication line down<br>Output unable to transmit<br>Internal error |
| Interface 0 is Ethernet0 | Names of interfaces, by number. |
| electrical interface is V.35 DTE | Line interface type for serial connections. If the jumper on the AGS+ applique enables NRZI mode, then this field will indicate V.35 NRZI DTE or DCE. |
| 15 total RX buffers | Number of buffers for received packets. |
| 11 buffer TX queue limit | Maximum number of buffers in transmit queue. |
| Transmitter delay is 0 microseconds | Delay between outgoing frames. |
| Station address 0000.0c00.d4a6 | Hardware address of the interface. |

**Note** The interface type is only queried at startup. If the hardware changes *subsequent* to initial startup, then the wrong type is reported. This has *no* adverse effect on the operation of the software. For instance, if a DCE cable is connected to a dual-mode V.35 applique after the unit has been booted, then the display presented for **show interfaces** incorrectly reports attachment to a DTE device although the software recognizes the DCE interface and behaves accordingly.

## Related Command

**tx-queue-limit**

# show controllers pcbus

To display all information about the ISA bus interface, use the **show controllers pcbus** privileged EXEC command.

**show controllers pcbus**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Privileged EXEC

## Usage Guidelines

This command is valid on LanOptics' Branchcard or Stacknet 2000 products only.

## Sample Display

The following is sample output from the **show controllers pcbus** command:

```
Router# show controllers pcbus

PCbus unit 0,  Name = PCbus0  Hardware is ISA PCbus shared RAM
IDB at 0x3719B0,  Interface driver data structure at 0x3735F8
Control/status register at 0x2110008,  Shared memory at 0xC000000
Shared memory is initialized

Shared memory interface control block :
Magic no = 0x41435A56 (valid)  Version = 1.0
Shared memory size = 64K bytes,  Interface is NOT shutdown
Interface state is up, line protocol is up

Tx buffer : (control block at 0xC000010)
Start offset = 0x30,  Size = 0x7FE8,  Overflows = 1
GET_ptr = 0x4F6C,  PUT_ptr = 0x4F6C,  WRAP_ptr = 0x3BB0

Rx buffer : (control block at 0xC000020)
Start offset = 0x8018,  Size 0x7FE8,  Overflows = 22250698
GET_ptr = 0x60,  PUT_ptr = 0x60,  WRAP_ptr = 0x7FD0

Interrupts received = 567
```

# show controllers serial

Use the **show controllers serial** privileged EXEC command to display information that is specific to the interface hardware. The information displayed is generally useful for diagnostic tasks performed by technical support personnel only.

**show controllers serial**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Privileged EXEC

## Sample Display

Sample output of the **show controllers serial** command on the Cisco 4000 follows:

```
Router# show controllers serial

MK5 unit 0, NIM slot 1, NIM type code 7, NIM version 1
idb = 0x6150, driver structure at 0x34A878, regaddr = 0x8100300
IB at 0x6045500: mode=0x0108, local_addr=0, remote_addr=0
N1=1524, N2=1, scaler=100, T1=1000, T3=2000, TP=1
buffer size 1524
DTE V.35 serial cable attached
RX ring with 32 entries at 0x45560 : RLEN=5, Rxhead 0
00 pak=0x6044D78  ds=0x6044ED4 status=80 max_size=1524 pak_size=0
01 pak=0x60445F0  ds=0x604474C status=80 max_size=1524 pak_size=0
02 pak=0x6043E68  ds=0x6043FC4 status=80 max_size=1524 pak_size=0
03 pak=0x60436E0  ds=0x604383C status=80 max_size=1524 pak_size=0
04 pak=0x6042F58  ds=0x60430B4 status=80 max_size=1524 pak_size=0
05 pak=0x60427D0  ds=0x604292C status=80 max_size=1524 pak_size=0
06 pak=0x6042048  ds=0x60421A4 status=80 max_size=1524 pak_size=0
07 pak=0x60418C0  ds=0x6041A1C status=80 max_size=1524 pak_size=0
08 pak=0x6041138  ds=0x6041294 status=80 max_size=1524 pak_size=0
09 pak=0x60409B0  ds=0x6040B0C status=80 max_size=1524 pak_size=0
10 pak=0x6040228  ds=0x6040384 status=80 max_size=1524 pak_size=0
11 pak=0x603FAA0  ds=0x603FBFC status=80 max_size=1524 pak_size=0
12 pak=0x603F318  ds=0x603F474 status=80 max_size=1524 pak_size=0
13 pak=0x603EB90  ds=0x603ECEC status=80 max_size=1524 pak_size=0
14 pak=0x603E408  ds=0x603E564 status=80 max_size=1524 pak_size=0
15 pak=0x603DC80  ds=0x603DDDC status=80 max_size=1524 pak_size=0
16 pak=0x603D4F8  ds=0x603D654 status=80 max_size=1524 pak_size=0
17 pak=0x603CD70  ds=0x603CECC status=80 max_size=1524 pak_size=0
18 pak=0x603C5E8  ds=0x603C744 status=80 max_size=1524 pak_size=0
19 pak=0x603BE60  ds=0x603BFBC status=80 max_size=1524 pak_size=0
20 pak=0x603B6D8  ds=0x603B834 status=80 max_size=1524 pak_size=0
21 pak=0x603AF50  ds=0x603B0AC status=80 max_size=1524 pak_size=0
22 pak=0x603A7C8  ds=0x603A924 status=80 max_size=1524 pak_size=0
23 pak=0x603A040  ds=0x603A19C status=80 max_size=1524 pak_size=0
24 pak=0x60398B8  ds=0x6039A14 status=80 max_size=1524 pak_size=0
25 pak=0x6039130  ds=0x603928C status=80 max_size=1524 pak_size=0
26 pak=0x60389A8  ds=0x6038B04 status=80 max_size=1524 pak_size=0
27 pak=0x6038220  ds=0x603837C status=80 max_size=1524 pak_size=0
28 pak=0x6037A98  ds=0x6037BF4 status=80 max_size=1524 pak_size=0
29 pak=0x6037310  ds=0x603746C status=80 max_size=1524 pak_size=0
30 pak=0x6036B88  ds=0x6036CE4 status=80 max_size=1524 pak_size=0
31 pak=0x6036400  ds=0x603655C status=80 max_size=1524 pak_size=0
```

```
TX ring with 8 entries at 0x45790 : TLEN=3, TWD=7
tx_count = 0, tx_head = 7, tx_tail = 7
00 pak=0x000000 ds=0x600D70C status=0x38 max_size=1524 pak_size=22
01 pak=0x000000 ds=0x600D70E status=0x38 max_size=1524 pak_size=2
02 pak=0x000000 ds=0x600D70E status=0x38 max_size=1524 pak_size=2
03 pak=0x000000 ds=0x600D70E status=0x38 max_size=1524 pak_size=2
04 pak=0x000000 ds=0x600D70E status=0x38 max_size=1524 pak_size=2
05 pak=0x000000 ds=0x600D70E status=0x38 max_size=1524 pak_size=2
06 pak=0x000000 ds=0x600D70E status=0x38 max_size=1524 pak_size=2
07 pak=0x000000 ds=0x6000000 status=0x38 max_size=1524 pak_size=0
XID/Test TX desc at 0xFFFFFF, status=0x30, max_buffer_size=0, packet_size=0
XID/Test RX desc at 0xFFFFFF, status=0x0, max_buffer_size=0, packet_size=0
Status Buffer at 0x60459C8: rcv=0, tcv=0, local_state=0, remote_state=0
phase=0, tac=0, currd=0x00000, curxd=0x00000
bad_frames=0, frmrs=0, T1_timeouts=0, rej_rxs=0, runts=0
0 missed datagrams, 0 overruns, 0 bad frame addresses
0 bad datagram encapsulations, 0 user primitive errors
0 provider primitives lost, 0 unexpected provider primitives
0 spurious primitive interrupts, 0 memory errors, 0 tr
%LINEPROTO-5-UPDOWN: Linansmitter underruns
mk5025 registers: csr0 = 0x0E00, csr1 = 0x0302, csr2 = 0x0704
                  csr3 = 0x5500, csr4 = 0x0214, csr5 = 0x0008
```

# show controllers t1

Use the **show controllers t1** privileged EXEC command on the Cisco 7000 to display information about the T1 links supported by the Multichannel Interface Processor (MIP).

**show controllers t1** [*slot*/*port*]

## Syntax Description

| | |
|---|---|
| *slot* | Specifies the backplane slot number and can be 0, 1, 2, 3, or 4. |
| *port* | Specifies the port number of the controller and can be 0 or 1. |

## Command Mode

EXEC

## Usage Guidelines

This command displays controller status that is specific to the controller hardware. The information displayed is generally useful for diagnostic tasks performed by technical support personnel only.

For the T1 interface on the Cisco 7000, the MIP can query the port adapters to determine their current status. Issue a **show controllers t1** command to display statistics about the T1 link.

If you specify a slot and port number, each 15 minute period will be displayed.

## Sample Display

The following is sample output from the **show controllers t1** command on the Cisco 7000 series:

```
Router# show controllers t1

T1 0/0 is up.
  No alarms detected.
  Data in current interval (725 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
  Total Data (last 24 hours)
     0 Line Code Violations, 0 Path Code Violations,
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins,
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
```

Table 6-18 describes the **show controllers t1** display fields.

**Table 6-18 Show Controllers T1 Field Descriptions**

| Field | Description |
|---|---|
| T1 0/0 is up. | The T1 controller 0 in slot 0 is operating. The controller's state can be up, down, administratively down. Loopback conditions are shown by (Locally looped) or (Remotely Looped). |

| Field | Description |
|---|---|
| No alarms detected. | Any alarms detected by the controller are displayed here. Possible alarms are as follows:<br>Transmitter is sending remote alarm.<br>Transmitter is sending AIS.<br>Receiver has loss of signal.<br>Receiver is getting AIS.<br>Receiver has loss of frame.<br>Receiver has remote alarm.<br>Receiver has no alarms. |
| Data in current interval (725 seconds elapsed) | Shows the current accumulation period, which rolls into the 24 hour accumulation every 15 minutes. Accumulation period is from 1 to 900 seconds. The oldest 15 minute period falls off the back of the 24-hr accumulation buffer |
| Line Code Violations | Indicates the occurrence of either a Bipolar Violation (BPV) or Excessive Zeroes (EXZ) error event. |
| Path Code Violations | Indicates a frame synchronization bit error in the D4 and E1-noCRC formats, or a CRC error in the ESF and E1-CRC formats. |
| Slip Secs | Indicates the replication or deletion of the payload bits of a DS1 frame. A slip may be performed when there is a difference between the timing of a synchronous receiving terminal and the received signal. |
| Fr Loss Secs | Indicates the number of seconds an Out Of Frame (OOF) error is detected. |
| Line Err Secs | Line Errored Seconds (LES) is a second in which one or more Line Code Violation errors are detected. |
| Degraded Mins | A Degraded Minute is one in which the estimated error rate exceeds 1E-6 but does not exceed 1E-3. |
| Errored Secs | In ESF and E1-CRC links, an Errored Second is a second in which one of the following are detected: one or more Path Code Violations; one or more Out of Frame defects; one or more Controlled Slip events; a detected AIS defect.<br>For D4 and E1-noCRC links, the presence of Bipolar Violations also triggers an Errored Second. |
| Bursty Err Secs | A second with fewer than 320 and more than 1 Path Coding Violation error, no Severely Errored Frame defects and no detected incoming AIS defects. Controlled slips are not included in this parameter. |
| Severely Err Secs | For ESF signals, a second with one of the following errors: 320 or more Path Code Violation errors; one or more Out of Frame defects; a detected AIS defect.<br>For E1-CRC signals, a second with one of the following errors: 832 or more Path Code Violation errors; one or more Out of Frame defects.<br>For E1-nonCRC signals, a second with 2048 Line Code Violations or more.<br>For D4 signals, a count of 1-second intervals with Framing Errors, or an Out of Frame defect, or 1544 Line Code Violations. |
| Unavail Secs | A count of the total number of seconds on the interface. |

# show controllers token

To display information about memory management, error counters, and the CSC-R, CSC-1R, CSC-2R, C2CTR, and CSC-R16 (or CSC-R16M) Token Ring interface cards or Token Ring Interface Processor (TRIP), in the case of the Cisco 7000 series, use the **show controllers token** privileged EXEC command.

**show controllers token**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Privileged EXEC

## Usage Guidelines

Depending on the board being used, the output can vary. This command also displays information that is proprietary to Cisco Systems. Thus, the information that **show controllers token** displays is of primary use to Cisco technical personnel. Information that is useful to users can be obtained with the **show interfaces tokenring** command, described later in this chapter.

## Sample Display

The following is sample output on the AGS+ from the **show controllers token** command:

```
Router# show controllers token

TR Unit 0 is board 0 - ring 0

 state 3, dev blk: 0x1D2EBC, mailbox: 0x2100010, sca: 0x2010000
   current address: 0000.3080.6f40, burned in address: 0000.3080.6f40
   current TX ptr: 0xBA8, current RX ptr: 0x800

   Last Ring Status: none

 Stats: soft:0/0, hard:0/0, sig loss:0/0
        tx beacon: 0/0, wire fault 0/0, recovery: 0/0
        only station: 0/0, remote removal: 0/0
   Bridge: local 3330, bnum 1, target 3583
     max_hops 7, target idb: 0x0, not local
   Interface failures: 0  -- Bkgnd Ints: 0
   TX shorts 0, TX giants 0

   Monitor state: (active)
     flags 0xC0, state 0x0, test 0x0, code 0x0, reason 0x0
 f/w ver: 1.0, chip f/w: '000000.ME31100', [bridge capable]
     SMT versions: 1.01 kernel, 4.02 fastmac
     ring mode: F00, internal enables:  SRB REM RPS CRS/NetMgr
     internal functional: 0000011A (0000011A), group: 00000000 (00000000)
     if_state: 1, ints: 0/0, ghosts: 0/0, bad_states: 0/0
     t2m fifo purges: 0/0
     t2m fifo current: 0, t2m fifo max: 0/0, proto_errs: 0/0
     ring: 3330, bridge num: 1, target: 3583, max hops: 7
Packet counts:
       receive total:  298/6197, small: 298/6197, large 0/0
               runts: 0/0, giants: 0/0
```

```
                  local: 298/6197, bridged: 0/0, promis: 0/0
                bad rif: 0/0, multiframe: 0/0
          ring num mismatch 0/0, spanning violations 0
          transmit total: 1/25, small: 1/25, large 0/0
                   runts: 0/0, giants: 0/0, errors 0/0
   bad fs: 0/0, bad ac: 0
   congested: 0/0, not present: 0/0
        Unexpected interrupts: 0/0,  last unexp. int: 0

        Internal controller counts:
      line errors: 0/0,  internal errors: 0/0
      burst errors: 0/0,  ari/fci errors: 0/0
      abort errors: 0/0, lost frame: 0/0
      copy errors: 0/0, rcvr congestion: 0/0
      token errors: 0/0, frequency errors: 0/0
      dma bus errors: -/-, dma parity errors: -/-
        Internal controller smt state:
      Adapter MAC:     0000.3080.6f40, Physical drop:     00000000
      NAUN Address:    0000.a6e0.11a6, NAUN drop:         00000000
      Last source:     0000.a6e0.11a6, Last poll:         0000.3080.6f40
      Last MVID:       0006,           Last attn code:    0006
      Txmit priority:  0006,           Auth Class:        7FFF
      Monitor Error:   0000,           Interface Errors:  FFFF
      Correlator:      0000,           Soft Error Timer:  00C8
      Local Ring:      0000,           Ring Status:       0000
      Beacon rcv type: 0000,           Beacon txmit type: 0000
      Beacon type:     0000,           Beacon NAUN:       0000.a6e0.11a6
```

Table 6-19 describes the fields shown in the following line of sample output:

```
TR Unit 0 is board 0 - ring 0
```

**Table 6-19 Show Controllers Token Field Descriptions—Part 1**

| Field | Description |
|---|---|
| TR Unit 0 | Unit number assigned to the Token Ring interface associated with this output. |
| is board 0 | Board number assigned to the Token Ring controller board associated with this interface. |
| ring 0 | Number of the Token Ring associated with this board. |

In the following output line, state 3 indicates the state of the board. The rest of this output line displays memory mapping that is of primary use to Cisco engineers.

```
state 3, dev blk: 0x1D2EBC, mailbox: 0x2100010, sca: 0x2010000
```

The following line also appears in **show interface token** output as the address and burned in address, respectively:

```
current address: 0000.3080.6f40, burned in address: 0000.3080.6f40
```

The following line of output displays buffer management pointers that change by board:

```
current TX ptr: 0xBA8, current RX ptr: 0x800
```

The following line of output indicates the ring status from the controller chip set. This information is used by LAN Network Manager:

```
Last Ring Status: none
```

The following lines of output show Token Ring statistics. See the Token Ring specification for more information.

```
Stats: soft:0/0, hard:0/0, sig loss:0/0
       tx beacon: 0/0, wire fault 0/0, recovery: 0/0
       only station: 0/0, remote removal: 0/0
```

The following line of output indicates that Token Ring communication has been enabled on the interface. If this line of output appears, the message “Source Route Bridge capable” should appear in the **show interfaces tokenring** display.

```
Bridge: local 3330, bnum 1, target 3583
```

Table 6-20 describes the fields shown in this line of sample output:

```
max_hops 7, target idb: 0x0, not local
```

**Table 6-20 Show Controllers Token Field Descriptions—Part 2**

| Field | Description |
|---|---|
| max_hops 7 | Maximum number of bridges. |
| target idb: 0x0 | Destination interface definition. |
| not local | Indicates whether the interface has been defined as a local or remote bridge. |

The following line of output is specific to the hardware:

```
Interface failures: 0  -- Bkgnd Ints: 0
```

In the following line of output, TX shorts are the number of packets the interface transmits that are discarded because they are smaller than the medium’s minimum packet size. TX giants are the number of packets the interface transmits that are discarded because they exceed the medium’s maximum packet size.

```
TX shorts 0, TX giants 0
```

The following line of output indicates the state of the controller. Possible values include active, failure, inactive, and reset:

```
Monitor state: (active)
```

The following line of output displays detailed information relating to the monitor state shown in the previous line of output. This information relates to the firmware on the controller. This information is relevant to Cisco engineers only if the monitor state is something other than active.

```
flags 0xC0, state 0x0, test 0x0, code 0x0, reason 0x0
```

Table 6-21 describes the fields in the following line of output:

```
f/w ver: 1.0 expr 0, chip f/w: '000000.ME31100', [bridge capable]
```

**Table 6-21 Show Controllers Token Field Descriptions—Part 3**

| Field | Description |
|---|---|
| f/w ver: 1.0 | Version of the Cisco firmware on the board. |

| Field | Description |
|---|---|
| chip f/w: '000000.ME31100' | Firmware on the chip set. |
| [bridge capable] | Interface has not been configured for bridging, but that it has that capability. |

The following line of output displays the version numbers for the kernel and the accelerator microcode of the Madge firmware on the board; this firmware is the LLC interface to the chip set:

```
SMT versions: 1.01 kernel, 4.02 fastmac
```

The following line of output displays LAN Network Manager information that relates to ring status:

```
ring mode: F00, internal enables:  SRB REM RPS CRS/NetMgr
```

The following line of output corresponds to the functional address and the group address shown in **show interfaces tokenring** output:

```
internal functional: 0000011A (0000011A), group: 00000000 (00000000)
```

The following line of output displays interface board state information that is proprietary to Cisco Systems:

```
if_state: 1, ints: 0/0, ghosts: 0/0, bad_states: 0/0
```

The following output lines display information that is proprietary to Cisco Systems. Cisco engineers use this information for debugging purposes.

```
t2m fifo purges: 0/0
t2m fifo current: 0, t2m fifo max: 0/0, proto_errs: 0/0
```

Each of the fields in the following line of output maps to a field in the **show source bridge** display, as follows: ring maps to srn; bridge num maps to bn; target maps to trn; and max hops maps to max:

```
ring: 3330, bridge num: 1, target: 3583, max hops: 7
```

In the following lines of output, the number preceding the slash (/) indicates the count since the value was last displayed; the number following the slash (/) indicates count since the system was last booted:

```
Packet counts:
       receive total:  298/6197, small: 298/6197, large 0/0
```

In the following line of output, the number preceding the slash (/) indicates the count since the value was last displayed; the number following the slash (/) indicates count since the system was last booted. The runts and giants values that appear here correspond to the runts and giants values that appear in **show interfaces tokenring** output.

```
runts: 0/0, giants: 0/0
```

The following lines of output are receiver-specific information that Cisco engineers can use for debugging purposes:

```
              local: 298/6197, bridged: 0/0, promis: 0/0
            bad rif: 0/0, multiframe: 0/0
      ring num mismatch 0/0, spanning violations 0
      transmit total: 1/25, small: 1/25, large 0/0
               runts: 0/0, giants: 0/0, errors 0/0
```

The following output lines include very specific statistics that are not relevant in most cases, but exist for historical purposes. In particular, the internal errors, burst errors, ari/fci, abort errors, copy errors, frequency errors, dma bus errors, and dma parity errors fields are not relevant.

```
     Internal controller counts:
line errors: 0/0,  internal errors: 0/0
burst errors: 0/0,  ari/fci errors: 0/0
abort errors: 0/0, lost frame: 0/0
copy errors: 0/0, rcvr congestion: 0/0
token errors: 0/0, frequency errors: 0/0
dma bus errors: -/-, dma parity errors: -/-
```

The following lines of output are low-level Token Ring interface statistics relating to the state and status of the Token Ring with respect to all other Token Rings on the line:

```
  Internal controller smt state:
Adapter MAC:     0000.3080.6f40, Physical drop:     00000000
NAUN Address:    0000.a6e0.11a6, NAUN drop:         00000000
Last source:     0000.a6e0.11a6, Last poll:         0000.3080.6f40
Last MVID:       0006,           Last attn code:    0006
Txmit priority:  0006,           Auth Class:        7FFF
Monitor Error:   0000,           Interface Errors:  FFFF
Correlator:      0000,           Soft Error Timer:  00C8
Local Ring:      0000,           Ring Status:       0000
Beacon rcv type: 0000,           Beacon txmit type: 0000
```

## Sample Display

Sample output for the **show controllers token** command on the Cisco 7000 follows:

```
Router> show controllers token
Tokenring4/0: state administratively down
  current address: 0000.3040.8b4a, burned in address: 0000.3040.8b4a
  Last Ring Status: none
    Stats: soft: 0/0, hard: 0/0, sig loss: 0/0
           tx beacon: 0/0, wire fault 0/0, recovery: 0/0
           only station: 0/0, remote removal: 0/0
  Monitor state: (active), chip f/w: '000000........', [bridge capable]
    ring mode: 0”
    internal functional: 00000000 (00000000), group: 00000000 (00000000)
    internal addrs: SRB: 0000, ARB: 0000, EXB 0000, MFB: 0000
                    Rev: 0000, Adapter: 0000, Parms 0000
    Microcode counters:
      MAC giants 0/0, MAC ignored 0/0
      Input runts 0/0, giants 0/0, overrun 0/0
      Input ignored 0/0, parity 0/0, RFED 0/0
      Input REDI 0/0, null rcp 0/0, recovered rcp 0/0
      Input implicit abort 0/0, explicit abort 0/0
      Output underrun 0/0, tx parity 0/0, null tcp 0/0
      Output SFED 0/0, SEDI 0/0, abort 0/0
      Output False Token 0/0, PTT Expired 0/0
    Internal controller counts:
      line errors: 0/0,  internal errors: 0/0
      burst errors: 0/0,  ari/fci errors: 0/0
      abort errors: 0/0, lost frame: 0/0
      copy errors: 0/0, rcvr congestion: 0/0
      token errors: 0/0, frequency errors: 0/0
    Internal controller smt state:
      Adapter MAC:     0000.0000.0000, Physical drop:     00000000
      NAUN Address:    0000.0000.0000, NAUN drop:         00000000
      Last source:     0000.0000.0000, Last poll:         0000.0000.0000
      Last MVID:       0000,           Last attn code:    0000
      Txmit priority:  0000,           Auth Class:        0000
      Monitor Error:   0000,           Interface Errors:  0000
      Correlator:      0000,           Soft Error Timer:  0000
      Local Ring:      0000,           Ring Status:       0000
      Beacon rcv type: 0000,           Beacon txmit type: 0000
      Beacon type:     0000,           Beacon NAUN:       0000.0000.0000
```

```
Beacon drop:     00000000,       Reserved:          0000
Reserved2:       0000
```

Table 6-22 describes key **show controllers token** display fields.

**Table 6-22 Show Controllers Token Field Descriptions**

| Field | Description |
|---|---|
| Tokenring4/0 | Interface processor type, slot, and port. |
| Last Ring Status | Last abnormal ring condition. Can be any of the following:<br>Signal Loss<br>HW Removal<br>Remote Removal<br>Counter Overflow<br>Only station<br>Ring Recovery |

# show hub

To display information about the hub (repeater) on an Ethernet interface of a Cisco 2505 or Cisco 2507, use the **show hub** EXEC command.

**show hub** [**ether** *number* [*port* [*end-port*]]]

## Syntax Description

| | |
|---|---|
| **ether** | (Optional) Indicates that this is an Ethernet hub. |
| *number* | (Optional) Hub number, starting with 0. Since there is currently only one hub, this number is 0. |
| *port* | (Optional) Port number on the hub. On the Cisco 2505, port numbers range from 1 through 8. On the Cisco 2507, port numbers range from 1 through 16. If a second port number follows, then this port number indicates the beginning of a port range. |
| *end-port* | (Optional) Ending port number of a range. |

## Command Mode

EXEC

## Usage Guidelines

If you do not specify a port or port range for the **show hub** command, the command displays all ports (for example, ports 1 through 16 on a Cisco 2507) by default. Therefore, the commands **show hub**, **show hub ethernet 0**, and **show hub ethernet 0 1 16** all produce the same result.

If no ports are specified, the command displays some additional data about the internal port. The internal port is the hub's connection to Ethernet interface 0 inside the box. Ethernet interface 0 still exists; physical access to the interface is via the hub.

## Sample Displays

The following is sample output from the **show hub** command for hub 0, port 2 only:

```
Router# show hub ethernet 0 2

Port 2 of 16 is administratively down, link state is down
  0 packets input, 0 bytes
  0 errors with 0 collisions
     (0 FCS, 0 alignment, 0 too long,
      0 short, 0 runts, 0 late,
      0 very long, 0 rate mismatches)
  0 auto partitions, last source address (none)
  Last clearing of “show hub” counters never

Repeater information (Connected to Ethernet0)
  2792429 bytes seen with 18 collisions, 1 hub resets
  Version/device ID 0/1 (0/1)
  Last clearing of “show hub” counters never
```

The following is sample output from the **show hub** command for hub 0, all ports:

```
Router# show hub ethernet 0

Port 1 of 16 is administratively down, link state is up
  2458 packets input, 181443 bytes
  3 errors with 18 collisions
     (0 FCS, 0 alignment, 0 too long,
      0 short, 3 runts, 0 late,
      0 very long, 0 rate mismatches)
  0 auto partitions, last source address was 0000.0cff.e257
  Last clearing of “show hub” counters never
.
.
.
Port 16 of 16 is down, link state is down
  0 packets input, 0 bytes
  0 errors with 0 collisions
     (0 FCS, 0 alignment, 0 too long,
      0 short, 0 runts, 0 late,
      0 very long, 0 rate mismatches)
  0 auto partitions, last source address (none)
  Last clearing of “show hub” counters never

Repeater information (Connected to Ethernet0)
  2792429 bytes seen with 18 collisions, 1 hub resets
  Version/device ID 0/1 (0/1)
  Last clearing of “show hub” counters never

Internal Port (Connected to Ethernet0)
  36792 packets input, 4349525 bytes
  0 errors with 14 collisions
     (0 FCS, 0 alignment, 0 too long,
      0 short, 0 runts, 0 late,
      0 very long, 0 rate mismatches)
  0 auto partitions, last source address (none)
  Last clearing of “show hub” counters never
```

Table 6-23 describes significant fields show in the display.

**Table 6-23 Show Hub Field Descriptions**

| Field | Description |
|---|---|
| Port ... of ... is administratively down | Port number out of total ports; indicates whether the interface hardware is currently active, or down due to the following:<br>• The link-state test failed.<br>• The MAC address mismatched when source address configured.<br>• It has been taken down by an administrator. |
| link state is up | Indicates whether port has been disabled by the link-test function. If the link-test function is disabled by the user, nothing will be shown here. |
| packets input | Total number of error-free packets received by the system. |
| bytes | Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system. |

| Field | Description |
|---|---|
| errors | Sum of FCS, alignment, too long, short, runts, very long, and rate mismatches. |
| collisions | Number of messages retransmitted due to Ethernet collisions. |
| FCS | Counter for the number of frames detected on the port with an invalid frame check sequence. |
| alignment | Counter for the number of frames of valid length (64 bytes to 1518 bytes) that have been detected on the port with an FCS error and a framing error. |
| too long | Counter for the number of frames that exceed the maximum valid packet length of 1518 bytes. |
| short | Counter for the number of instances when activity is detected with duration less than 74-82 bit times. |
| runts | Number of packets that are discarded because they are smaller than the medium's minimum packet size. For example, any Ethernet packet that is less than 64 bytes is considered a runt. |
| late | Counter for the number of instances when a collision is detected after 480-565 bit times in the frame. |
| very long | Counter for the number of times the transmitter is active in excess of 4 ms to 7.5 ms. |
| rate mismatches | Counter for the number of occurrences when the frequency, or data rate of incoming signal is noticably different from the local transmit frequency. |
| auto partitions | Counter for the number of instances where the repeater has partitioned the port from the network. |
| last source address | Source address of last packet received by this port. Indicates "none" if no packets have been received since power on or a hub reset. |
| Last clearing of "show hub" counters | Elapsed time since **clear hub counters** command. Indicates "never" if counters have never been cleared. |
| Repeater information (Connected to Ethernet0) | Indicates that the following information is about the hub connected to the Ethernet interface shown. |
| ... bytes seen with ... collisions, ... hub resets | Hub resets is the number of times the hub has been reset by network management software or by the **clear hub** command. |
| Version/device ID 0/1 (0/1) | Hub hardware version. IMR+ version device of daughter board. |
| Internal Port (Connected to Ethernet0) | Set of counters for the internal AUI port connected to the Ethernet interface. |

## Related Command

**hub**

# show interfaces

Use the **show interfaces** EXEC command to display statistics for all interfaces configured on the router. The resulting output varies, depending on the network for which an interface has been configured.

**show interfaces** [*type* {*unit*}] [*first*] [*last*] [**accounting**]
**show interfaces** [*type slot*/*port*] [**accounting**] (for the Cisco 7000 series)

## Syntax Description

| | |
|---|---|
| *type unit* | (Optional) Specify that information for a particular interface controller be displayed. Allowed values for type include **async**, **bri0**, **ethernet**, **fddi**, **hssi**, **loopback**, **null**, **serial**, **tokenring**, and **tunnel**.<br>For the Cisco 7000 series, *type* can be **atm**, **ethernet**, **fddi**, **serial**, or **tokenring**.<br>The argument *unit* must match a port number on the selected interface controller. |
| *first last* | (Optional) The Cisco 2500 and 3000 support the ISDN Basic Rate Interface (BRI). The argument *first* can be either 1 or 2. The argument *last* can only be 2, indicating B channels 1 and 2. D-channel information is obtained by using the command without the optional arguments. |
| **accounting** | (Optional) Displays the number of packets of each protocol type that has been sent through the interface. You can show these numbers for all interfaces, or you can specify a specific *type* and *unit*. |
| *slot* | Specifies the backplane slot number and can be 0, 1, 2, 3, or 4. |
| *port* | Specifies the port number of the interface and can be 0, 1, 2, 3, 4, or 5 depending on the type of interface, as follows:<br>• AIP (ATM Interface Processor) 0<br>• EIP (Ethernet Interface Processor) 0, 1, 2, 3, 4, or 5<br>• FIP (FDDI Interface Processor) 0<br>• FSIP (Fast Serial Interface Processor) 0, 1, 2, 3, 4, 5, 6, or 7<br>• HIP (HSSI Interface Processor) 0<br>• TRIP (Token Ring Interface Processor) 0, 1, 2, or 3 |

## Command Mode

EXEC

## Usage Guidelines

The **show interfaces** command displays statistics for the network interfaces. The resulting display on the Cisco 7000 series will show the interface processors in slot order. If you add interface processors after booting the system, they will appear at the end of the list, in the order in which they were inserted.

If you use the **show interfaces** command on the Cisco 7000 series without the *slot/port* arguments, information for all interface types will be shown. For example, if you type **show interfaces ethernet** you will receive information for all ethernet, serial, Token Ring, and FDDI interfaces. Only by adding the *type slot/port* argument can you specify a particular interface.

If you enter a **show interfaces** command for an interface type that has been removed from the router, interface statistics will be displayed accompanied by the following text: "Hardware has been removed."

You will use the **show interfaces** command frequently while configuring and monitoring routers. The various forms of the **show interfaces** commands are described in detail in the sections immediately following this command.

## Sample Display

The following is sample output from the **show interfaces** command. Because your display will depend on the type and number of interface cards in your router, only a portion of the display is shown.

```
Router# show interfaces
Ethernet 0 is up, line protocol is up
  Hardware is MCI Ethernet, address is 0000.0c00.750c (bia 0000.0c00.750c)
  Internet address is 131.108.28.8, subnet mask is 255.255.255.0
  MTU 1500 bytes, BW 10000 Kbit, DLY 100000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 4:00:00
  Last input 0:00:00, output 0:00:00, output hang never
  Last clearing of "show interface" counters 0:00:00
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 2000 bits/sec, 4 packets/sec
     1127576 packets input, 447251251 bytes, 0 no buffer
     Received 354125 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     5332142 packets output, 496316039 bytes, 0 underruns
     0 output errors, 432 collisions, 0 interface resets, 0 restarts
---More---
```

## Sample Display with Accounting Option

To display the number of packets of each protocol type that have been sent through all configured interfaces, use the **show interfaces accounting** EXEC command. When you use the **accounting** option, only the accounting statistics are displayed.

---

**Note** Except for protocols that are encapsulated inside other protocols, such as IP over X.25, the accounting option also shows the total of all bytes sent and received, including the MAC header. For example, it totals the size of the Ethernet packet or the size of a packet that includes HDLC encapsulation.

---

Table 6-24 lists the protocols for which per-packet accounting information is kept.

**Table 6-24 Per-Packet Counted Protocols**

| Protocol | Notes |
|---|---|
| Apollo | No note. |
| AppleTalk | No note. |
| ARP | For IP, Apollo, Frame Relay, SMDS. |
| CLNS | No note. |
| DEC MOP | The routers use MOP packets to advertise their existence to DEC machines that use the MOP protocol. A router periodically broadcasts MOP packets to identify itself as a MOP host. This results in MOP packets being counted, even when DECnet is not being actively used. |
| DECnet | No note. |
| HP Probe | No note. |
| IP | No note. |
| LAN Manager | LAN Network Manager and IBM Network Manager. |
| Novell | No note. |
| Serial Tunnel | SDLC. |
| Spanning Tree | No note. |
| SR Bridge | No note. |
| Transparent Bridge | No note. |
| VINES | No note. |
| XNS | No note. |

## Sample Display

The following is sample output from the **show interfaces accounting** command:

```
Router# show interfaces accounting

Interface TokenRing0 is disabled

Ethernet0
                Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                      IP     873171  735923409      34624    9644258
                  Novell     163849   12361626      57143    4272468
                 DEC MOP          0          0          1         77
                     ARP      69618    4177080       1529      91740
Interface Serial0 is disabled

Ethernet1
                Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                      IP          0          0         37      11845
                  Novell          0          0       4591     275460
                 DEC MOP          0          0          1         77
                     ARP          0          0          7        420

Interface Serial1 is disabled
Interface Ethernet2 is disabled
Interface Serial2 is disabled
Interface Ethernet3 is disabled
Interface Serial3 is disabled
Interface Ethernet4 is disabled
Interface Ethernet5 is disabled
Interface Ethernet6 is disabled
```

```
Interface Ethernet7 is disabled
Interface Ethernet8 is disabled
Interface Ethernet9 is disabled

Fddi0
                Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                  Novell          0          0        183      11163
                     ARP          1         49          0          0
```

When the output indicates an interface is “disabled,” the router has received excessive errors (over 5000 in a keepalive period).

# show interfaces async

Use the **show interfaces async** privileged EXEC command to display information about the serial interface.

**show interfaces async**[*unit*] [**accounting**]

## Syntax Description

| | |
|---|---|
| *unit* | (Optional) Must be 1. |
| **accounting** | (Optional) Displays the number of packets of each protocol type that have been sent through the interface. |

## Command Mode

Privileged EXEC

## Sample Display

The following is sample output from the **show interfaces async** command:

```
Router# show interfaces async 1

Async 1 is up, line protocol is up
   Hardware is Async Serial
Internet address is 1.0.0.1, subnet mask is 255.0.0.0
MTU 1500 bytes, BW 9 Kbit, DLY 100000 usec, rely 255/255, load 56/255
Encapsulation SLIP, keepalive set (0 sec)
Last input 0:00:03, output 0:00:03, output hang never
Last clearing of “show interface” counters never
Output queue 0/3, 2 drops; input queue 0/0, 0 drops
Five minute input rate 0 bits/sec, 1 packets/sec
Five minute output rate 2000 bits/sec, 1 packets/sec
273 packets input, 13925 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
221 packets output, 41376 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets, 0 restarts
0 carrier transitions
```

Table 6-25 describes the fields shown in the display.

**Table 6-25 Show Interfaces Async Field Descriptions**

| Field | Description |
|---|---|
| Async... is {up \| down}<br>...is administratively down | Indicates whether the interface hardware is currently active (whether carrier detect is present) and if it has been taken down by an administrator. |
| line protocol<br>is {up \| down \|<br>administratively down} | Indicates whether the software processes that handle the line protocol think the line is usable (that is, whether keepalives are successful). |
| Hardware is | Hardware type. |
| Internet address is | Internet address and subnet mask, followed by packet size. |

| Field | Description |
|---|---|
| MTU | Maximum Transmission Unit of the interface. |
| BW | Bandwidth of the interface in kilobits per second. |
| DLY | Delay of the interface in microseconds. |
| rely | Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over 5 minutes. |
| load | Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over 5 minutes. The calculation uses the value from the **bandwidth** interface configuration command. |
| Encapsulation | Encapsulation method assigned to interface. |
| keepalive | Indicates whether keepalives are set or not. |
| Last input | Number of hours, minutes, and seconds since the last packet was successfully received by an interface. Useful for knowing when a dead interface failed. |
| Last output | Number of hours, minutes, and seconds since the last packet was successfully transmitted by an interface. |
| output hang | Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the “last” fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed. |
| Last clearing | The time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared.<br>*** indicates the elapsed time is too large to be displayed.<br>0:00:00 indicates the counters were cleared more than $2^{31}$ms (and less than $2^{32}$ms) ago. |
| Output queue, drops<br>input queue, drops | Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue. |
| Five minute input rate,<br>Five minute output rate | Average number of bits and packets transmitted per second in the last 5 minutes. |
| packets input | Total number of error-free packets received by the system. |
| bytes input | Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system. |
| no buffers | Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events. |
| broadcasts | Total number of broadcast or multicast packets received by the interface. |
| runts | Number of packets that are discarded because they are smaller than the medium’s minimum packet size. |
| giants | Number of packets that are discarded because they exceed the medium’s maximum packet size. |

| Field | Description |
| --- | --- |
| input errors | Total number of no buffer, runts, giants, CRCs, frame, overrun, ignored, and abort counts. Other input-related errors can also increment the count, so that this sum may not balance with the other counts. |
| CRC | Cyclic redundancy checksum generated by the originating LAN station or far end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRC's is usually the result of collisions or a station transmitting bad data. On a serial link, CRC's usually indicate noise, gain hits or other transmission problems on the data link. |
| frame | Number of packets received incorrectly having a CRC error and a noninteger number of octets. On a serial line, this is usually the result of noise or other transmission problems. |
| overrun | Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data. |
| ignored | Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be incremented. |
| abort | Illegal sequence of one bits on a serial interface. This usually indicates a clocking problem between the serial interface and the data link equipment. |
| packets output | Total number of messages transmitted by the system. |
| bytes | Total number of bytes, including data and MAC encapsulation, transmitted by the system. |
| output errors | Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this may not balance with the sum of the enumerated output errors, as some datagrams may have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories. |
| interface resets | Number of times an interface has been completely reset. This can happen if packets queued for transmission were not sent within several seconds. On a serial line, this can be caused by a malfunctioning modem that is not supplying the transmit clock signal, or by a cable problem. If the system notices that the carrier detect line of a serial interface is up, but the line protocol is down, it periodically resets the interface in an effort to restart it. Interface resets can also occur when an interface is looped back or shut down. |
| restarts | Number of times the controller was restarted because of errors. |
| carrier transitions | Number of times the carrier detect signal of a serial interface has changed state. Indicates modem or line problems if the carrier detect line is changing state often. |
| Protocol | Protocol that is operating on the interface. |
| Pkts In | Number of packets received for that protocol. |
| Chars In | Number of characters received for that protocol. |

| Field | Description |
|---|---|
| Pkts Out | Number of packets transmitted for that protocol. |
| Chars Out | Number of characters transmitted for that protocol. |

## Sample Display with Accounting Option

The following is a sample display from the **show interfaces async accounting** command:

```
Router# show interfaces async 0 accounting

Async 0
   Protocol  Pkts In   Chars In  Pkts Out   Chars Out
   IP        7344      4787842   1803       1535774
   DEC MOP   0         0         127        9779
   ARP       7         420       39         2340
```

The **show line** and **show slip** commands can also be useful in monitoring asynchronous interfaces.

# show interfaces atm

Use the **show interfaces atm** privileged EXEC command to display information about the ATM interface.

**show interfaces atm** [*slot*/*port*]

## Syntax Description

| | |
|---|---|
| *slot*/*port* | (Optional) Slot on the Cisco 7000 can be 0, 1, 2, 3, or 4. On the Cisco 7010, slot can be 0, 1, or 2. Port must be 0. |

## Command Mode

Privileged EXEC

## Sample Display

The following is sample output from the **show interfaces atm** command:

```
Router# show interfaces atm4/0

ATM4/0 is up, line protocol is up
  Hardware is cxBus ATM
  Internet address is 131.108.97.165, subnet mask is 255.255.255.0
  MTU 4470 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 1/255
  Encapsulation ATM, loopback not set, keepalive set (10 sec)
  Encapsulation(s): AAL5, PVC mode
  256 TX buffers, 256 RX buffers, 1024 Maximum VCs, 1 Current VCs
  Signalling vc = 1, vpi = 0, vci = 5
  ATM NSAP address: BC.CDEF.01.234567.890A.BCDE.F012.3456.7890.1234.13
  Last input 0:00:05, output 0:00:05, output hang never
  Last clearing of "show interface" counters never
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
     144 packets input, 3148 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     154 packets output, 4228 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets, 0 restarts
```

Table 6-26 describes the fields shown in the display.

**Table 6-26 Show Interfaces ATM Field Descriptions**

| Field | Description |
|---|---|
| ATM... is {up \| down}<br>...is administratively down | Indicates whether the interface hardware is currently active (whether carrier detect is present) and if it has been taken down by an administrator. |
| line protocol is {up \| down \| administratively down} | Indicates whether the software processes that handle the line protocol think the line is usable (that is, whether keepalives are successful). |
| Hardware is | Hardware type. |
| Internet address is | Internet address and subnet mask. |

| Field | Description |
|---|---|
| MTU | Maximum Transmission Unit of the interface. |
| BW | Bandwidth of the interface in kilobits per second. |
| DLY | Delay of the interface in microseconds. |
| rely | Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over 5 minutes. |
| load | Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over 5 minutes. The calculation uses the value from the **bandwidth** interface configuration command. |
| Encapsulation | Encapsulation method assigned to interface. |
| Encapsulation(s) | AAL5, PVC or SVC mode. |
| TX buffers | Number of buffers configured with the **atm txbuff** command. |
| RX buffers | Number of buffers configured with the **atm rxbuff** command. |
| Maximum VCs | Maximum number of virtual circuits. |
| Current VCs | Current number of virtual circuits. |
| Signaling VC | Number of the signaling PVC. |
| vpi | Virtual path identifier number. |
| vci | Virtual channel identifier number. |
| ATM NSAP address | NSAP address of the ATM interface. |
| keepalive | Indicates whether keepalives are set or not. |
| Last input | Number of hours, minutes, and seconds since the last packet was successfully received by an interface. Useful for knowing when a dead interface failed. |
| Last output | Number of hours, minutes, and seconds since the last packet was successfully transmitted by an interface. |
| output hang | Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the “last” fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed. |
| Last clearing | The time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared.<br>*** indicates the elapsed time is too large to be displayed.<br>0:00:00 indicates the counters were cleared more than $2^{31}$ms (and less than $2^{32}$ms) ago. |
| Output queue, drops<br>input queue, drops | Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue. |
| Five minute input rate,<br>Five minute output rate | Average number of bits and packets transmitted per second in the last 5 minutes. |
| packets input | Total number of error-free packets received by the system. |
| bytes input | Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system. |

| Field | Description |
|---|---|
| no buffer | Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events. |
| broadcasts | Total number of broadcast or multicast packets received by the interface. |
| runts | Number of packets that are discarded because they are smaller than the medium's minimum packet size. |
| giants | Number of packets that are discarded because they exceed the medium's maximum packet size. |
| input errors | Total number of no buffer, runts, giants, CRCs, frame, overrun, ignored, and abort counts. Other input-related errors can also increment the count, so that this sum may not balance with the other counts. |
| CRC | Cyclic redundancy checksum generated by the originating LAN station or far end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRC's is usually the result of collisions or a station transmitting bad data. On a serial link, CRC's usually indicate noise, gain hits or other transmission problems on the data link. |
| frame | Number of packets received incorrectly having a CRC error and a noninteger number of octets. |
| overrun | Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data. |
| ignored | Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be incremented. |
| abort | Illegal sequence of one bits on a serial interface. This usually indicates a clocking problem between the serial interface and the data link equipment. |
| packets output | Total number of messages transmitted by the system. |
| bytes | Total number of bytes, including data and MAC encapsulation, transmitted by the system. |
| underruns | Number of times that the transmitter has been running faster than the router can handle. This may never be reported on some interfaces. |
| output errors | Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this may not balance with the sum of the enumerated output errors, as some datagrams may have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories. |

| Field | Description |
|---|---|
| interface resets | Number of times an interface has been completely reset. This can happen if packets queued for transmission were not sent within several seconds. On a serial line, this can be caused by a malfunctioning modem that is not supplying the transmit clock signal, or by a cable problem. If the system notices that the carrier detect line of a serial interface is up, but the line protocol is down, it periodically resets the interface in an effort to restart it. Interface resets can also occur when an interface is looped back or shut down. |
| restarts | Number of times the controller was restarted because of errors. |

# show interfaces ethernet

Use the **show interfaces ethernet** privileged EXEC command to display information about an Ethernet interface on the router.

**show interfaces ethernet** *unit* [**accounting**]
**show interfaces ethernet** [*slot/port*] [**accounting**] (for the Cisco 7000 series)

## Syntax Description

| | |
|---|---|
| *unit* | Must match a port number on the selected interface. |
| **accounting** | (Optional) Displays the number of packets of each protocol type that have been sent through the interface. |
| *slot* | (Optional) On the Cisco 7000 series, slot location of the interface processor. |
| *port* | (Optional) On the Cisco 7000 series, port number on interface. |

## Command Mode

Privileged EXEC

## Usage Guidelines

If you do not provide values for the argument *unit* (or *slot* and *port* on the Cisco 7000 series), the command will display statistics for all network interfaces. The optional keyword **accounting** displays the number of packets of each protocol type that have been sent through the interface.

## Sample Display

The following is sample output from the **show interfaces** command for the Ethernet 0 interface:

```
Router# show interfaces ethernet 0

Ethernet 0 is up, line protocol is up
   Hardware is MCI Ethernet, address is aa00.0400.0134 (bia 0000.0c00.4369)
   Internet address is 131.108.1.1, subnet mask is 255.255.255.0
   MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
   Encapsulation ARPA, loopback not set, keepalive set (10 sec)
   ARP type: ARPA, PROBE, ARP Timeout 4:00:00
   Last input 0:00:00, output 0:00:00, output hang never
   Output queue 0/40, 0 drops; input queue 0/75, 2 drops
   Five minute input rate 61000 bits/sec, 4 packets/sec
   Five minute output rate 1000 bits/sec, 2 packets/sec
       2295197 packets input, 305539992 bytes, 0 no buffer
       Received 1925500 broadcasts, 0 runts, 0 giants
       3 input errors, 3 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
       0 input packets with dribble condition detected
       3594664 packets output, 436549843 bytes, 0 underruns
       8 output errors, 1790 collisions, 10 interface resets, 0 restarts
```

Table 6-27 describes significant fields shown in the display.

**Table 6-27 Show Interfaces Ethernet Field Descriptions**

| Field | Description |
|---|---|
| Ethernet ... is up<br>...is administratively down | Indicates whether the interface hardware is currently active and if it has been taken down by an administrator. “Disabled” indicates the router has received over 5000 errors in a keepalive interval, which is 10 seconds by default. |
| line protocol<br>is {up \| down \|<br>administratively down} | Indicates whether the software processes that handle the line protocol believe the interface is usable (that is, whether keepalives are successful) or if it has been taken down by an administrator. |
| Hardware | Hardware type (for example, MCI Ethernet, SCI, cBus Ethernet) and address. |
| Internet address | Internet address followed by subnet mask. |
| MTU | Maximum Transmission Unit of the interface. |
| BW | Bandwidth of the interface in kilobits per second. |
| DLY | Delay of the interface in microseconds. |
| rely | Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over 5 minutes. |
| load | Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over 5 minutes. |
| Encapsulation | Encapsulation method assigned to interface. |
| ARP type: | Type of Address Resolution Protocol assigned. |
| loopback | Indicates whether loopback is set or not. |
| keepalive | Indicates whether keepalives are set or not. |
| Last input | Number of hours, minutes, and seconds since the last packet was successfully received by an interface. Useful for knowing when a dead interface failed. |
| Last output | Number of hours, minutes, and seconds since the last packet was successfully transmitted by an interface. |
| output | Number of hours, minutes, and seconds since the last packet was successfully transmitted by the interface. Useful for knowing when a dead interface failed. |
| output hang | Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the “last” fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed. |
| Last clearing | Time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared.<br>*** indicates the elapsed time is too large to be displayed.<br>0:00:00 indicates the counters were cleared more than $2^{31}$ms (and less than $2^{32}$ms) ago. |
| Output queue, input queue, drops | Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue. |

| Field | Description |
|---|---|
| Five minute input rate,<br>Five minute output rate | Average number of bits and packets transmitted per second in the last 5 minutes. If the interface is not in promiscuous mode, it senses network traffic it sends and receives (rather than all network traffic).<br><br>The 5-minute input and output rates should be used only as an approximation of traffic per second during a given 5-minute period. These rates are exponentially weighted averages with a time constant of 5 minutes. A period of four time constants must pass before the average will be within two percent of the instantaneous rate of a uniform stream of traffic over that period. |
| packets input | Total number of error-free packets received by the system. |
| bytes input | Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system. |
| no buffers | Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events. |
| Received ... broadcasts | Total number of broadcast or multicast packets received by the interface. |
| runts | Number of packets that are discarded because they are smaller than the medium's minimum packet size. For instance, any Ethernet packet that is less than 64 bytes is considered a runt. |
| giants | Number of packets that are discarded because they exceed the medium's maximum packet size. For example, any Ethernet packet that is greater than 1,518 bytes is considered a giant. |
| input error | Includes runts, giants, no buffer, CRC, frame, overrun, and ignored counts. Other input-related errors can also cause the input errors count to be increased, and some datagrams may have more than one error; therefore, this sum may not balance with the sum of enumerated input error counts. |
| CRC | Cyclic redundancy checksum generated by the originating LAN station or far-end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRCs is usually the result of collisions or a station transmitting bad data. |
| frame | Number of packets received incorrectly having a CRC error and a noninteger number of octets. On a LAN, this is usually the result of collisions or a malfunctioning Ethernet device. |
| overrun | Number of times the receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data. |
| ignored | Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be increased. |
| input packets with dribble condition detected | Dribble bit error indicates that a frame is slightly too long. This frame error counter is incremented just for informational purposes; the router accepts the frame. |

| Field | Description |
|---|---|
| packets output | Total number of messages transmitted by the system. |
| bytes | Total number of bytes, including data and MAC encapsulation, transmitted by the system. |
| underruns | Number of times that the transmitter has been running faster than the router can handle. This may never be reported on some interfaces. |
| output errors | Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this may not balance with the sum of the enumerated output errors, as some datagrams may have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories. |
| collisions | Number of messages retransmitted due to an Ethernet collision. This is usually the result of an overextended LAN (Ethernet or transceiver cable too long, more than two repeaters between stations, or too many cascaded multiport transceivers). A packet that collides is counted only once in output packets. |
| interface resets | Number of times an interface has been completely reset. This can happen if packets queued for transmission were not sent within several seconds. On a serial line, this can be caused by a malfunctioning modem that is not supplying the transmit clock signal, or by a cable problem. If the system notices that the carrier detect line of a serial interface is up, but the line protocol is down, it periodically resets the interface in an effort to restart it. Interface resets can also occur when an interface is looped back or shut down. |
| restarts | Number of times a Type 2 Ethernet controller was restarted because of errors. |

## Sample Display on Cisco 7000

The following sample output illustrates the **show interfaces ethernet** command on the Cisco 7000:

```
Router> show interfaces ethernet 4/2

Ethernet4/2 is up, line protocol is up
  Hardware is cxBus Ethernet, address is 0000.0c02.d0ce (bia 0000.0c02.d0ce)
  Internet address is 131.108.7.1, subnet mask is 255.255.255.0
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 4:00:00
  Last input 0:00:00, output 0:00:09, output hang never
  Last clearing of “show interface” counters 0:56:40
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 3000 bits/sec, 4 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
     4961 packets input, 715381 bytes, 0 no buffer
     Received 2014 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     567 packets output, 224914 bytes, 0 underruns
     0 output errors, 168 collisions, 0 interface resets, 0 restarts
```

### Sample Display with Accounting Option

The following is sample output from the **show interfaces ethernet** command with the **accounting** option on the Cisco 7000:

```
Router# show interfaces ethernet 4/2 accounting

Ethernet4/2
       Protocol    Pkts In   Chars In   Pkts Out  Chars Out
             IP       7344    4787842       1803    1535774
      Appletalk      33345    4797459      12781    1089695
        DEC MOP          0          0        127       9779
            ARP          7        420         39       2340
```

# show interfaces fddi

Use the **show interfaces fddi** EXEC command to display information about the FDDI interface.

**show interfaces fddi** *unit* [**accounting**]
**show interfaces fddi** [*slot*/*port*] [**accounting**]  (for the Cisco 7000 series)

## Syntax Description

| | |
|---|---|
| *unit* | Must match a port number on the selected interface. |
| **accounting** | (Optional) Displays the number of packets of each protocol type that have been sent through the interface. |
| *slot* | (Optional) On the Cisco 7000 series, slot location of the interface processor. |
| *port* | (Optional) On the Cisco 7000 series, port number on interface. |

## Command Mode

EXEC

## Sample Displays

The following is a sample partial display of FDDI-specific data from the **show interfaces fddi** command:

```
Router> show interfaces fddi 0

Fddi0 is up, line protocol is up
  Hardware is cBus Fddi, address is 0000.0c06.8de8 (bia 0000.0c06.8de8)
  Internet address is 131.108.33.9, subnet mask is 255.255.255.0
  MTU 4470 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 1/255
  Encapsulation SNAP, loopback not set, keepalive not set
  ARP type: SNAP, ARP Timeout 4:00:00
  Phy-A state is  active, neighbor is   B, cmt signal bits 008/20C, status ILS
  Phy-B state is connect, neighbor is unk, cmt signal bits 20C/000, status QLS
  ECM is insert, CFM is c_wrap_a, RMT is ring_op
  token rotation 5000 usec, ring operational 1d01
  Upstream neighbor 0000.0c06.8b7d, downstream neighbor 0000.0c06.8b7d
  Last input 0:00:08, output 0:00:08, output hang never
  Last clearing of "show interface" counters never
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 5000 bits/sec, 1 packets/sec
  Five minute output rate 76000 bits/sec, 51 packets/sec
     852914 packets input, 205752094 bytes, 0 no buffer
     Received 126752 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     8213126 packets output, 616453062 bytes, 0 underruns
     0 output errors, 0 collisions, 4 interface resets, 0 restarts
     5 transitions, 0 traces
```

The following is a sample partial display of FDDI-specific data from the **show interfaces fddi** command on a Cisco 7000:

```
Router> show interfaces fddi 3/0

Fddi3/0 is up, line protocol is up
  Hardware is cxBus Fddi, address is 0000.0c02.adf1 (bia 0000.0c02.adf1)
  Internet address is 131.108.33.14, subnet mask is 255.255.255.0
  MTU 4470 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 1/255
  Encapsulation SNAP, loopback not set, keepalive not set
  ARP type: SNAP, ARP Timeout 4:00:00
  Phy-A state is  active, neighbor is   B, cmt signal bits 008/20C, status ILS
  Phy-B state is  active, neighbor is   A, cmt signal bits 20C/008, status ILS
  ECM is in, CFM is thru, RMT is ring_op
  Token rotation 5000 usec, ring operational 21:32:34
  Upstream neighbor 0000.0c02.ba83, downstream neighbor 0000.0c02.ba83
  Last input 0:00:05, output 0:00:00, output hang never
  Last clearing of “show interface” counters 0:59:10
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 69000 bits/sec, 44 packets/sec
  Five minute output rate 0 bits/sec, 1 packets/sec
     113157 packets input, 21622582 bytes, 0 no buffer
     Received 276 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     4740 packets output, 487346 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets, 0 restarts
     0 transitions, 2 traces, 3 claims, 2 beacons
```

The following is an example that includes the **accounting** option. When you use the **accounting** option, only the accounting statistics are displayed.

```
Router> show interfaces fddi 3/0 accounting

Fddi3/0
       Protocol    Pkts In   Chars In   Pkts Out  Chars Out
             IP       7344    4787842       1803    1535774
      Appletalk      33345    4797459      12781    1089695
        DEC MOP          0          0        127       9779
            ARP          7        420         39       2340
```

Table 6-28 describes the **show interfaces fddi** display fields.

**Table 6-28 Show Interfaces FDDI Field Descriptions**

| Field | Description |
|---|---|
| Fddi is {up \|down}<br>...is administratively down | Gives the interface processor unit number and tells whether the interface hardware is currently active and can transmit and receive or if it has been taken down by an administrator. “Disabled” indicates the router has received over 5000 errors in a keepalive interval, which is 10 seconds by default. |
| line protocol is {up \| down \| administratively down} | Indicates whether the interface hardware is currently active and can transmit and receive or if it has been taken down by an administrator. |
| Hardware | Provides the hardware type, followed by the hardware address. |
| Internet address | IP address, followed by subnet mask. |
| MTU | Maximum Transmission Unit of the interface. |
| BW | Bandwidth of the interface in kilobits per second. |

| | |
|---|---|
| DLY | Delay of the interface in microseconds. |
| rely | Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over 5 minutes. |
| load | Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over 5 minutes. |
| Encapsulation | Encapsulation method assigned to interface. |
| loopback | Indicates whether or not loopback is set. |
| keepalive | Indicates whether or not keepalives are set. |
| ARP type: | Type of Address Resolution Protocol assigned. |
| Phy-{A \| B} | Lists the state the Physical A or Physical B connection is in; one of: off, active, trace, connect, next, signal, join, verify, or break. |
| neighbor | State of the neighbor:<br>• A—Indicates that the CMT process has established a connection with its neighbor. The bits received during the CMT signaling process indicate that the neighbor is a Physical A type dual-attachment station or concentrator that attaches to the primary ring IN and the secondary ring OUT when attaching to the dual ring.<br>• S—Indicates that the CMT process has established a connection with its neighbor and that the bits received during the CMT signaling process indicate that the neighbor is one Physical type in a single-attached station (SAS).<br>•B—Indicates that the CMT process has established a connection with its neighbor and that the bits received during the CMT signaling process indicate that the neighbor is a Physical B dual-attached station or concentrator that attaches to the secondary ring IN and the primary ring OUT when attaching to the dual ring.<br>• M—Indicates that the CMT process has established a connection with its neighbor and that the bits received during the CMT signaling process indicate that the router’s neighbor is a Physical M-type concentrator that serves as a Master to a connected station or concentrator.<br>• unk—Indicates that the network server has not completed the CMT process, and as a result, does not know about its neighbor. See the section “Setting Bit Control” for an explanation of the bit patterns. |
| cmt signal bits | Shows the transmitted/received CMT bits. The transmitted bits are 0x008 for a Physical A type and 0x20C for Physical B type. The number after the slash (/) is the received signal bits. If the connection is not active, the received bits are zero (0); see the line beginning Phy-B earlier in this display. |

| | |
|---|---|
| status | Status value displayed is the actual status on the fiber. The FDDI standard defines the following values:<br>• LSU—Line State Unknown, the criteria for entering or remaining in any other line state have not been met.<br>• NLS—Noise Line State is entered upon the occurrence of 16 potential noise events without satisfying the criteria for entry into another line state.<br>• MLS—Master Line State is entered upon the reception of eight or nine consecutive HQ or QH symbol pairs.<br>• ILS—Idle Line State is entered upon receipt of four or five idle symbols.<br>• HLS—Halt Line State is entered upon the receipt of 16 or 17 consecutive H symbols.<br>• QLS—Quiet Line State is entered upon the receipt of 16 or 17 consecutive Q symbols or when carrier detect goes low.<br>• ALS—Active Line State is entered upon receipt of a JK symbol pair when carrier detect is high.<br>• OVUF—Elasticity buffer Overflow/Underflow. The normal states for a connected Physical type are ILS or ALS. If the report displays the QLS status, this indicates that the fiber is disconnected from Physical B, or that it is not connected to another Physical type, or that the other station is not running. |
| Off | Indicates that the CMT is not running on the Physical Sublayer. The state will be off if the interface has been shutdown or if the **cmt disconnect** command has been issued for Physical A or Physical B. |
| Brk | Break State is the entry point in the start of a PCM connection. |
| Tra | Trace State localizes a stuck beacon condition. |
| Con | Connect State is used to synchronize the ends of the connection for the signaling sequence. |
| Nxt | Next State separates the signaling performed in the Signal State and transmits Protocol Data Units (PDUs) while MAC Local Loop is performed. |
| Sig | Signal State is entered from the Next State when a bit is ready to be transmitted. |
| Join | Join State is the first of three states in a unique sequence of transmitted symbol streams received as line states—the Halt Line State, Master Line State, and Idle Line State, or HLS-MLS-ILS—that leads to an active connection. |
| Vfy | Verify State is the second state in the path to the Active State and will not be reached by a connection that is not synchronized. |
| Act | Active State indicates that the CMT process has established communications with its physical neighbor.<br>The transition states are defined in the X3T9.5 specification. You are referred to the specification for details about these states. |

| | |
|---|---|
| ECM is ... | ECM is the SMT entity coordination management, which overlooks the operation of CFM and PCM. The ECM state can be one of the following:<br>• out—The router is isolated from the network.<br>• in—The router is actively connected to the network. This is the normal state for a connected router.<br>• trace—The router is trying to localize a stuck beacon condition.<br>• leave—The router is allowing time for all the connections to break before leaving the network.<br>• path_test—The router is testing its internal paths.<br>• insert—The router is allowing time for the optical bypass to insert.<br>• check—The router is making sure optical bypasses switched correctly.<br>• deinsert—The router is allowing time for the optical bypass to deinsert. |
| CFM is ... | Contains information about the current state of the MAC connection. The Configuration Management (CFM) state can be one of the following:<br>• isolated—The MAC is not attached to any Physical type.<br>• _wrap_a—The MAC is attached to Physical A. Data is received on Physical A and transmitted on Physical A.<br>• wrap b—The MAC is attached to Physical B. Data is received on Physical B and transmitted on Physical B.<br>• wrap_s—The MAC is attached to Physical S. Data is received on Physical S and transmitted on Physical S. This is the normal mode for a single attachment station (SAS).<br>• thru—The MAC is attached to Physical A and B. Data is received on Physical A and transmitted on Physical B. This is the normal mode for a dual attachment station (DAS) with one MAC. The ring has been operational for 1 minute and 42 seconds. |
| RMT is ... | RMT (Ring Management) is the SMT MAC-related state machine. The RMT state can be one of the following:<br>• isolated—The MAC is not trying to participate in the ring. This is the initial state.<br>• non_op—The MAC is participating in ring recovery and ring is not operational.<br>• ring_op—The MAC is participating in an operational ring. This is the normal state while the MAC is connected to the ring.<br>• detect—The ring has been nonoperational for longer than normal. Duplicate address conditions are being checked.<br>• non_op_dup—Indications have been received that the address of the MAC is a duplicate of another MAC on the ring. Ring is not operational.<br>• ring_op_dup—Indications have been received that the address of the MAC is a duplicate of another MAC on the ring. Ring is operational in this state.<br>• directed—The MAC is sending beacon frames notifying the ring of the stuck condition.<br>• trace—Trace has been initiated by this MAC and the RMT state machine is waiting for its completion before starting an internal path test. |

| | |
|---|---|
| token rotation | Token rotation value is the default or configured rotation value as determined by the **fddi token-rotation-time** command. This value is used by all stations on the ring. The default is 5000 microseconds. |
| ring operational | When the ring is operational, the displayed value will be the negotiated token rotation time of all stations on the ring. Operational times are displayed by the number of hours:minutes:seconds the ring has been up. If the ring is not operational, the message “ring not operational” is displayed. |
| Upstream \| downstream neighbor | Displays the canonical MAC address of outgoing upstream and downstream neighbors. If the address is unknown, the value will be the FDDI unknown address (0x00 00 f8 00 00 00). |
| Last input | Number of hours, minutes, and seconds since the last packet was successfully received by an interface. Useful for knowing when a dead interface failed. |
| output | Number of hours, minutes, and seconds since the last packet was successfully transmitted by an interface. |
| output hang | Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the “last” fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed. |
| Last clearing | Time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared.<br>*** indicates the elapsed time is too large to be displayed.<br>0:00:00 indicates the counters were cleared more than $2^{31}$ms (and less than $2^{32}$ms) ago. |
| Output queue, input queue, drops | Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue. |
| Five-minute input rate<br>Five-minute output rate | Average number of bits and packets transmitted per second in the last 5 minutes.<br>The five-minute input and output rates should be used only as an approximation of traffic per second during a given 5-minute period. These rates are exponentially weighted averages with a time constant of 5 minutes. A period of four time constants must pass before the average will be within two percent of the instantaneous rate of a uniform stream of traffic over that period. |
| packets input | Total number of error-free packets received by the system. |
| bytes | Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system. |
| no buffer | Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events. |
| broadcasts | Total number of broadcast or multicast packets received by the interface. |
| runts | Number of packets that are discarded because they are smaller than the medium’s minimum packet size. |

| | |
|---|---|
| giants | Number of packets that are discarded because they exceed the medium's maximum packet size. |
| CRC | Cyclic redundancy checksum generated by the originating LAN station or far-end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRCs is usually the result of collisions or a station transmitting bad data. |
| frame | Number of packets received incorrectly that have a CRC error and a noninteger number of octets. On a LAN, this is usually the result of collisions or a malfunctioning Ethernet device. On an FDDI LAN, this also may be the result of a failing fiber (cracks) or a hardware malfunction. |
| overrun | Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data. |
| ignored | Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be increased. |
| packets output | Total number of messages transmitted by the system. |
| bytes | Total number of bytes, including data and MAC encapsulation, transmitted by the system. |
| underruns | Number of transmit aborts (when the router cannot feed the transmitter fast enough). |
| output errors | Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this may not balance with the sum of the enumerated output errors, because some datagrams may have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories. |
| collisions | Because an FDDI ring cannot have collisions, this statistic is always zero. |
| interface resets | Number of times an interface has been reset. The interface may be reset by the administrator or automatically when an internal error occurs. |
| restarts | Should always be zero for FDDI interfaces. |
| transitions | The number of times the ring made a transition from ring operational to ring nonoperational, or vice versa. A large number of transitions indicates a problem with the ring or the interface. |
| traces | Trace count applies to both the FCI, FCIT, and FIP. Indicates the number of times this interface started a trace. |
| claims | Pertains to FCIT and FIP only. Indicates the number of times this interface has been in claim state. |
| beacons | Pertains to FCIT and FIP only. Indicates the number of times the interface has been in beacon state. |
| Protocol | Protocol that is operating on the interface. |
| Pkts In | Number of packets received for that protocol. |
| Chars In | Number of characters received for that protocol. |

| | |
|---|---|
| Pkts Out | Number of packets transmitted for that protocol. |
| Chars Out | Number of characters transmitted for that protocol. |

# show interfaces hssi

Use the **show interfaces hssi** privileged EXEC command to display information about the HSSI interface.

**show interfaces hssi** *unit* [**accounting**]
**show interfaces hssi** [*slot*/*port*] [**accounting**] (for the Cisco 7000 series)

## Syntax Description

| | |
|---|---|
| *unit* | Must match a port number on the selected interface. |
| **accounting** | (Optional) Displays the number of packets of each protocol type that have been sent through the interface. |
| *slot* | (Optional) On the Cisco 7000 series, slot location of the interface processor. |
| *port* | (Optional) On the Cisco 7000 series, port number on interface. |

## Command Mode

Privileged EXEC

## Sample Display

The following is sample output from the **show interfaces hssi** command when HSSI is enabled:

```
Router# show interfaces hssi 0

HSSI 0 is up, line protocol is up
Hardware is cBus HSSI
Internet address is 150.136.67.190, subnet mask is 255.255.255.0
MTU 4470 bytes, BW 45045 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation HDLC, loopback not set, keepalive set (10 sec)
Last input 0:00:03, output 0:00:00, output hang never
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
Five minute input rate 0 bits/sec, 0 packets/sec
Five minute output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants
              0 parity, 0 rx disabled
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
17 packets output, 994 bytes, 0 underruns
0 output errors, 0 applique, 4 interface resets, 0 restarts
2 carrier transitions
```

Table 6-29 describes significant fields shown in the display.

**Table 6-29 Show Interfaces HSSI Field Descriptions**

| Field | Description |
|---|---|
| HSSI is {up \| down}<br>...is administratively down | Indicates whether the interface hardware is currently active (whether carrier detect is present) and if it has been taken down by an administrator. “Disabled” indicates the router has received over 5000 errors in a keepalive interval, which is 10 seconds by default. |
| line protocol<br>is {up \| down \|<br>administratively down} | Indicates whether the software processes that handle the line protocol considers the line usable (that is, whether keepalives are successful). |
| Hardware | Specifies the hardware type. |
| Internet address | Lists the Internet address followed by subnet mask. |
| MTU | Maximum Transmission Unit of the interface. |
| BW | Bandwidth of the interface in kilobits per second. |
| DLY | Delay of the interface in microseconds. |
| rely | Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over 5 minutes. |
| load | Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over 5 minutes. |
| Encapsulation | Encapsulation method assigned to interface. |
| loopback | Indicates whether loopback is set and type of loopback test. |
| keepalive | Indicates whether keepalives are set or not. |
| Last input | Number of hours, minutes, and seconds since the last packet was successfully received by an interface. Useful for knowing when a dead interface failed. |
| Last output | Number of hours, minutes, and seconds since the last packet was successfully transmitted by an interface. |
| output hang | Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the “last” fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed. |
| Last clearing | Time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared.<br>*** indicates the elapsed time is too large to be displayed.<br>0:00:00 indicates the counters were cleared more than $2^{31}$ms (and less than $2^{32}$ms) ago. |
| Output queue, drops<br>Input queue, drops | Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue. |
| Five minute input rate,<br>Five minute output rate | Average number of bits and packets transmitted per second in the last 5 minutes. |
| packets input | Total number of error-free packets received by the system. |

| Field | Description |
|---|---|
| bytes input | Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system. |
| no buffers | Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events. |
| broadcasts | Total number of broadcast or multicast packets received by the interface. |
| runts | Number of packets that are discarded because they are smaller than the medium’s minimum packet size. |
| giants | Number of packets that are discarded because they exceed the medium’s maximum packet size. |
| parity | Report of the parity errors on the HSSI. |
| rx disabled | Indicates the HSSI could not find a free buffer on the ciscoBus controller to reserve for use for the HSSI receiver. When this happens, the HSSI shuts down its receiver and waits until a buffer is available. Data is not lost unless a packet comes in and overflows the HSSI FIFO. Usually, the receive disables are frequent but do not last for long, and the number of dropped packets is less than the count in the “rx disabled” field. A receive disabled condition can happen in systems that are under heavy traffic load and that have shorter packets. In this situation, the number of buffers available on the ciscoBus controller is at a premium. One way to alleviate this problem is to reduce the mtu on the HSSI interface from 4500 (FDDI size) to 1500 (Ethernet size). Doing so allows the software to take the fixed memory of the ciscoBus controller and divide it into a larger number of smaller buffers, rather than a small number of large buffers. Receive disables are not errors, so they are not included in any error counts. |
| input errors | Sum of all errors that prevented the receipt of datagrams on the interface being examined. This may not balance with the sum of the enumerated output errors, because some datagrams may have more than one error and others may have errors that do not fall into any of the specifically tabulated categories. |
| CRC | Cyclic redundancy checksum generated by the originating LAN station or far-end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRCs is usually the result of collisions or a station transmitting bad data. On a serial link, CRCs usually indicate noise, gain hits, or other transmission problems on the data link. CRC errors are also reported when a far-end abort occurs, and when the idle flag pattern is corrupted. This makes it possible to get CRC errors even when there is no data traffic. |
| frame | Number of packets received incorrectly having a CRC error and a noninteger number of octets. On a serial line, this is usually the result of noise or other transmission problems. |

| Field | Description |
|---|---|
| overrun | Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data. |
| ignored | Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be increased. |
| abort | Number of packets whose receipt was aborted. |
| packets output | Total number of messages transmitted by the system. |
| bytes output | Total number of bytes, including data and MAC encapsulation, transmitted by the system. |
| underruns | Number of times that the far-end transmitter has been running faster than the near-end router's receiver can handle. This may never happen (be reported) on some interfaces. |
| congestion drop | Number of messages discarded because the output queue on an interface grew too long. This can happen on a slow, congested serial link. |
| output errors | Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this may not balance with the sum of the enumerated output errors, as some datagrams may have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories. |
| applique | Indicates an unrecoverable error has occurred on the HSA applique. The system then invokes an interface reset. |
| interface resets | Number of times an interface has been completely reset. This can happen if packets queued for transmission were not sent within several seconds time. On a serial line, this can be caused by a malfunctioning modem that is not supplying the transmit clock signal, or by a cable problem. If the system notices that the carrier detect line of a serial interface is up, but the line protocol is down, it periodically resets the interface in an effort to restart it. Interface resets can also occur when an interface is looped back or shut down. |
| restarts | Number of times the controller was restarted because of errors. |
| carrier transitions | Number of times the carrier detect signal of a serial interface has changed state. Indicates modem or line problems if the carrier detect line is changing state often. |
| Protocol | Protocol that is operating on the interface. |
| Pkts In | Number of packets received for that protocol. |
| Chars In | Number of characters received for that protocol. |
| Pkts Out | Number of packets transmitted for that protocol. |
| Chars Out | Number of characters transmitted for that protocol. |

The following is an example of the **show interfaces hssi** command on a Cisco 7000:

```
Router# show in hssi 1/0
```

```
Hssi1/0 is up, line protocol is up
  Hardware is cxBus HSSI
  Internet address is 131.108.38.14, subnet mask is 255.255.255.0
  MTU 1500 bytes, BW 45045 Kbit, DLY 1000000 usec, rely 255/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input 0:00:00, output 0:00:08, output hang never
  Last clearing of "show interface" counters never
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 1000 bits/sec, 2 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
     630573548 packets input, 2077237628 bytes, 0 no buffer
     Received 2832063 broadcasts, 0 runts, 0 giants
              0 parity, 1970 rx disabled
     113 input errors, 20 CRC, 93 frame, 0 overrun, 0 ignored, 0 abort
     629721628 packets output, 1934313295 bytes, 0 underruns
     0 output errors, 0 applique, 62 interface resets, 0 restarts
     309 carrier transitions
```

The following is an example of the **show interfaces hssi** command with the **accounting** option on a Cisco 7000:

```
Router# show interfaces hssi 1/0 accounting

HIP1/0
       Protocol    Pkts In   Chars In   Pkts Out  Chars Out
             IP       7344    4787842       1803    1535774
      Appletalk      33345    4797459      12781    1089695
        DEC MOP          0          0        127       9779
            ARP          7        420         39       2340
```

# show interfaces lex

To display statistics about a LAN Extender interface, use the **show interface lex** EXEC command.

**show interfaces lex** *number* [**ethernet** | **serial**]

## Syntax Description

| | |
|---|---|
| *number* | Number of the LAN Extender interface that resides on the core router about which to display statistics. |
| **ethernet** | (Optional) Displays statistics about the Ethernet interface that resides on the LAN Extender chassis. |
| serial | (Optional) Displays statistics about the serial interface that resides on the LAN Extender chassis. |

## Command Mode

EXEC

## Usage Guidelines

To display statistics about the LAN Extender interface on the core router, use the **show interfaces lex** command without any keywords.

Administratively, the physical serial interface that connects the core router to the LAN Extender is completely hidden. The **show interfaces serial** command will show only that the serial interface is present. However, it will not report any statistics about the traffic passing over the physical line. All statistics are report by the **show interfaces lex** command.

## Sample Displays

The following is sample output from the **show interfaces lex** command, showing the LAN Extender interface on the host router. Note the “Bound to ...” field, which is displayed only on a LAN Extender interface.

```
Router# show interfaces lex 0

Lex0 is up, line protocol is up
  Hardware is Lan Extender, address is 0204.0301.1526 (bia 0000.0000.0000)
  MTU 1500 bytes, BW 10000 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 4:00:00
  Bound to Serial3
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 1000 bits/sec, 0 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
     1022 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     2070 packets output, 23663 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets, 0 restarts
```

The following is sample output from the **show interfaces lex** command when you specify the **ethernet** keyword:

```
Router# show interfaces lex 0 ethernet

Lex0-Ethernet0 is up, line protocol is up
  Hardware is LAN-Extender, address is 0000.0c01.1526 (bia 0000.0c01.1526)
  Last input 6w3d, output 6w3d
  Last clearing of "show interface" counters 0:02:30
  Output queue 40/50, 60 drops; input queue 10/40, 2 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
     3916 packets input, 960303 bytes, 3 no buffer
     Received 2 broadcasts, 3 runts, 3 giants
     2 input errors, 1 CRC, 1 frame, 1 overrun, 3 ignored, 2 abort
     2500 packets output, 128288 bytes, 1 underruns
     1 output errors, 1 collisions, 0 interface resets, 0 restarts
```

The following is sample output from the **show interfaces lex** command when you specify the **serial** keyword:

```
Router# show interfaces lex 0 serial

Lex0-Serial0 is up, line protocol is up
  Hardware is LAN-Extender
  Last input 6w3d, output 6w3d
  Last clearing of "show interface" counters 0:03:05
  Input queue: 5/15/4 (size/max/drops); Total output drops: 450
  Output queue: high 25/35/90, medium 70/80/180, normal 40/50/120, low 10/20/60
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
     1939 packets input, 30998 bytes, 6 no buffer
     Received 4 broadcasts, 6 runts, 6 giants
     4 input errors, 2 CRC, 2 frame, 2 overrun, 6 ignored, 4 abort
     1939 packets output, 219535 bytes, 2 underruns
     2 output errors, 2 collisions, 0 interface resets, 0 restarts
     2 carrier transitions
```

Table 6-30 describes the fields shown in these displays.

**Table 6-30 Show Interfaces Lex Field Descriptions**

| Field | Description |
|---|---|
| Lex0 is up, line protocol is up | Indicates whether the logical LAN Extender interface on the core router is currently active (that is, whether carrier detect is present) and whether it has been taken down by an administrator. |
| Lex0-Ethernet0 is up, line protocol is up<br>Lex0-Serial0 is up, line protocol is up | Indicates whether the physical Ethernet and serial interfaces on the LAN Extender chassis are currently active (that is, whether carrier detect is present) and if it has been taken down by an administrator. |
| Hardware is LAN-Extender | Hardware type of the interfaces on the LAN Extender. |
| address is... | Logical MAC address of the interface. |
| bia | Burned-in MAC address of the interface. The LAN Extender interface does not have a burned in address; hence it appears as all zeroes. |
| MTU | Maximum transmission unit size of the interface. |

| Field | Description |
|---|---|
| BW | Value of the bandwidth parameter that has been configured for the interface (in kilobits per second). The bandwidth parameter is used to compute IGRP metrics only. If the interface is attached to a serial line with a line speed that does not match the default (1536 or 1544 for T1 and 56 for a standard synchronous serial line), use the **bandwidth** command to specify the correct line speed for this serial line. |
| DLY | Delay of the interface in microseconds. |
| rely | Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over 5 minutes. |
| load | Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over 5 minutes. |
| Encapsulation | Encapsulation method assigned to interface. |
| ARP type | Type of Address Resolution Protocol assigned. |
| ARP Timeout | Number of hours, minutes, and seconds an ARP cache entry will stay in the cache. |
| Bound to ... | Number of the serial interface to which the logical LAN Extender interface is bound. |
| Last input | Number of hours, minutes, and seconds (or never) since the last packet was successfully received by an interface. This is useful for knowing when a dead interface failed. |
| Last output | Number of hours, minutes, and seconds (or never) since the last packet was successfully transmitted by an interface. |
| output hang | Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the “last” fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed. |
| Last clearing of “show interface” counters | Time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared.<br>*** indicates the elapsed time is too large to be displayed.<br>0:00:00 indicates the counters were cleared more than 231ms (and less than 232ms) ago |
| Output queue, drops<br>input queue, drops | Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue. |
| Five minute input rate<br>Five minute output rate | Average number of bits and packets transmitted per second in the last 5 minutes.<br>The 5-minute input and output rates should be used only as an approximation of traffic per second during a given 5-minute period. These rates are exponentially weighted averages with a time constant of 5 minutes. A period of four time constants must pass before the average will be within two percent of the instantaneous rate of a uniform stream of traffic over that period. |
| packets input | Total number of error-free packets received by the system. |
| bytes | Total number of bytes, including data and MAC encapsulation, in the error-free packets received by the system. |

| Field | Description |
|---|---|
| no buffer | Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events. |
| Received ... broadcasts | Total number of broadcast or multicast packets received by the interface. |
| runts | Number of packets that are discarded because they are smaller than the medium's minimum packet size. |
| giants | Number of packets that are discarded because they exceed the medium's maximum packet size. |
| input errors | Total number of no buffer, runts, giants, CRCs, frame, overrun, ignored, and abort counts. Other input-related errors can also increment the count, so that this sum might not balance with the other counts. |
| CRC | Cyclic redundancy checksum generated by the originating station or far-end device does not match the checksum calculated from the data received. On a serial link, CRCs usually indicate noise, gain hits, or other transmission problems on the data link. |
| frame | Number of packets received incorrectly having a CRC error and a noninteger number of octets. On a serial line, this is usually the result of noise or other transmission problems. |
| overrun | Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data. |
| ignored | Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. Broadcast storms and bursts of noise can cause the ignored count to be increased. |
| abort | Illegal sequence of one bits on a serial interface. This usually indicates a clocking problem between the serial interface and the data link equipment. |
| input packets with dribble condition detected | Does not apply to a LAN Extender interface. |
| packets output | Total number of messages transmitted by the system. |
| bytes | Total number of bytes, including data and MAC encapsulation, transmitted by the system. |
| underruns | Number of times that the transmitter has been running faster than the router can handle. This might never be reported on some interfaces. |
| output errors | Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this might not balance with the sum of the enumerated output errors, as some datagrams may have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories. |
| collisions | Number of messages retransmitted due to an Ethernet collision. This usually is the result of an overextended LAN (Ethernet or transceiver cable too long, more than two repeaters between stations, or too many cascaded multiport transceivers). Some collisions are normal. However, if your collision rate climbs to around 4 or 5%, you should consider verifying that there is no faulty equipment on the segment and/or moving some existing stations to a new segment. A packet that collides is counted only once in output packets. |

| Field | Description |
| --- | --- |
| interface resets | Number of times an interface has been completely reset. This can happen if packets queued for transmission were not sent within several seconds' time. On a serial line, this can be caused by a malfunctioning modem that is not supplying the transmit clock signal, or by a cable problem. If the system notices that the carrier detect line of a serial interface is up, but the line protocol is down, it periodically resets the interface in an effort to restart it. Interface resets can also occur when an interface is looped back or shut down. |
| restarts | Number of times the controller was restarted because of errors. |

# show interfaces loopback

Use the **show interfaces loopback** privileged EXEC command to display information about the loopback interface.

**show interfaces loopback** [*unit*] [**accounting**]

## Syntax Description

| | |
|---|---|
| *unit* | (Optional) Must match a port number on the selected interface. |
| **accounting** | (Optional) Displays the number of packets of each protocol type that have been sent through the interface. |

## Command Mode

Privileged EXEC

## Sample Displays

The following is sample output from the **show interfaces loopback** command:

```
Router# show interfaces loopback 0

Loopback0 is up, line protocol is up
  Hardware is Loopback
  MTU 1500 bytes, BW 1 Kbit, DLY 50 usec, rely 255/255, load 1/255
  Encapsulation UNKNOWN, loopback not set, keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Output queue 0/0, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets, 0 restarts
```

The following is sample output when the **accounting** keyword is included:

```
Router# show interfaces loopback 0 accounting

Loopback0
                Protocol    Pkts In   Chars In   Pkts Out  Chars Out
No traffic sent or received on this interface.
```

Table 6-31 describes significant fields shown in the displays.

**Table 6-31 Show Interfaces Loopback Descriptions**

| Field | Description |
|---|---|
| Loopback is {up \| down}<br>...is administratively down | Indicates whether the interface hardware is currently active (whether carrier detect is present) and if it has been taken down by an administrator. “Disabled” indicates the router has received over 5000 errors in a keepalive interval, which is 10 seconds by default. |

| Field | Description |
|---|---|
| line protocol is {up \| down \| administratively down} | Indicates whether the software processes that handle the line protocol considers the line usable (that is, whether keepalives are successful). |
| Hardware | Hardware is Loopback. |
| MTU | Maximum Transmission Unit of the interface. |
| BW | Bandwidth of the interface in kilobits per second. |
| DLY | Delay of the interface in microseconds. |
| rely | Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over 5 minutes. |
| load | Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over 5 minutes. |
| Encapsulation | Encapsulation method assigned to interface. |
| loopback | Indicates whether loopback is set and type of loopback test. |
| keepalive | Indicates whether keepalives are set or not. |
| Last input | Number of hours, minutes, and seconds since the last packet was successfully received by an interface. Useful for knowing when a dead interface failed. |
| Last output | Number of hours, minutes, and seconds since the last packet was successfully transmitted by an interface. |
| output hang | Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the “last” fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed. |
| Last clearing | Time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared.<br>*** indicates the elapsed time is too large to be displayed.<br>0:00:00 indicates the counters were cleared more than $2^{31}$ms (and less than $2^{32}$ms) ago. |
| Output queue, drops<br>Input queue, drops | Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue. |
| Five minute input rate,<br>Five minute output rate | Average number of bits and packets transmitted per second in the last 5 minutes. |
| packets input | Total number of error-free packets received by the system. |
| bytes input | Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system. |
| no buffer | Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events. |

| Field | Description |
|---|---|
| broadcasts | Total number of broadcast or multicast packets received by the interface. |
| runts | Number of packets that are discarded because they are smaller than the medium's minimum packet size. |
| giants | Number of packets that are discarded because they exceed the medium's maximum packet size. |
| input errors | Sum of all errors that prevented the receipt of datagrams on the interface being examined. This may not balance with the sum of the enumerated output errors, because some datagrams may have more than one error and others may have errors that do not fall into any of the specifically tabulated categories. |
| CRC | Cyclic redundancy checksum generated by the originating LAN station or far-end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRCs is usually the result of collisions or a station transmitting bad data. On a serial link, CRCs usually indicate noise, gain hits, or other transmission problems on the data link. CRC errors are also reported when a far-end abort occurs, and when the idle flag pattern is corrupted. This makes it possible to get CRC errors even when there is no data traffic. |
| frame | Number of packets received incorrectly having a CRC error and a noninteger number of octets. On a serial line, this is usually the result of noise or other transmission problems. |
| overrun | Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data. |
| ignored | Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be increased. |
| abort | Number of packets whose receipt was aborted. |
| packets output | Total number of messages transmitted by the system. |
| bytes output | Total number of bytes, including data and MAC encapsulation, transmitted by the system. |
| underruns | Number of times that the far-end transmitter has been running faster than the near-end router's receiver can handle. This may never happen (be reported) on some interfaces. |
| output errors | Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this may not balance with the sum of the enumerated output errors, as some datagrams may have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories. |
| collisions | A loopback interface does not have collisions. |

| Field | Description |
|---|---|
| interface resets | Number of times an interface has been completely reset. This can happen if packets queued for transmission were not sent within several seconds time. On a serial line, this can be caused by a malfunctioning modem that is not supplying the transmit clock signal, or by a cable problem. If the system notices that the carrier detect line of a serial interface is up, but the line protocol is down, it periodically resets the interface in an effort to restart it. Interface resets can also occur when an interface is looped back or shut down. |
| restarts | Number of times the controller was restarted because of errors. |
| Protocol | Protocol that is operating on the interface. |
| Pkts In | Number of packets received for that protocol. |
| Chars In | Number of characters received for that protocol. |
| Pkts Out | Number of packets transmitted for that protocol. |
| Chars Out | Number of characters transmitted for that protocol. |

# show interfaces serial

Use the **show interfaces serial** privileged EXEC command to display information about a serial interface.

**show interfaces serial** [*number*] [**accounting**]
**show interfaces serial** [*slot*/*port*] [**accounting**] (for the Cisco 7000 series)

## Syntax Description

| | |
|---|---|
| *number* | (Optional) Must match the interface port number. |
| **accounting** | (Optional) Displays the number of packets of each protocol type that have been sent through the interface. |
| *slot* | (Optional) On the Cisco 7000 series, slot location of the interface processor. |
| *port* | (Optional) On the Cisco 7000 series, port number on interface. |

## Command Mode

Privileged EXEC

## Sample Display

The following is sample output from the **show interfaces** command for a synchronous serial interface:

```
Router# show interfaces serial

Serial 0 is up, line protocol is up
   Hardware is MCI Serial
   Internet address is 150.136.190.203, subnet mask is 255.255.255.0
   MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
   Encapsulation HDLC, loopback not set, keepalive set (10 sec)
   Last input 0:00:07, output 0:00:00, output hang never
   Output queue 0/40, 0 drops; input queue 0/75, 0 drops
   Five minute input rate 0 bits/sec, 0 packets/sec
   Five minute output rate 0 bits/sec, 0 packets/sec
       16263 packets input, 1347238 bytes, 0 no buffer
       Received 13983 broadcasts, 0 runts, 0 giants
       2 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 2 abort
1 carrier transitions

     22146 packets output, 2383680 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets, 0 restarts
```

Table 6-32 describes significant fields shown in the display.

**Table 6-32 Show Interfaces Serial Field Descriptions**

| Field | Description |
|---|---|
| Serial ... is {up \| down}<br>...is administratively down | Indicates whether the interface hardware is currently active (whether carrier detect is present) and if it has been taken down by an administrator. "Disabled" indicates the router has received over 5000 errors in a keepalive interval, which is 10 seconds by default. |
| line protocol<br>is {up \| down} | Indicates whether the software processes that handle the line protocol consider the line usable (that is, whether keepalives are successful) or if it has been taken down by an administrator. |
| Hardware is | Specifies the hardware type. |
| Internet address is | Specifies the Internet address and subnet mask. |
| MTU | Maximum Transmission Unit of the interface. |
| BW 1544 Kbit | Indicates the value of the bandwidth parameter that has been configured for the interface (in kilobits per second). The bandwidth parameter is used to compute IGRP metrics only. If the interface is attached to a serial line with a line speed that does not match the default (1536 or 1544 for T1 and 56 for a standard synchronous serial line), use the **bandwidth** command to specify the correct line speed for this serial line. |
| DLY | Delay of the interface in microseconds. |
| rely | Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over 5 minutes. |
| load | Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over 5 minutes. |
| Encapsulation | Encapsulation method assigned to interface. |
| loopback | Indicates whether loopback is set or not. |
| keepalive | Indicates whether keepalives are set or not. |
| Last input | Number of hours, minutes, and seconds since the last packet was successfully received by an interface. Useful for knowing when a dead interface failed. |
| Last output | Number of hours, minutes, and seconds since the last packet was successfully transmitted by an interface. |
| output hang | Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the "last" fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed. |
| Output queue, drops<br>input queue, drops | Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue. |
| Five minute input rate<br>Five minute output rate | Average number of bits and packets transmitted per second in the last 5 minutes.<br>The 5-minute input and output rates should be used only as an approximation of traffic per second during a given 5-minute period. These rates are exponentially weighted averages with a time constant of 5 minutes. A period of four time constants must pass before the average will be within two percent of the instantaneous rate of a uniform stream of traffic over that period. |
| packets input | Total number of error-free packets received by the system. |

| Field | Description |
|---|---|
| bytes input | Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system. |
| no buffers | Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events. |
| Received ... broadcasts | Total number of broadcast or multicast packets received by the interface. |
| runts | Number of packets that are discarded because they are smaller than the medium's minimum packet size. |
| giants | Number of packets that are discarded because they exceed the medium's maximum packet size. |
| input error | Total number of no buffer, runts, giants, CRCs, frame, overrun, ignored, and abort counts. Other input-related errors can also increment the count, so that this sum may not balance with the other counts. |
| CRC | Cyclic redundancy checksum generated by the originating station or far-end device does not match the checksum calculated from the data received. On a serial link, CRCs usually indicate noise, gain hits, or other transmission problems on the data link. |
| frame | Number of packets received incorrectly having a CRC error and a noninteger number of octets. On a serial line, this is usually the result of noise or other transmission problems. |
| overrun | Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data. |
| ignored | Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. Broadcast storms and bursts of noise can cause the ignored count to be increased. |
| abort | Illegal sequence of one bits on a serial interface. This usually indicates a clocking problem between the serial interface and the data link equipment. |
| packets output | Total number of messages transmitted by the system. |
| bytes output | Total number of bytes, including data and MAC encapsulation, transmitted by the system. |
| underruns | Number of times that the transmitter has been running faster than the router can handle. This may never be reported on some interfaces. |
| output errors | Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this may not balance with the sum of the enumerated output errors, as some datagrams may have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories. |
| collisions | Number of messages retransmitted due to an Ethernet collision. This usually is the result of an overextended LAN (Ethernet or transceiver cable too long, more than two repeaters between stations, or too many cascaded multiport transceivers). Some collisions are normal. However, if your collision rate climbs to around 4 or 5%, you should consider verifying that there is no faulty equipment on the segment and/or moving some existing stations to a new segment. A packet that collides is counted only once in output packets. |

| Field | Description |
|---|---|
| interface resets | Number of times an interface has been completely reset. This can happen if packets queued for transmission were not sent within several seconds' time. On a serial line, this can be caused by a malfunctioning modem that is not supplying the transmit clock signal, or by a cable problem. If the system notices that the carrier detect line of a serial interface is up, but the line protocol is down, it periodically resets the interface in an effort to restart it. Interface resets can also occur when an interface is looped back or shut down. |
| restarts | Number of times the controller was restarted because of errors. |
| carrier transitions | Number of times the carrier detect signal of a serial interface has changed state. For example, if data carrier detect (DCD) goes down and comes up, the carrier transition counter will increment two times. Indicates modem or line problems if the carrier detect line is changing state often. |
| alarm indications, remote alarms, rx LOF, rx LOS | Number of CSU/DSU alarms, and number of occurrences of receive loss of frame and receive loss of signal. |
| BER inactive, NELR inactive, FELR inactive | Status of G.703-E1 counters for bit error rate (BER) alarm, near-end loop remote (NELR), and far-end loop remote (FELR). Note that you cannot set the NELR or FELR. |

The following is sample output of the **show interfaces serial** command for the HDLC synchronous serial interface on a Cisco 7000:

```
Router# show interfaces serial 1/0

Serial1/0 is up, line protocol is up
  Hardware is cxBus Serial
  Internet address is 150.136.190.203, subnet mask is 255.255.255.0
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input 0:00:07, output 0:00:00, output hang never
  Last clearing of “show interface” counters 2w4d
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
     16263 packets input, 1347238 bytes, 0 no buffer
     Received 13983 broadcasts, 0 runts, 0 giants
     2 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 2 abort
     22146 packets output, 2383680 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets, 0 restarts
     1 carrier transitions
```

The following is sample output of the **show interfaces serial** command for a G.703 interface on which framing is enabled:

```
Router# show interfaces serial 2/3

Serial2/3 is up, line protocol is up
  Hardware is cxBus Serial
  Internet address is 5.4.4.1, subnet mask is 255.255.255.0
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive not set
  Last input 0:00:21, output 0:00:21, output hang never
  Last clearing of “show interface” counters never
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
     53 packets input, 7810 bytes, 0 no buffer
```

```
Received 53 broadcasts, 0 runts, 0 giants
2 input errors, 2 CRC, 0 frame, 0 overrun, 0 ignored, 2 abort
56 packets output, 8218 bytes, 0 underruns
0 output errors, 0 collisions, 2 interface resets, 0 restarts
1 carrier transitions
2 alarm indications, 333 remote alarms, 332 rx LOF, 0 rx LOS
RTS up, CTS up, DTR up, DCD up, DSR up
BER inactive, NELR inactive, FELR inactive
```

Table 6-32 describes significant fields shown in the display.

## Sample Display with Frame Relay Encapsulation

When using the Frame Relay encapsulation, use the **show interfaces** command to display information on the multicast DLCI, the DLCI of the interface, and the LMI DLCI used for the local management interface.

The multicast DLCI and the local DLCI can be set using the **frame-relay multicast-dlci** and the **frame-relay local-dlci** configuration commands, or provided through the local management interface. The status information is taken from the LMI, when active.

The following is sample output from the **show interfaces serial** command when using Frame Relay encapsulation:

```
Router# show interfaces serial

Serial 2 is up, line protocol is up
   Hardware type is MCI Serial
   Internet address is 131.108.122.1, subnet mask is 255.255.255.0
   MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
   Encapsulation FRAME-RELAY, loopback not set, keepalive set (10 sec)
   multicast DLCI 1022,  status defined, active
   source DLCI    20, status defined, active
   LMI DLCI 1023, LMI sent 10, LMI stat recvd 10, LMI upd recvd 2
   Last input 7:21:29, output 0:00:37, output hang never
   Output queue 0/100, 0 drops; input queue 0/75, 0 drops
   Five minute input rate 0 bits/sec, 0 packets/sec
   Five minute output rate 0 bits/sec, 0 packets/sec
       47 packets input, 2656 bytes, 0 no buffer
       Received 5 broadcasts, 0 runts, 0 giants
       5 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 57 abort
       518 packets output, 391205 bytes
       0 output errors, 0 collisions, 0 interface resets, 0 restarts
       1 carrier transitions
```

In this display, the multicast DLCI has been changed to 1022 with the **frame-relay multicast-dlci** interface configuration command.

The display shows the statistics for the LMI are the number of status inquiry messages sent (LMI sent), the number of status messages received (LMI recvd), and the number of status updates received (upd recvd). See the *Frame Relay Interface* specification for additional explanations of this output.

## Sample Display with ANSI LMI

For a serial interface with the ANSI LMI enabled, use the **show interfaces** command to determine the LMI type implemented.

The following is a sample display from the **show interfaces** output for a serial interface with the ANSI LMI enabled:

```
Router# show interfaces serial

Serial 1 is up, line protocol is up
   Hardware is MCI Serial
   Internet address is 131.108.121.1, subnet mask is 255.255.255.0
   MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
   Encapsulation FRAME-RELAY, loopback not set, keepalive set
   LMI DLCI    0, LMI sent 10, LMI stat recvd 10
   LMI type is ANSI Annex D
   Last input 0:00:00, output 0:00:00, output hang never
   Output queue 0/40, 0 drops; input queue 0/75, 0 drops
   Five minute input rate 0 bits/sec, 1 packets/sec
   Five minute output rate 1000 bits/sec, 1 packets/sec
       261 packets input, 13212 bytes, 0 no buffer
       Received 33 broadcasts, 0 runts, 0 giants
       0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
       238 packets output, 14751 bytes, 0 underruns
       0 output errors, 0 collisions, 0 interface resets, 0 restarts
```

Notice that the **show interfaces** output for a serial interface with ANSI LMI shown in this display is very similar to that for encapsulation set to Frame Relay, as shown in the previous display. Table 6-33 describes the few differences that exist.

**Table 6-33 Show Interfaces Serial Field Description with ANSI LMI**

| Field | Description |
|---|---|
| LMI DLCI 0 | Identifies the DLCI used by the LMI for this interface. Default: 1023. |
| LMI sent 10 | Number of LMI packets the router sent. |
| LMI type is ANSI Annex D | Indicates that the interface is configured for the ANSI-adopted Frame Relay specification T1.617 Annex D. |

## Sample Display with LAPB Encapsulation

Use the **show interfaces** command to display operation statistics for an interface using LAPB encapsulation.

The following is sample output from the **show interfaces** command for a serial interface using LAPB encapsulation:

```
Router# show interfaces

LAPB state is DISCONNECT, T1 3000, N1 12000, N2 20, K7, TH 3000
Window is closed
IFRAMEs 12/28 RNRs 0/1 REJs 13/1 SABMs 1/13 FRMRs 3/0 DISCs 0/11
```

Table 6-34 shows the fields relevant to all LAPB connections.

**Table 6-34 Show Interfaces Serial Field Descriptions when LAPB Is Enabled**

| Parameter | Description |
|---|---|
| LAPB state is DISCONNECT | State of the LAPB protocol. |
| T1 3000, N1 12000, ... | Current parameter settings. |
| Window is closed | Indicates that no more frames can be transmitted until some outstanding frames have been acknowledged. |

| Parameter | Description |
|---|---|
| IFRAMEs 12/28 RNRs 0/1 ... | Count of the different types of frames in the form of sent/received. |

## Show Interfaces Serial with PPP

An interface configured for synchronous PPP encapsulation differs from the standard **show interface serial** output. An interface configured for PPP might include the following information.

```
lcp state = OPEN
ncp ipcp state = OPEN   ncp osicp state = NOT NEGOTIATED
ncp ipxcp state = NOT NEGOTIATED   ncp xnscp state = NOT NEGOTIATED
ncp vinescp state = NOT NEGOTIATED   ncp deccp state = NOT NEGOTIATED
ncp bridgecp state = NOT NEGOTIATED   ncp atalkcp state = NOT NEGOTIATED
```

Table 6-35 show the fields relevant to PPP connections.

**Table 6-35 Show Interfaces Serial Field Descriptions with PPP Encapsulation**

| Field | Description |
|---|---|
| lcp state | Link Control Protocol |
| ncp ipcp state | Network Control Protocol Internet Protocol Control Protocol |
| ncp osicp state | Network Control Protocol OSI (CLNS) Control Protocol |
| ncp ipxcp state | Network Control Protocol IPX (Novell) Control Protocol |
| ncp xnscp state | Network Control Protocol XNS Control Protocol |
| ncp vinescp state | Network Control Protocol VINES Control Protocol |
| ncp deccp state | Network Control Protocol DECnet Control Protocol |
| ncp bridgecp state | Network Control Protocol Bridging Control Protocol |
| ncp atalkcp state | Network Control Protocol AppleTalk Control Protocol |

## Sample Display with SDLC Connections

Use the **show interfaces** command to display the SDLC information for a given SDLC interface. The following is sample output from the **show interfaces** command for an SDLC primary interface supporting the SDLLC function.

```
Router# show interfaces

Serial 0 is up, line protocol is up
Hardware is MCI Serial
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation SDLC-PRIMARY, loopback not set
    Timers (msec): poll pause 100 fair poll 500. Poll limit 1
    [T1 3000, N1 12016, N2 20, K 7] timer: 56608 Last polled device: none
    SDLLC [ma: 0000.0C01.14--, ring: 7 bridge: 1, target ring: 10
             largest token ring frame 2052]
SDLC addr C1 state is CONNECT
     VS 6, VR 3, RCNT 0, Remote VR 6, Current retransmit count 0
     Hold queue: 0/12 IFRAMEs 77/22 RNRs 0/0 SNRMs 1/0 DISCs 0/0
     Poll: clear, Poll count: 0, chain: p: C1 n: C1
     SDLLC [largest SDLC frame: 265, XID: disabled]
 Last input 00:00:02, output 00:00:01, output hang never
 Output queue 0/40, 0 drops; input queue 0/75, 0 drops
```

```
Five minute input rate 517 bits/sec, 30 packets/sec
Five minute output rate 672 bits/sec, 20 packets/sec
    357 packets input, 28382 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    926 packets output, 77274 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets, 0 restarts
    2 carrier transitions
```

Table 6-36 shows the fields relevant to all SDLC connections.

**Table 6-36 Show Interfaces Serial Field Descriptions when SDLC Is Enabled**

| Parameter | Description |
|---|---|
| Timers (msec): poll pause, fair poll, Poll limit | Current values of these timers, as described in the configuration section, for this interface. |
| T1, N1, N2, K | Values for these parameters, as described in the configuration section, for this interface. |

Table 6-37 shows other data given for each SDLC secondary configured to be attached to this interface.

**Table 6-37 SDLC Secondary Descriptions**

| SDLC Secondary | Description |
|---|---|
| addr | Address of this secondary. |
| state is | Current state of this connection, which is one of the following: |
| DISCONNECT | No communication is being attempted to this secondary. |
| CONNECT | A normal connect state exists between this router and this secondary. |
| DISCSENT | This router has sent a disconnect request to this secondary and is awaiting its response. |
| SNRMSENT | This router has sent a connect request (SNRM) to this secondary and is awaiting its response. |
| THEMBUSY | This secondary has told this router that it is temporarily unable to receive any more information frames. |
| USBUSY | This router has told this secondary that it is temporarily unable to receive any more information frames. |
| BOTHBUSY | Both sides have told each other that they are temporarily unable to receive any more information frames. |
| ERROR | This router has detected an error and is waiting for a response from the secondary acknowledging this. |
| VS | Sequence number of the next information frame this station sends. |
| VR | Sequence number of the next information frame from this secondary that this station expects to receive. |
| Remote VR | Last frame transmitted by this station that has been acknowledged by the other station. |
| Current retransmit count: | Number of times the current I-frame or sequence of I-frames has been retransmitted. |
| Hold Queue | Number of frames in hold queue/Maximum size of hold queue. |

| SDLC Secondary | Description |
|---|---|
| IFRAMEs, RNRs, SNRMs, DISCs | Sent/received count for these frames. |
| Poll | "Set" if this router has a poll outstanding to the secondary; "clear" if it does not. |
| Poll Count | Number of polls in a row that have been given to this secondary at this time. |
| Chain | Shows the previous (p) and next (n) secondary address on this interface in the *round robin loop* of polled devices. |

## Sample Display with SDLLC

Use the **show interfaces serial** command to display the SDLLC statistics for SDLLC configured interfaces.

The following is sample output from the **show interfaces serial** command for an a serial interface configured for SDLLC:

```
Router# show interfaces serial

Serial 0 is up, line protocol is up
   Hardware is MCI Serial
   MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
   Encapsulation SDLC-PRIMARY, loopback not set
       Timers (msec): poll pause 100 fair poll 500. Poll limit 1
       [T1 3000, N1 12016, N2 20, K 7] timer: 56608 Last polled device: none
       SDLLC [ma: 0000.0C01.14--, ring: 7 bridge: 1, target ring: 10
             largest token ring frame 2052]
   SDLC addr C1 state is CONNECT
       VS 6, VR 3, RCNT 0, Remote VR 6, Current retransmit count 0
       Hold queue: 0/12 IFRAMEs 77/22 RNRs 0/0 SNRMs 1/0 DISCs 0/0
       Poll: clear, Poll count: 0, chain: p: C1 n: C1
       SDLLC [largest SDLC frame: 265, XID: disabled]
   Last input 00:00:02, output 00:00:01, output hang never
   Output queue 0/40, 0 drops; input queue 0/75, 0 drops
   Five minute input rate 517 bits/sec, 30 packets/sec
   Five minute output rate 672 bits/sec, 20 packets/sec
       357 packets input, 28382 bytes, 0 no buffer
       Received 0 broadcasts, 0 runts, 0 giants
       0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
       926 packets output, 77274 bytes, 0 underruns
       0 output errors, 0 collisions, 0 interface resets, 0 restarts
       6608 Last polled device: none
       SDLLC [ma: 0000.0C01.14--, ring: 7 brid2 carrier transitions
```

Most of the output shown in the display is generic to all SDLC encapsulated interfaces and is described in the "LLC2 and SDLC Commands" chapter. Table 6-38 shows the parameters specific to SDLLC.

**Table 6-38 SDLLC Parameters**

| Parameter | Description |
|---|---|
| SDLLC ma | Lists the MAC address configured for this interface. The last byte is shown as "--" to indicate that it is filled in with the SDLC address of the connection. |
| ring, bridge, target ring | Lists the parameters as configured by the **sdllc traddr** command. |

| Parameter | Description |
|---|---|
| largest token ring frame | Shows the largest Token Ring frame that is accepted on the LLC2 side of the connection. |
| largest SDLC frame | Shows the largest SDLC frame that is accepted and will be generated on the SDLC side of the connection. |
| XID | Enabled or disabled: Shows whether XID processing is enabled on the SDLC side of the connection. If enabled, it will show the XID value for this address. |

## Sample Display with Accounting Option

The following example illustrates the **show interfaces serial** command with the **accounting** option on a Cisco 7000:

```
Router# show interfaces serial 1/0 accounting

Serial1/0
       Protocol    Pkts In   Chars In   Pkts Out  Chars Out
             IP       7344    4787842       1803    1535774
      Appletalk      33345    4797459      12781    1089695
        DEC MOP          0          0        127       9779
            ARP          7        420         39       2340
```

# show interfaces tokenring

Use the **show interfaces tokenring** privileged EXEC command to display information about the Token Ring interface and the state of source route bridging.

**show interfaces tokenring** *unit* [**accounting**]
**show interfaces tokenring** *slot*/*port* [**accounting**] (for the Cisco 7000 series)

## Syntax Description

| | |
|---|---|
| *unit* | Must match the interface port line number. |
| **accounting** | (Optional) Displays the number of packets of each protocol type that have been sent through the interface. |
| *slot* | On the Cisco 7000 series, optional slot location of the interface processor. On the 7000, value can be 0, 1, 2, 3, or 4. On the 7010, value can be 0, 1, or 2. |
| *port* | On the Cisco 7000 series, optional port number on interface. Value can be 0, 1, 2, or 3. |

## Command Mode

Privileged EXEC

## Usage Guidelines

If you do not provide values for the parameters *slot* and *port*, the command will display statistics for all the network interfaces. The optional keyword **accounting** displays the number of packets of each protocol type that have been sent through the interface.

## Sample Display

The following is sample output from the **show interfaces tokenring** command:

```
Router# show interfaces tokenring

TokenRing 0 is up, line protocol is up
Hardware is 16/4 Token Ring, address is 5500.2000.dc27 (bia 0000.3000.072b)
   Internet address is 150.136.230.203, subnet mask is 255.255.255.0
   MTU 8136 bytes, BW 16000 Kbit, DLY 630 usec, rely 255/255, load 1/255
   Encapsulation SNAP, loopback not set, keepalive set (10 sec)
   ARP type: SNAP, ARP Timeout 4:00:00
   Ring speed: 16 Mbps
   Single ring node, Source Route Bridge capable
   Group Address: 0x00000000, Functional Address: 0x60840000
   Last input 0:00:01, output 0:00:01, output hang never
   Output queue 0/40, 0 drops; input queue 0/75, 0 drops
   Five minute input rate 0 bits/sec, 0 packets/sec
   Five minute output rate 0 bits/sec, 0 packets/sec
   16339 packets input, 1496515 bytes, 0 no buffer
        Received 9895 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     32648 packets output, 9738303 bytes, 0 underruns
0 output errors, 0 collisions, 2 interface resets, 0 restarts
     5 transitions
```

Table 6-39 describes significant fields shown in the display.

**Table 6-39 Show Interfaces Tokenring Field Descriptions**

| Field | Description |
|---|---|
| Token Ring is up \| down | Interface is either currently active and inserted into ring (up) or inactive and not inserted (down).<br>On the Cisco 7000 series, gives the interface processor type, slot number, and port number.<br>"Disabled" indicates the router has received over 5000 errors in a keepalive interval, which is 10 seconds by default. |
| Token Ring is Reset | Hardware error has occurred. |
| Token Ring is Initializing | Hardware is up, in the process of inserting the ring. |
| Token Ring is Administratively Down | Hardware has been taken down by an administrator. |
| line protocol is {up \| down \| administratively down} | Indicates whether the software processes that handle the line protocol believe the interface is usable (that is, whether keepalives are successful). |
| Hardware | Hardware type. "Hardware is Token Ring" indicates that the board is a CSC-R board. "Hardware is 16/4 Token Ring" indicates that the board is a CSC-R16 board. Also shows the address of the interface. |
| Internet address | Lists the Internet address followed by subnet mask. |
| MTU | Maximum Transmission Unit of the interface. |
| BW | Bandwidth of the interface in kilobits per second. |
| DLY | Delay of the interface in microseconds. |
| rely | Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over 5 minutes. |
| load | Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over 5 minutes. |
| Encapsulation | Encapsulation method assigned to interface. |
| loopback | Indicates whether loopback is set or not. |
| keepalive | Indicates whether keepalives are set or not. |
| ARP type: | Type of Address Resolution Protocol assigned. |
| Ring speed: | Speed of Token Ring—4 or 16 Mbps. |
| {Single ring/multiring node} | Indicates whether a node is enabled to collect and use source routing information (RIF) for routable Token Ring protocols. |
| Group Address: | Interface's group address, if any. The group address is a multicast address; any number of interfaces on the ring may share the same group address. Each interface may have at most one group address. |
| Last input | Number of hours, minutes, and seconds since the last packet was successfully received by an interface. Useful for knowing when a dead interface failed. |

| Field | Description |
|---|---|
| Last output | Number of hours, minutes, and seconds since the last packet was successfully transmitted by an interface. |
| output hang | Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the "last" fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed. |
| Last clearing | Time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared.<br>*** indicates the elapsed time is too large to be displayed.<br>0:00:00 indicates the counters were cleared more than $2^{31}$ms (and less than $2^{32}$ms) ago. |
| Output queue, drops<br>Input queue, drops | Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue. |
| Five minute input rate,<br>Five minute output rate | Average number of bits and packets transmitted per second in the last 5 minutes.<br><br>The 5-minute input and output rates should be used only as an approximation of traffic per second during a given 5-minute period. These rates are exponentially weighted averages with a time constant of 5 minutes. A period of four time constants must pass before the average will be within two percent of the instantaneous rate of a uniform stream of traffic over that period. |
| packets input | Total number of error-free packets received by the system. |
| bytes input | Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system. |
| no buffers | Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events. |
| broadcasts | Total number of broadcast or multicast packets received by the interface. |
| runts | Number of packets that are discarded because they are smaller than the medium's minimum packet size. |
| giants | Number of packets that are discarded because they exceed the medium's maximum packet size. |
| CRC | Cyclic redundancy checksum generated by the originating LAN station or far-end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRCs is usually the result of a station transmitting bad data. |
| frame | Number of packets received incorrectly having a CRC error and a noninteger number of octets. |

| Field | Description |
|---|---|
| overrun | Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data. |
| ignored | Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be increased. |
| packets output | Total number of messages transmitted by the system. |
| bytes output | Total number of bytes, including data and MAC encapsulation, transmitted by the system. |
| underruns | Number of times that the far-end transmitter has been running faster than the near-end router's receiver can handle. This may never be reported on some interfaces. |
| output errors | Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this may not balance with the sum of the enumerated output errors, as some datagrams may have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories. |
| collisions | Since a Token Ring cannot have collisions, this statistic is nonzero only if an unusual event occurred when frames were being queued or dequeued by the system software. |
| interface resets | Number of times an interface has been reset. The interface may be reset by the administrator or automatically when an internal error occurs. |
| Restarts | Should always be zero for Token Ring interfaces. |
| transitions | Number of times the ring made a transition from up to down, or vice versa. A large number of transitions indicates a problem with the ring or the interface. |

The following is sample output from the **show interfaces tokenring** command on a Cisco 7000:

```
Router# show interfaces tokenring 2/0

TokenRing2/0 is administratively down, line protocol is down
  Hardware is cxBus Token Ring, address is 0000.3040.8b4a (bia 0000.3040.8b4a)
  MTU 8136 bytes, BW 16000 Kbit, DLY 630 usec, rely 255/255, load 1/255
  Encapsulation SNAP, loopback not set, keepalive set (10 sec)
  ARP type: SNAP, ARP Timeout 4:00:00
  Ring speed: 0 Mbps
  Single ring node, Source Route Transparent Bridge capable
  Ethernet Transit OUI: 0x0000F8
  Last input never, output never, output hang never
  Last clearing of “show interface” counters never
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets, 0 restarts
     1 transitions
```

The following example on the Cisco 70000 includes the **accounting** option.When you use the accounting option, only the accounting statistics are displayed.

```
Router# show interfaces tokenring 2/0 accounting

TokenRing2/0
       Protocol    Pkts In   Chars In   Pkts Out  Chars Out
             IP       7344    4787842       1803    1535774
      Appletalk      33345    4797459      12781    1089695
        DEC MOP          0          0        127       9779
            ARP          7        420         39       2340
```

# show interfaces tunnel

To list tunnel interface information, use the **show interfaces tunnel** privileged EXEC command.

**show interfaces tunnel** *unit* [**accounting**]

## Syntax Description

| | |
|---|---|
| *unit* | Must match the interface port line number. |
| **accounting** | (Optional) Displays the number of packets of each protocol type that have been sent through the interface. |

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show interface tunnel** command:

```
Router# show interfaces tunnel 4

Tunnel4 is up, line protocol is down
  Hardware is Routing Tunnel
  MTU 1500 bytes, BW 9 Kbit, DLY 500000 usec, rely 255/255, load 1/255
  Encapsulation TUNNEL, loopback not set, keepalive set (10 sec)
  Tunnel source 0.0.0.0, destination 0.0.0.0
  Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Output queue 0/0, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets, 0 restarts
```

Table 6-40 describes significant fields shown in the display.

**Table 6-40 Show Interfaces Tunnel Field Descriptions**

| Field | Description |
|---|---|
| Tunnel is up \| down | Interface is currently active and inserted into ring (up) or inactive and not inserted (down).<br>On the Cisco 7000 series, gives the interface processor type, slot number, and port number. |
| line protocol is {up \| down \| administratively down} | Shows line protocol up if a valid route is available to the tunnel destination. Shows line protocol down if no route is available, or if the route would be recursive. |
| Hardware | Specifies the hardware type. |
| MTU | Maximum Transmission Unit of the interface. |

| Field | Description |
|---|---|
| BW | Bandwidth of the interface in kilobits per second. |
| DLY | Delay of the interface in microseconds. |
| rely | Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over 5 minutes. |
| load | Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over 5 minutes. |
| Encapsulation | Encapsulation method is always TUNNEL for tunnels. |
| loopback | Indicates whether loopback is set or not. |
| keepalive | Indicates whether keepalives are set or not. |
| Tunnel source | IP address used as the source address for packets in the tunnel. |
| destination | IP address of the host destination. |
| Tunnel protocol | Tunnel transport protocol (the protocol the tunnel is using). This is based on the **tunnel mode** command, which defaults to GRE. |
| key | ID key for the tunnel interface, unless disabled. |
| sequencing | Indicates whether the tunnel interface drops datagrams that arrive out of order. Can be disabled. |
| Last input | Number of hours, minutes, and seconds since the last packet was successfully received by an interface. Useful for knowing when a dead interface failed. |
| Last output | Number of hours, minutes, and seconds since the last packet was successfully transmitted by an interface. |
| output hang | Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the “last” fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed. |
| Last clearing | Time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared.<br>*** indicates the elapsed time is too large to be displayed.<br>0:00:00 indicates the counters were cleared more than $2^{31}$ms (and less than $2^{32}$ms) ago. |
| Output queue, drops<br>Input queue, drops | Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue. |
| Five minute input rate,<br>Five minute output rate | Average number of bits and packets transmitted per second in the last 5 minutes.<br>The 5-minute input and output rates should be used only as an approximation of traffic per second during a given 5-minute period. These rates are exponentially weighted averages with a time constant of 5 minutes. A period of four time constants must pass before the average will be within two percent of the instantaneous rate of a uniform stream of traffic over that period. |
| packets input | Total number of error-free packets received by the system. |
| bytes input | Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system. |

| Field | Description |
|---|---|
| no buffers | Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events. |
| broadcasts | Total number of broadcast or multicast packets received by the interface. |
| runts | Number of packets that are discarded because they are smaller than the medium's minimum packet size. |
| giants | Number of packets that are discarded because they exceed the medium's maximum packet size. |
| CRC | Cyclic redundancy checksum generated by the originating LAN station or far-end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRCs is usually the result of a station transmitting bad data. |
| frame | Number of packets received incorrectly having a CRC error and a noninteger number of octets. |
| overrun | Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data. |
| ignored | Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be increased. |
| abort | Illegal sequence of one bits on a serial interface. This usually indicates a clocking problem between the serial interface and the data link equipment. |
| packets output | Total number of messages transmitted by the system. |
| bytes output | Total number of bytes, including data and MAC encapsulation, transmitted by the system. |
| underruns | Number of times that the far-end transmitter has been running faster than the near-end router's receiver can handle. This may never be reported on some interfaces. |
| output errors | Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this may not balance with the sum of the enumerated output errors, as some datagrams may have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories. |
| collisions | Number of messages retransmitted due to an Ethernet collision. This usually is the result of an overextended LAN (Ethernet or transceiver cable too long, more than two repeaters between stations, or too many cascaded multiport transceivers). Some collisions are normal. However, if your collision rate climbs to around 4 or 5%, you should consider verifying that there is no faulty equipment on the segment and/or moving some existing stations to a new segment. A packet that collides is counted only once in output packets. |

| Field | Description |
|---|---|
| interface resets | Number of times an interface has been reset. The interface may be reset by the administrator or automatically when an internal error occurs. |
| Restarts | Number of times the controller was restarted because of errors. |

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**show interfaces**
**show ip route** †
**show route** †

# show interfaces vty

Use the **show interfaces vty** EXEC command to display information about virtual asynchronous interfaces.

**show interfaces vty** *number*

## Syntax Description

| | |
|---|---|
| *number* | Number of the virtual terminal (VTY) that has been configured for asynchronous protocol features (vty-async). |

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show interfaces vty** command:

Router# **show interfaces vty 17**

```
VTY-Async17 is up, line protocol is up
  Hardware is Virtual Async Serial
  Interface is unnumbered.  Using address of Ethernet0 (171.69.60.44)
MTU 1500 bytes, BW 9 Kbit, DLY 100000 usec, rely 255/255, load 1/255
  Encapsulation SLIP, loopback not set
  DTR is pulsed for 5 seconds on reset
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Output queue 0/10, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets, 0 restarts
     0 carrier transitions
```

Table 6-41 describes the fields shown in the sample display.

**Table 6-41 Show Interfaces VTY Field Descriptions**

| Field | Description |
|---|---|
| Async... is {up \| down \| administratively down} | Indicates whether the interface is currently active (whether carrier detect is present) and if it has been taken down by an administrator. |
| line protocol is {up \| down \| administratively down} | Indicates whether the software processes that handle the line protocol think the line is usable (that is, whether keepalives are successful). |
| Hardware is | Hardware type. |
| Internet address \| unnumbered | IP address, or IP unnumbered for the line. If unnumbered, the output lists the interface and IP address to which the line is assigned (Ethernet0 at 171.69.60.44 in this example). |
| MTU | Maximum transmission unit of the vty-async interface. |

| Field | Description |
|---|---|
| BW | Bandwidth of the vty-async interface in kilobits per second. |
| DLY | Delay of the vty-async interface in microseconds. |
| rely | Reliability of the vty-async interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over five minutes. |
| load | Load on the vty-async interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over five minutes. The calculation uses the value from the **bandwidth** interface configuration command. |
| Encapsulation | Encapsulation method assigned to the vty-async interface. |
| loopback | Test in which signals are sent and then directed back toward the source at some point along the communication path. Used to test network interface usability. |
| DTR | Data Terminal Ready. An RS232-C circuit that is activated to let the DCE know when the DTE is ready to send and receive data. |
| Last input | Number of hours, minutes, and seconds since the last packet was successfully received by a vty-async interface. Useful for knowing when a dead interface failed. |
| output | The number of hours, minutes, and seconds since the last packet was successfully transmitted by a vty-async interface. |
| output hang | Number of hours, minutes, and seconds (or never) since the vty-async interface was last reset because of a transmission that took too long. When the number of hours in any of the “last” fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed. |
| Last clearing | The time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared.<br>*** indicates the elapsed time is too large to be displayed.<br>0:00:00 indicates the counters were cleared more than $2^{31}$ms (and less than $2^{32}$ms) ago. |
| Output queue, drops<br>input queue, drops | Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue. |
| Five minute input rate,<br>Five minute output rate | Average number of bits and packets transmitted per second in the last five minutes. |
| packets input | Total number of error-free packets received by the system. |
| bytes | Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system. |
| no buffer | Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events. |
| broadcasts | Total number of broadcast or multicast packets received by the vty-async interface. |
| runts | Number of packets that are discarded because they are smaller than the medium’s minimum packet size. |

| Field | Description |
|---|---|
| giants | Number of packets that are discarded because they exceed the medium's maximum packet size. |
| input errors | Total number of no buffer, runts, giants, CRCs, frame, overrun, ignored, and abort counts. Other input-related errors can also increment the count, so that this sum might not balance with the other counts. |
| CRC | The cyclic redundancy checksum generated by the originating LAN station or far end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRC's is usually the result of collisions or a station transmitting bad data. On a serial link, CRC's usually indicate noise, gain hits or other transmission problems on the data link. |
| frame | Number of packets received incorrectly having a CRC error and a noninteger number of octets. On a serial line, this is usually the result of noise or other transmission problems. |
| overrun | Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data. |
| ignored | Number of received packets ignored by the vty-async interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be incremented. |
| abort | Illegal sequence of one bits on a vty-async interface. This usually indicates a clocking problem between the vty-async interface and the data link equipment. |
| packets output | Total number of messages transmitted by the system. |
| bytes | Total number of bytes, including data and MAC encapsulation, transmitted by the system. |
| underruns | Number of times that the far-end transmitter has been running faster than the near-end communication server's receiver can handle. This might never be reported on some vty-async interfaces. |
| output errors | Sum of all errors that prevented the final transmission of datagrams out of the vty-async interface being examined. Note that this might not balance with the sum of the enumerated output errors, as some datagrams might have more than one error, and others might have errors that do not fall into any of the specifically tabulated categories. |
| collisions | Number of packets colliding. |
| interface resets | Number of times a vty-async interface has been completely reset. This can happen if packets queued for transmission were not sent within several seconds. This can be caused by a malfunctioning modem that is not supplying the transmit clock signal, or by a cable problem. If the system notices that the carrier detect line of a vty-async interface is up, but the line protocol is down, it periodically resets the interface in an effort to restart it. Interface resets can also occur when a vty-async interface is looped back or shut down. |
| restarts | Number of times the controller was restarted because of errors. |

| Field | Description |
|---|---|
| carrier transitions | Number of times the carrier detect signal of a vty-async interface has changed state. Indicates modem or line problems if the carrier detect line is changing state often. |

# show ip interface

To list a summary of an interface's IP information and status, use the **show ip interface** privileged EXEC command.

**show ip interface** [**brief**] [*type*] [*number*]

## Syntax Description

| | |
|---|---|
| **brief** | (Optional) Displays a brief summary of IP status and configuration. |
| *type* | (Optional) Specifies that information be displayed about that interface type only. The possible value depends on the type of interfaces the system has. For example, it could be **ethernet**, **null**, **serial**, **tokenring**, etc. |
| *number* | (Optional) Interface number. |

## Command Mode

Privileged EXEC

## Sample Displays

The following is sample output from the **show ip interface** command:

```
Router# show ip interface

Ethernet0 is administratively down, line protocol is down
  Internet address is 1.0.46.10, subnet mask is 255.0.0.0
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is enabled
  Multicast groups joined: 224.0.0.1 224.0.0.2
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP SSE switching is disabled
  Router Discovery is disabled
  IP accounting is disabled
  TCP/IP header compression is disabled
  Probe proxy name replies are disabled
  Gateway Discovery is disabled
PCbus0 is administratively down, line protocol is down
  Internet address is 198.135.1.43, subnet mask is 255.255.255.0
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is enabled
  Multicast groups joined: 224.0.0.1 224.0.0.2
```

```
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP SSE switching is disabled
  Router Discovery is disabled
  IP accounting is disabled
  TCP/IP header compression is disabled
  Probe proxy name replies are disabled
  Gateway Discovery is disabled
Serial0 is administratively down, line protocol is down
  Internet address is 198.135.2.49, subnet mask is 255.255.255.0
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is enabled
  Multicast groups joined: 224.0.0.1 224.0.0.2
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP SSE switching is disabled
  Router Discovery is disabled
  IP accounting is disabled
  TCP/IP header compression is disabled
  Probe proxy name replies are disabled
  Gateway Discovery is disabled
```

The following is sample output from the **show ip interface brief** command:

```
Router# show ip interface brief

Interface    IP-Address      OK?  Method     Status                 Protocol
Ethernet0    1.0.46.10       YES  manual     administratively down  down
PCbus0       198.135.1.43    YES  manual     administratively down  down
Serial0      198.135.2.49    YES  manual     administratively down  down
```

The following is sample output from the **show ip interface brief pcbus 0** command:

```
Router# show ip interface brief pcbus 0

Interface    IP-Address      OK?  Method     Status                 Protocol
PCbus0       198.135.1.43    YES  manual     administratively down  down
```

## Related Command

**show interfaces**

# show rif

Use the **show rif** EXEC command to display the current contents of the RIF cache.

**show rif**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show rif** command:

```
Router# show rif

Codes: * interface, - static, + remote
Hardware Addr  How   Idle (min)  Routing Information Field
5C02.0001.4322 rg5           -   0630.0053.00B0
5A00.0000.2333 TR0           3   08B0.0101.2201.0FF0
5B01.0000.4444 -             -   -
0000.1403.4800 TR1           0   -
0000.2805.4C00 TR0           *   -
0000.2807.4C00 TR1           *   -
0000.28A8.4800 TR0           0   -
0077.2201.0001 rg5          10   0830.0052.2201.0FF0
```

In the display, entries marked with an asterisk (*) are the router/bridge's interface addresses. Entries marked with a dash (–) are static entries. Entries with a number are cached entries. If the RIF timeout is set to something other than the default of 15 minutes, the timeout is displayed at the top of the display.

Table 6-42 describes significant fields shown in the display.

**Table 6-42 Show RIF Cache Display Field Descriptions**

| Field | Description |
|---|---|
| Hardware Addr | Lists the MAC-level addresses. |
| How | Describes how the RIF has been learned. Possible values include a ring group (rg), or interface (TR). |
| Idle (min) | Indicates how long, in minutes, since the last response was received directly from this node. |
| Routing Information Field | Lists the RIF. |

# shutdown

To disable an interface, use the **shutdown** interface configuration command. To restart a disabled interface, use the **no shutdown** command.

**shutdown**
**no shutdown**

## Syntax Description

This command has no arguments or keywords.

## Default

Enabled

## Command Mode

Interface configuration

## Usage Guidelines

The **shutdown** command disables all functions on the specified interface. On serial interfaces, this command causes the DTR signal to be dropped. On Token Ring interfaces, this command causes the interface to be deinserted from the ring. On FDDI interfaces, this command causes the optical bypass switch, if present, to go into bypass mode.

This command also marks the interface as unavailable. To check whether an interface is disabled, use the EXEC command **show interfaces**. An interface that has been shut down is shown as administratively down in the display from this command.

## Examples

The following example turns off Ethernet interface 0:

```
interface ethernet 0
shutdown
```

The following example turns the interface back on:

```
interface ethernet 0
no shutdown
```

## Related Command

**show interfaces**

# shutdown (hub configuration)

To shut down a port on an Ethernet hub of a Cisco 2505 or Cisco 2507, use the **shutdown** hub configuration command. To restart the disabled hub, use the **no** form of this command.

**shutdown**
**no shutdown**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Hub configuration

## Example

The following example shuts down hub 0, ports 1 through 3:

```
hub ethernet 0 1 3
shutdown
```

## Related Command

**hub**

# smt-queue-threshold

To set the maximum number of unprocessed FDDI station management (SMT) frames that will be held for processing, use the **smt-queue-threshold** global configuration command. Use the **no smt-queue-threshold** command to restore the queue to the default.

**smt-queue-threshold** *number*
**no smt-queue-threshold**

## Syntax Description

| | |
|---|---|
| *number* | Number of buffers used to store unprocessed SMT messages that are to be queued for processing. Acceptable values are positive integers. |

## Default

The default threshold value is equal to the number of FDDI interfaces installed in the router.

## Command Mode

Global configuration

## Usage Guidelines

This command helps ensure that the routers keep track of FDDI *upstream* and *downstream* neighbors, particularly when a router includes more than one FDDI interface.

In FDDI, upstream and downstream neighbors are determined by transmitting and receiving SMT Neighbor Information Frames (NIFs). The router can appear to lose track of neighbors when it receives an SMT frame and the queue currently contains an unprocessed frame. This occurs because the router discards incoming SMT frames if the queue is full. Discarding SMT NIF frames can cause the router to lose its upstream or downstream neighbor.

---

**Note** Use this command carefully, because the SMT buffer is charged to the inbound interface (input hold queue) until the frame is completely processed by the system. Setting this value to a high limit can impact buffer usage and the ability of the router to receive routable packets or routing updates.

---

## Example

The following example specifies that the SMT queue can hold ten messages. As SMT frames are processed by the system, the queue is decreased by one:

```
smt-queue-threshold 10
```

# source-address

To configure source address control on a port on an Ethernet hub of a Cisco 2505 or Cisco 2507, use the **source-address** hub configuration command. To remove a previously defined source address, use the **no** form of this command.

**source-address** [*mac-address*]
**no source-address**

## Syntax Description

| | |
|---|---|
| *mac-address* | (Optional) MAC address in the packets that the hub will allow to access the network. |

## Default

Disabled

## Command Mode

Hub configuration

## Usage Guidelines

If you omit the MAC address, the hub uses the value in the last source address register, and if the address register is invalid, it will remember the first MAC address it receives on the previously specified port, and allow only packets from that MAC address onto that port.

## Examples

The following example configures the hub to allow only packets from MAC address 1111.2222.3333 on port 2 of hub 0:

```
hub ethernet 0 2
source-address 1111.2222.3333
```

The following example configures the hub use the value of the last source address register. If the address register is invalid, it will remember the first MAC address it receives on port 2, and allow only packets from the learned MAC address on port 2:

```
hub ethernet 0 2
source-address
```

## Related Command

**hub**

# squelch

To extend the Ethernet twisted-pair 10BaseT capability beyond the standard 100 meters on the Cisco 4000 platform, use the **squelch** interface configuration command. To restore the default, use the **no** form of this command.

**squelch** {**normal** | **reduced**}
**no squelch** {**normal** | **reduced**}

## Syntax Description

| | |
|---|---|
| **normal** | Allows normal capability. |
| **reduced** | Allows extended 10BaseT capability. |

## Default

Normal range

## Command Mode

Interface configuration

## Example

The following example extends the twisted-pair 10BaseT capability on the cable attached to Ethernet interface 2:

```
interface ethernet 2
squelch reduced
```

# timeslot

To enable framed mode on a G.703-E1 interface, use the **timeslot** interface configuration command. To restore the default, use the **no** form of this command or set the start slot to 0.

**timeslot** *start-slot* **–** *stop-slot*
**no timeslot**

## Syntax Description

| | |
|---|---|
| *start-slot* | The first subframe in the major frame. Range is 1 to 31 and must be less than or equal to *stop-slot*. |
| *stop-slot* | The last subframe in the major frame. Range is 1 to 31 and must be greater than or equal to *start-slot*. |

## Default

A G.703-E1 interface is configured for unframed mode.

## Command Mode

Interface configuration

## Usage Guidelines

This command applies to a Cisco 4000 router or Cisco 7000 series router. G.703-E1 interfaces have two modes of operation, framed and unframed. When in framed mode, the range from *start-slot* to *stop-slot* gives the number of 64-Kbps slots in use. There are 32 64-Kbps slots available.

## Example

The following example enables framed mode on a G.703-E1 interface:

```
timeslot 1-3
```

## Related Command

**ts16**

# transmit-clock-internal

When a DTE does not return a transmit clock, use the **transmit-clock-internal** interface command to enable the internally generated clock on a serial interface on a Cisco 7000. Use the **no** form of this command to disable the feature.

**transmit-clock-internal**
**no transmit-clock-internal**

## Syntax Description

This command has no keywords or arguments.

## Default

Disabled

## Command Mode

Interface configuration

## Example

In the following example, the internally generated clock is enabled on serial interface 3/0:

```
interface serial 3/0
transmit-clock-internal
```

# transmitter-delay

To specify a minimum dead-time after transmitting a packet, use the **transmitter-delay** interface configuration command. The **no transmitter-delay** command restores the default.

**transmitter-delay** {*microseconds* | *hdlc-flags*}
**no transmitter-delay**

## Syntax Description

| | |
|---|---|
| *microseconds* | Approximate number of microseconds of minimum delay after transmitting a packet on the MCI and SCI interface cards. |
| *hdlc-flags* | Minimum number of HDLC flags to be sent between each packet on the HIP, HSCI, FSIP, or HSSI. The valid range on the HSSI is 2 to 128000. |

## Default

0 microseconds

## Command Mode

Interface configuration

## Usage Guidelines

This command is especially useful for serial interfaces that can send back-to-back data packets over serial interfaces faster than some hosts can receive them.

The transmitter delay feature is implemented for the following Token Ring cards: CSC-R16, CSC-R16M, CSC-1R, CSC-2R, and CSC-CTR. For the first four cards, the command syntax is the same as the existing command and specifies the number of milliseconds to delay between sending frames that are generated by the router. Transmitter delay for the CSC-CTR uses the same syntax, but specifies a relative time interval to delay between transmission of all frames.

## Example

The following example specifies a delay of 300 microseconds on serial interface 0:

```
interface serial 0
transmitter-delay 300
```

# ts16

To control the use of time slot 16 for data on a G.703-E1 interface, use the **ts16** interface configuration command. To restore the default, use the **no** form of this command.

**ts16**
**no ts16**

## Syntax Description

This command has no arguments or keywords.

## Default

Time slot 16 is used for signaling.

## Command Mode

Interface configuration

## Usage Guidelines

This command applies to a Cisco 4000 router or Cisco 7000 series router. By default, time slot 16 is used for signaling. Use this command to configure time slot 16 to be used for data. When in framed mode, in order to get all possible subframes or timeslots, you must use the **ts16** command.

## Example

The following example configures time slot 16 to be used for data on a G.703-E1 interface:

```
ts16
```

## Related Command

**timeslot**

# tunnel checksum

To enable encapsulator-to-decapsulator checksumming of packets on a tunnel interface, use the **tunnel checksum** interface configuration command. To disable checksumming, use the **no** form of this command.

**tunnel checksum**
**no tunnel checksum**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

This command currently applies to generic route encapsulation (GRE) only. Some passenger protocols rely on media checksums to provide data integrity. By default, the tunnel does not guarantee packet integrity. By enabling end-to-end checksums, the routers will drop corrupted packets.

## Example

In the following example, all protocols will have encapsulator-to-decapsulator checksumming of packets on the tunnel interface:

```
tunnel checksum
```

# tunnel destination

To specify a tunnel interface's destination, use the **tunnel destination** interface configuration command. To remove the destination, use the **no** form of this command.

**tunnel destination** {*hostname* | *ip-address*}
**no tunnel destination**

## Syntax Description

| | |
|---|---|
| *hostname* | Name of the host destination |
| *ip-address* | IP address of the host destination expressed in decimal in four-part, dotted notation |

## Default

No tunnel interface destination is specified.

## Command Mode

Interface configuration

## Usage Guidelines

You cannot have two tunnels using the same encapsulation mode with exactly the same source and destination address. The workaround is to create a loopback interface and source packets off of the loopback interface.

## Examples

The following example enables Cayman tunneling:

```
interface tunnel0
tunnel source ethernet0
tunnel destination 131.108.164.19
tunnel mode cayman
```

The following example enables GRE tunneling:

```
interface tunnel0
appletalk cable-range 4160-4160 4160.19
appletalk zone Engineering
tunnel source ethernet0
tunnel destination 131.108.164.19
tunnel mode gre ip
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**appletalk cable-range**†
**appletalk zone**†
**tunnel mode**
**tunnel source**

# tunnel key

To enable an ID key for a tunnel interface, use the **tunnel key** interface configuration command. To remove the ID key, use the **no** form of this command.

**tunnel key** *key-number*
**no tunnel key**

## Syntax Description

| | |
|---|---|
| *key-number* | Integer from 0 to 4294967295 |

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

This command currently applies to generic route encapsulation (GRE) only. Tunnel ID keys can be used as a form of *weak* security to prevent misconfiguration or injection of packets from a foreign source.

**Note** When using GRE, the ID key is carried in each packet. We do *not* recommend relying on this key for security purposes.

## Example

In the following example, the tunnel key is set to 3:

```
tunnel key 3
```

# tunnel mode

To set the encapsulation mode for the tunnel interface, use the **tunnel mode** interface configuration command. To set to the default, use the **no** form of this command.

**tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre ip** | **nos**}
**no tunnel mode**

## Syntax Description

| | |
|---|---|
| **aurp** | AppleTalk Update Routing Protocol (AURP). |
| **cayman** | Cayman TunnelTalk AppleTalk encapsulation. |
| **dvmrp** | Distance Vector Multicast Routing Protocol . |
| **eon** | EON compatible CLNS tunnel. |
| **gre ip** | Generic route encapsulation (GRE) protocol over IP. |
| **nos** | KA9Q/NOS compatible IP over IP. |

## Default

GRE tunneling

## Command Mode

Interface configuration

## Usage Guidelines

You cannot have two tunnels using the same encapsulation mode with exactly the same source and destination address. The workaround is to create a loopback interface and source packets off of the loopback interface.

Cayman tunneling implements tunneling as designed by Cayman Systems. This enables our routers to interoperate with Cayman GatorBoxes. With Cayman tunneling, you can establish tunnels between two routers or between our router and a GatorBox. When using Cayman tunneling, you must not configure the tunnel with an AppleTalk network address. This means that there is no way to ping the other end of the tunnel.

Use DVMRP when a router connects to a mrouted router to run DVMRP over a tunnel. It is required to configure Protocol-Independent Multicast (PIM) and an IP address on a DVMRP tunnel.

Generic route encapsulation (GRE) tunneling can be done between our routers only. When using GRE tunneling for AppleTalk, you configure the tunnel with an AppleTalk network address. This means that you can ping the other end of the tunnel.

## Examples

The following example enables Cayman tunneling:

```
interface tunnel 0
tunnel source ethernet 0
tunnel destination 131.108.164.19
```

```
tunnel mode cayman
```

The following example enables GRE tunneling:

```
interface tunnel 0
appletalk cable-range 4160-4160 4160.19
appletalk zone Engineering
tunnel source ethernet0
tunnel destination 131.108.164.19
tunnel mode gre ip
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**appletalk cable-range**†
**appletalk zone**†
**tunnel destination**
**tunnel source**

# tunnel sequence-datagrams

To configure a tunnel interface to drop datagrams that arrive out of order, use the **tunnel sequence-datagrams** interface configuration command. To disable this function, use the **no** form of this command.

**tunnel sequence-datagrams**
**no tunnel sequence-datagrams**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

This command currently applies to generic route encapsulation (GRE) only. This command is useful when carrying passenger protocols that behave poorly when they receive packets out of order (for example, LLC2-based protocols).

## Example

In the following example, the tunnel is configured to drop datagrams that arrive out of order:

```
tunnel sequence-datagrams
```

# tunnel source

To set a tunnel interface's source address, use the **tunnel source** interface configuring command. To remove the source address, use the **no** form of this command.

**tunnel source** {*ip-address* | *type number*}
**no tunnel source**

## Syntax Description

| | |
|---|---|
| *ip-address* | IP address to use as the source address for packets in the tunnel. |
| *type* | All interface types. |
| *number* | Specifies the port, connector, or interface card number. The numbers are assigned at the factory at the time of installation or when added to a system, and can be displayed with the **show interfaces** command. |

## Default

No tunnel interface's source address is set.

## Command Mode

Interface configuration

## Usage Guidelines

You cannot have two tunnels using the same encapsulation mode with exactly the same source and destination address. The workaround is to create a loopback interface and source packets off of the loopback interface.

When using tunnels to Cayman boxes, you must set the **tunnel source** to an explicit IP address on the same subnet as the Cayman box, not the tunnel itself.

## Examples

The following example enables Cayman tunneling:

```
interface tunnel0
tunnel source ethernet0
tunnel destination 131.108.164.19
tunnel mode cayman
```

The following example enables GRE tunneling:

```
interface tunnel0
appletalk cable-range 4160-4160 4160.19
appletalk zone Engineering
tunnel source ethernet0
tunnel destination 131.108.164.19
tunnel mode gre ip
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**appletalk cable-range**†
**appletalk zone**†
**tunnel destination**

# tx-queue-limit

To control the number of transmit buffers available to a specified interface on the MCI and SCI cards, use the **tx-queue-limit** interface configuration command.

**tx-queue-limit** *number*

## Syntax Description

| | |
|---|---|
| *number* | Maximum number of transmit buffers that the specified interface can subscribe. |

## Default

Defaults depend on the total transmit buffer pool size and the traffic patterns of all the interfaces on the card. Defaults and specified limits are displayed with the **show controllers mci** EXEC command.

## Command Mode

Interface configuration

## Usage Guidelines

This command should be used only under the guidance of a technical support representative.

## Example

The following example sets the maximum number of transmit buffers on the interface to 5:

```
interface ethernet 0
tx-queue-limit 5
```

## Related Command

**show controllers mci**

PART 3

# Wide Area Networking

CHAPTER 7

# ATM Commands

This chapter describes the commands available to configure an Asynchronous Transfer Mode (ATM) interface in the Cisco 7000 series routers and Cisco 4500 routers, and to configure a serial interface for ATM access in other routers.

For ATM configuration information and examples, refer to the chapter entitled “Configuring ATM” in the *Router Products Configuration Guide*.

# atm aal aal3/4

To enable support for ATM adaptation layer 3/4 (AAL3/4) on an ATM interface, use the **atm aal aal3/4** interface configuration command.

**atm aal aal3/4**

## Syntax Description

This command has no arguments or keywords.

## Default

Support for AAL3/4 is disabled.

## Command Mode

Interface configuration

## Usage Guidelines

This command is supported on the Cisco 7000, but not on the Cisco 4500.

Only one virtual circuit can exist on a subinterface that is being used for AAL3/4 processing, and that virtual circuit must be an AAL3/4 virtual circuit.

The AAL3/4 support feature requires static mapping of all protocols except IP.

## Example

The following example enables AAL3/4 on ATM interface 2/0:

```
interface atm2/0
ip address 131.108.177.178 255.255.255.0
atm aal aal3/4
```

## Related Commands

**atm multicast**
**atm mid-per-vc**
**atm pvc**
**atm smds**
**interface atm**

# atm backward-max-burst-size-clp0

To change the maximum number of high-priority cells coming from the destination router to the source router at the burst level on the switched virtual circuit (SVC), use the **atm backward-max-burst-size-clp0** map-class configuration command. The **no** form of this command restores the default.

**atm backward-max-burst-size-clp0** *cell-count*
**no atm backward-max-burst-size-clp0**

## Syntax Description

| | |
|---|---|
| *cell-count* | Maximum number of high-priority cells coming from the destination router at the burst level. Default is -1. |

## Default

-1. The router does not request this quality of service (QOS) parameter of the ATM switch, so the switch provides a "best effort service." The switch will drop cells if there is not enough buffer space.

## Command Mode

Map-class configuration

## Usage Guidelines

The keyword **clp0** indicates that this command affects only cells with a cell loss priority (CLP) of 0 (high-priority cells).

## Example

The following example sets the maximum number of high-priority cells coming from the destination router at the burst level to 800 cells:

```
atm backward-max-burst-size-clp0 800
```

# atm backward-max-burst-size-clp1

To change the maximum number of low-priority cells coming from the destination router to the source router at the burst level on the SVC, use the **atm backward-max-burst-size-clp1** map-class configuration command. The **no** form of this command restores the default value.

**atm backward-max-burst-size-clp1** *cell-count*
**no atm backward-max-burst-size-clp1**

## Syntax Description

| | |
|---|---|
| *cell-count* | Maximum number of low-priority cells coming from the destination router at the burst level. Default is -1. |

## Default

-1. The router does not request this quality of service (QOS) parameter of the ATM switch, so the switch provides a "best effort service." The switch will drop cells if there is not enough buffer space.

## Command Mode

Map-class configuration

## Usage Guidelines

The keyword **clp1** indicates that this command affects only cells with a cell loss priority (CLP) of 1 (low-priority cells).

## Example

The following example sets the maximum number of low-priority cells coming from the destination router at the burst level to 100,000:

```
atm backward-max-burst-size-clp1 100000
```

# atm backward-peak-cell-rate-clp0

To change the peak rate of high-priority cells coming from the destination router to the source router on the SVC, use the **atm backward-peak-cell-rate-clp0** map-class configuration command. The **no** form of this command restores the default.

**atm backward-peak-cell-rate-clp0** *rate*
**no atm backward-peak-cell-rate-clp0**

## Syntax Description

| | |
|---|---|
| *rate* | Maximum rate in kilobits per second (kbps) that this SVC can receive high-priority cells from the destination router. Default is -1. Maximum upper range is 155,000 kbps. |

## Default

-1. The router does not request this quality of service (QOS) parameter of the ATM switch, so the switch provides a “best effort service.” The switch will drop cells if there is not enough buffer space.

## Command Mode

Map-class configuration

## Usage Guidelines

The keyword **clp0** indicates that this command affects only cells with a cell loss priority (CLP) of 0 (high-priority cells).

## Example

The following example sets the peak rate for high-priority cells from the destination router to 8000 kbps:

```
atm backward-peak-cell-rate-clp0 8000
```

# atm backward-peak-cell-rate-clp1

To change the peak rate of low-priority cells coming from the destination router to the source router on the SVC, use the **atm backward-peak-cell-rate-clp1** map-class configuration command. The **no** form of this command restores the default.

**atm backward-peak-cell-rate-clp1** *rate*
**no atm backward-peak-cell-rate-clp1**

## Syntax Description

| | |
|---|---|
| *rate* | Maximum rate in kilobits per second (kbps) that this SVC can receive low-priority cells from the destination router. Default is -1. Maximum upper range is 155,000 kbps. |

## Default

-1. The router does not request this quality of service (QOS) parameter of the ATM switch, so the switch provides a “best effort service.” The switch will drop cells if there is not enough buffer space.

## Command Mode

Map-class configuration

## Usage Guidelines

The keyword **clp1** indicates that this command affects only cells with a cell loss priority (CLP) of 1 (low-priority cells).

## Example

The following example sets the peak rate for low-priority cells from the destination router to 7000 kbps:

```
atm backward-peak-cell-rate-clp1 7000
```

# atm backward-sustainable-cell-rate-clp0

To change the sustainable rate of high-priority cells coming from the destination router to the source router on the SVC, use the **atm backward-sustainable-cell-rate-clp0** map-class configuration command. The **no** form of this command restores the default.

**atm backward-sustainable-cell-rate-clp0** *rate*
**no atm backward-sustainable-cell-rate-clp0**

## Syntax Description

| | |
|---|---|
| *rate* | Sustainable rate in kilobits per second (kbps) that this SVC can receive high-priority cells from the destination router. Default is -1. Maximum upper range is 155,000 kbps. |

## Default

-1. The router does not request this quality of service (QOS) parameter of the ATM switch, so the switch provides a "best effort service." The switch will drop cells if there is not enough buffer space.

## Command Mode

Map-class configuration

## Usage Guidelines

The keyword **clp0** indicates that this command affects only cells with a cell loss priority (CLP) of 0 (high-priority cells).

## Example

The following example sets the sustainable rate for high-priority cells from the destination router to 800 kbps:

```
atm backward-sustainable-cell-rate-clp0 800
```

# atm backward-sustainable-cell-rate-clp1

To change the sustainable rate of low-priority cells coming from the destination router to the source router on the SVC, use the **atm backward-sustainable-cell-rate-clp1** map-class configuration command. The **no** form of this command restores the default value.

**atm backward-sustainable-cell-rate-clp1** *rate*
**no atm backward-sustainable-cell-rate-clp1**

## Syntax Description

| | |
|---|---|
| *rate* | Sustainable rate in kilobits per second (kbps) that this SVC can receive low-priority cells from the destination router. Default is -1. Maximum upper range is 155,000 kbps. |

## Default

-1. The router does not request this quality of service (QOS) parameter of the ATM switch, so the switch provides a "best effort service." The switch will drop cells if there is not enough buffer space.

## Command Mode

Map-class configuration

## Usage Guidelines

The keyword **clp1** indicates that this command affects only cells with a cell loss priority (CLP) of 1 (low-priority cells).

## Example

The following example sets the sustainable rate for low-priority cells from the destination router to 700 kbps:

```
atm backward-sustainable-cell-rate-clp1 700
```

# atm clock internal

To cause the AIP to generate the transmit clock internally, use the **atm clock internal** interface configuration command. The **no** form of this command restores the default value.

**atm clock internal**
**no atm clock internal**

## Syntax Description

This command has no arguments or keywords.

## Default

The AIP uses the transmit clock signal from the remote connection (the line). The switch provides the clocking.

## Command Mode

Interface configuration

## Usage Guidelines

This command is meaningless on a 4B/5B PLIM.

## Example

The following example causes the AIP to generate the transmit clock internally:

```
atm clock internal
```

# atm exception-queue

To set the exception-queue length, use the **atm exception-queue** interface configuration command. The **no** form of this command restores the default value.

**atm exception-queue** *number*
**no atm exception-queue**

## Syntax Description

| | |
|---|---|
| *number* | Number of entries in the range of 8 to 256. Default is 32 entries. |

## Default

32 entries

## Command Mode

Interface configuration

## Usage Guidelines

This command is supported on the Cisco 7000, but not on the Cisco 4500.

The exception-queue is used for reporting ATM events, such as CRC errors.

## Example

In the following example, the exception-queue is set to 50 entries:

```
atm exception-queue 50
```

# atm forward-max-burst-size-clp0

To change the maximum number of high-priority cells going from the source router to the destination router at the burst level on the SVC, use the **atm forward-max-burst-size-clp0** map-class configuration command. The **no** form of this command restores the default value.

**atm forward-max-burst-size-clp0** *cell-count*
**no atm forward-max-burst-size-clp0**

## Syntax Description

| | |
|---|---|
| *cell-count* | Maximum number of high-priority cells going from the source router at the burst level. Default is -1. |

## Default

-1. The router does not request this quality of service (QOS) parameter of the ATM switch, so the switch provides a “best effort service.” The switch will drop cells if there is not enough buffer space.

## Command Mode

Map-class configuration

## Usage Guidelines

The keyword **clp0** indicates that this command affects only cells with a cell loss priority (CLP) of 0 (high-priority cells).

## Example

The following example sets the maximum number of high-priority cells going from the source router at the burst level to 100,000:

```
atm forward-max-burst-size-clp0 100000
```

# atm forward-max-burst-size-clp1

To change the maximum number of low-priority cells going from the source router to the destination router at the burst level on the SVC, use the **atm forward-max-burst-size-clp1** map-class configuration command. The **no** form of this command restores the default value.

**atm forward-max-burst-size-clp1** *cell-count*
**no atm forward-max-burst-size-clp1**

## Syntax Description

| | |
|---|---|
| *cell-count* | Maximum number of low-priority cells going from the source router at the burst level. Default is -1. |

## Default

-1. The router does not request this quality of service (QOS) parameter of the ATM switch, so the switch provides a "best effort service." The switch will drop cells if there is not enough buffer space.

## Command Mode

Map-class configuration

## Usage Guidelines

The keyword **clp1** indicates that this command affects only cells with a cell loss priority (CLP) of 1 (low-priority cells).

## Example

The following example sets the maximum number of low-priority cells going from the source router at the burst level to 100,000:

```
atm forward-max-burst-size-clp1 100000
```

# atm forward-peak-cell-rate-clp0

To change the peak rate of high-priority cells going from the source router to the destination router on the SVC, use the **atm forward-peak-cell-rate-clp0** map-class configuration command. The **no** form of this command restores the default value.

**atm forward-peak-cell-rate-clp0** *rate*
**no atm forward-peak-cell-rate-clp0**

## Syntax Description

| | |
|---|---|
| *rate* | Maximum rate in kilobits per second (kbps) that this SVC can send high-priority cells from the source router. Default is -1. Maximum upper range is 155,000 kbps. |

## Default

-1. The router does not request this quality of service (QOS) parameter of the ATM switch, so the switch provides a "best effort service." The switch will drop cells if there is not enough buffer space.

## Command Mode

Map-class configuration

## Usage Guidelines

The keyword **clp0** indicates that this command affects only cells with a cell loss priority (CLP) of 0 (high-priority cells).

## Example

The following example sets the peak high-priority cell rate from the source router to 1000 Kbps:

```
atm forward-peak-cell-rate-clp0 1000
```

# atm forward-peak-cell-rate-clp1

To change the peak rate of low-priority cells coming from the source router to the destination router on the SVC, use the **atm forward-peak-cell-rate-clp1** map-class configuration command. The **no** form of this command restores the default value.

**atm forward-peak-cell-rate-clp1** *rate*
**no atm forward-peak-cell-rate-clp1**

## Syntax Description

| | |
|---|---|
| *rate* | Maximum rate in kilobits per second (kbps) that this SVC can send low-priority cells from the source router. Default is -1. Maximum upper range is 155,000 kbps. |

## Default

-1. The router does not request this quality of service (QOS) parameter of the ATM switch, so the switch provides a "best effort service." The switch will drop cells if there is not enough buffer space.

## Command Mode

Map-class configuration

## Usage Guidelines

The keyword **clp1** indicates that this command affects only cells with a cell loss priority (CLP) of 1 (low-priority cells).

## Example

The following example sets the peak low-priority cell rate from the source router to 100,000 kbps:

```
atm forward-peak-cell-rate-clp1 100000
```

# atm forward-sustainable-cell-rate-clp0

To change the sustainable rate of high-priority cells coming from the source router to the destination router on the SVC, use the **atm forward-sustainable-cell-rate-clp0** map-class configuration command. The **no** form of this command restores the default value.

**atm forward-sustainable-cell-rate-clp0** *rate*
**no atm forward-sustainable-cell-rate-clp0**

## Syntax Description

| | |
|---|---|
| *rate* | Sustainable rate in kilobits per second (kbps) that this SVC can send high-priority cells from the source router. Default is -1. Maximum upper range is 155,000 kbps. |

## Default

-1. The router does not request this quality of service (QOS) parameter of the ATM switch, so the switch provides a "best effort service." The switch will drop cells if there is not enough buffer space.

## Command Mode

Map-class configuration

## Usage Guidelines

The keyword **clp0** indicates that this command affects only cells with a cell loss priority (CLP) of 0 (high-priority cells).

## Example

The following example sets the sustainable rate for high-priority cells from the source router to100,000 kbps:

```
atm forward-sustainable-cell-rate-clp0 100000
```

# atm forward-sustainable-cell-rate-clp1

To change the sustainable rate of low-priority cells coming from the source router to the destination router on the SVC, use the **atm forward-sustainable-cell-rate-clp1** map-class configuration command. The **no** form of this command restores the default value.

**atm forward-sustainable-cell-rate-clp1** *rate*
**no atm forward-sustainable-cell-rate-clp1**

## Syntax Description

| | |
|---|---|
| *rate* | Sustainable rate in kilobits per second (kbps) that this SVC can send low-priority cells from the source router. Default is -1. Maximum upper range is 155,000 kbps. |

## Default

-1. The router does not request this quality of service (QOS) parameter of the ATM switch, so the switch provides a "best effort service." The switch will drop cells if there is not enough buffer space.

## Command Mode

Map-class configuration

## Usage Guidelines

The keyword **clp1** indicates that this command affects only cells with a cell loss priority (CLP) of 1 (low-priority cells).

## Example

The following example sets the sustainable rate for low-priority cells from the source router to100,000 kbps:

```
atm forward-sustainable-cell-rate-clp1 100000
```

# atm maxvc

To set the ceiling value of the virtual circuit descriptor (VCD) on the AIP card, use the **atm maxvc** interface configuration command. The **no** form of this command restores the default value.

**atm maxvc** *number*
**no atm maxvc**

## Syntax Description

| | |
|---|---|
| *number* | Maximum number of supported virtual circuits. Valid values are 256, 512, 1024, 2048, or 4096. Default is 4096. |

## Default

4096 virtual circuits

## Command Mode

Interface configuration

## Usage Guidelines

This command is supported on the Cisco 7000, but not on the Cisco 4500.

This command sets the maximum value supported for the *vcd* argument in the **atm pvc** command. It also determines the maximum number of virtual circuits on which the AIP allows segmentation and reassembly (SAR) to occur.

However, if you set a **maxvc** limit and then enter the **atm pvc** command with a larger value for the *vcd* argument, the software does not generate an error message.

This command does not affect the VPI/VCI of each virtual circuit.

## Example

The following example sets a ceiling VCD value of 2048 and restricts the AIP to supporting at most 2048 virtual circuits:

```
atm maxvc 2048
```

# atm mid-per-vc

To limit the number of message identifier (MID) numbers allowed on each virtual circuit, use the **atm mid-per-vc** interface configuration command.

**atm mid-per-vc** *maximum*

## Syntax Description

| | |
|---|---|
| *maximum* | Number of MIDs allowed per virtual circuit on this interface. The values allowed are 16, 32, 64, 128, 256, 512, and 1024. The default is 16 MIDs per virtual circuit. |

## Default

The default limit is 16 MIDs per virtual circuit.

## Command Mode

Interface configuration

## Usage Guidelines

This command is supported on the Cisco 7000, but not on the Cisco 4500.

Message identifier (MID) numbers are used by receiving devices to reassemble cells from multiple sources into packets.

This command limits the number of discrete messages allowed on the PVC at the same time. It does not limit the number of cells associated with each message.

The *maximum* set by the **atm mid-per-vc** command overrides the range between the *midhigh* and *midlow* values set by the **atm pvc** command. If you set a *maximum* of 16 but a *midlow* of 0 and a *midhigh* of 255, only 16 MIDs (not 256) will be allowed on the virtual circuit.

## Example

The following example allows 64 MIDs per ATM virtual circuit:

```
atm mid-per-vc 64
```

## Related Command

**atm pvc**

# atm multicast

To assign an SMDS E.164 multicast address to the ATM subinterface that supports AAL3/4 and SMDS encapsulation, use the **atm multicast** interface configuration command.

**atm multicast** *address*

## Syntax Description

| | |
|---|---|
| *address* | Multicast E.164 address assigned to the subinterface. |

## Default

No multicast E.164 address is defined.

## Command Mode

Interface configuration

## Usage Guidelines

This command is supported on the Cisco 7000, but not on the Cisco 4500. The Cisco 4500 does not support AAL3/4.

Each AAL3/4 subinterface is allowed only one multicast E.164 address. This multicast address is used for all protocol broadcast operations.

## Example

The following example assigns a multicast E.164 address to the ATM subinterface that is being configured:

```
atm multicast e180.0999.000
```

## Related Commands

**atm aal aal3/4**
**atm pvc**
**atm smds**
**interface atm**

# atm nsap-address

To set the NSAP address for an ATM interface using SVC mode, use the **atm nsap-address** interface configuration command. The **no** form of this command removes any configured address for the interface.

**atm nsap-address** *nsap-address*
**no atm nsap-address**

## Syntax Description

| | |
|---|---|
| *nsap-address* | The 40-digit (hexadecimal) NSAP address of this interface (the source address). |

## Default

No NSAP address is defined for this interface.

## Command Mode

Interface configuration

## Usage Guidelines

When you are configuring an SVC, the **atm nsap-address** command is required, as it defines the source NSAP address. It identifies a particular port on the ATM network and must be unique across the network.

Configuring a new address on the interface will overwrite the previous address. The router considers the address as a string of bytes and will not prefix or suffix the address with any other strings or digits. The complete NSAP address must be specified, because this value will be used in the Calling Party Address Information Element in the SETUP message to establish a virtual circuit.

ATM NSAP addresses have a fixed length of 40 hexadecimal digits. You must configure the complete address in the following dotted format:

xx.xxxx.xx.xxxxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xx

---

**Note** All ATM NSAP addresses must be entered in the dotted hexadecimal format shown above, which conforms to the UNI specification.

---

## Example

In the following example, the source NSAP address for the interface is AB.CDEF.01.234567.890A.BCDE.F012.3456.7890.1234.12:

```
atm nsap-address AB.CDEF.01.234567.890A.BCDE.F012.3456.7890.1234.12
```

# atm pvc

To create a permanent virtual circuit (PVC) on the AIP interface, use the **atm pvc** interface configuration command. The **no** form of this command removes the specified PVC.

**atm pvc** *vcd vpi vci aal-encap* [[*midlow midhigh*] [*peak average burst*]]
**no atm pvc** *vcd vpi vci aal-encap* [[*midlow midhigh*] [*peak average burst*]]

## Syntax Description

*vcd* — Virtual circuit descriptor. A unique number per AIP that identifies to the AIP which VPI/VCI to use for a particular packet. Valid values range from 1 to the value set with the **atm maxvc** command. The AIP requires this feature to manage packet transmission. The vcd is not associated with the VPI/VCI used for the ATM network cells.

*vpi* — ATM network virtual path identifier (VPI) of this PVC, in the range from 0 through 255. The VPI is an 8-bit field in the header of the ATM cell. The VPI value is unique only on a single interface, not throughout the ATM network (it has local significance only).

Both *vpi* and *vci* cannot be specified as 0; if one is 0, the other cannot be 0.

*vci* — ATM network virtual channel identifier (VCI) of this PVC, in the range of 0 through 65535. The VCI is a 16-bit field in the header of the ATM cell. The VCI value is unique only on a single interface, not throughout the ATM network (it has local significance only).

Both *vpi* and *vci* cannot be specified as 0; if one is 0, the other cannot be 0.

*aal-encap* — ATM adaptation layer (AAL) and encapsulation type. When **aal5mux** is specified, a protocol is required. Possible values are as follows:

- **aal34smds** (encapsulation for SMDS networks); not supported on the Cisco 4500
- **aal5nlpid** (encapsulation that allows ATM interfaces to interoperate with HSSI interfaces that are using an ADSU and running ATM-DXI)
- **aal5mux decnet** (a MUX-type virtual circuit)
- **aal5mux ip** (a MUX-type virtual circuit)
- **aal5mux novell** (a MUX-type virtual circuit)
- **aal5mux vines** (a MUX-type virtual circuit)
- **aal5mux xns** (a MUX-type virtual circuit)
- **aal5snap** (LLC/SNAP precedes the protocol datagram).
- **qsaal** (a signaling-type PVC used for setting up or tearing down SVCs)

*midlow* — (Optional) Starting message identifier (MID) number for this PVC. The default is 0. If you set the *peak*, *average*, and *burst* values, you must also set the *midlow* and *midhigh* values.

*midhigh* — (Optional) Ending MID number for this PVC. The default is 0.If you set the *peak*, *average*, and *burst* values, you must also set the *midlow* and *midhigh* values.

| | |
|---|---|
| *peak* | (Optional) Maximum rate (in kbps) at which this virtual circuit can transmit. Valid values are in the range from 1 to the maximum rate set for a rate queue. The value should match a value specified by the **atm rate-queue** command. If you set this value, you must also specify a value for the *average*, *burst*, *midlow* and *midhigh* arguments. |
| *average* | (Optional) Average rate (in kbps) at which this virtual circuit will transmit. Valid values are in the range from 1 to the maximum rate set for a rate queue. If you set this value, you must also specify a value for the *peak*, *burst*, *midlow* and *midhigh* arguments. |
| *burst* | (Optional) Value (in the range 1 through 2047) that relates to the maximum number of ATM cells the virtual circuit can transmit to the network at the *peak* rate of the PVC. The actual burst cells equals *burst* * 32 cells, thereby allowing for a burst size of 32 cells to 65504 cells. The largest practical value of *burst* is the MTU size of the AIP card. If you set this value, you must also specify a value for the *peak* and *average* arguments. |

## Default

If *peak* and *average* rates are omitted, the PVC defaults to the highest bandwidth rate-queue available. *Peak* and *average* rates are then equal. By default, the virtual circuit is configured to run as fast as possible.

The default value of both *midlow* and *midhigh* is 0.

## Command Mode

Interface configuration

## Usage Guidelines

The IOS software dynamically creates rate queues as necessary to satisfy the requests of **atm pvc** commands. The software dynamically creates a rate queue when an **atm pvc** command specifies a peak/average rate that does not match any user-configured rate queue.

The **atm pvc** command creates a PVC and attaches it to the VPI and VCI specified. Both *vpi* and *vci* cannot be specified as 0; if one is 0, the other cannot be 0. The *aal-encap* argument determines the AAL mode and the encapsulation method used. The *peak* and *average* arguments determine the rate queue used.

Use one of the **aal5mux** encapsulation options to dedicate the specified virtual circuit to a single protocol; use the **aal5snap** encapsulation option to multiplex two or more protocols over the same virtual circuit. Whether you select **aal5mux** or **aal5snap** encapsulation might depend on practical considerations, such as the type of network and the pricing offered by the network. If the network's pricing depends on the number of virtual circuits set up, **aal5snap** might be the appropriate choice. If pricing depends on the number of bytes transmitted, **aal5mux** might be the appropriate choice because it has slightly less overhead.

If you choose to specify any of the *peak*, *average* and *burst* values, you must specify all three values. You can specify *midlow* and *midhigh* values only if you have also specified the *peak*, *average*, and *burst* values.

Message identifier (MID) numbers are used by receiving devices to reassemble cells from multiple sources into packets. You can assign different *midlow* to *midhigh* ranges to different PVCs to ensure that the message identifiers will be unique at the receiving end and, therefore, that messages can be reassembled correctly.

If you are configuring an SVC, this command is required to configure the PVC that handles the SVC call setup and termination. In this case, specify **qsaal** for the *aal-encap* argument. See the second example that follows.

## Examples

The following example creates a PVC with VPI 0 and VCI 6. The PVC uses AAL aal5mux with IP protocol.

```
atm pvc 1 0 6 aal5mux ip
```

The following example creates a PVC with VPI 0 and VCI 6. The PVC uses AAL aal3/4-SMDS protocol.

```
atm pvc 1 0 6 aal34smds 0 15 150000 70000 10
```

The following example creates a PVC to be used for ATM signaling for an SVC. It specifies VPI 0 and VCI 5.

```
atm pvc 1 0 5 qsaal
```

Assuming that no static rate queue has been defined, the following example creates the PVC and also creates a dynamic rate queue with the peak rate set to the maximum allowed by the PLIM and the average set to equal the peak rate:

```
atm pvc 1 1 1 aal5snap
```

Assuming that no static rate queue has been defined, the following example creates the PVC and also creates a dynamic rate queue with the peak rate set to100 Mbps (100,000 Kbps), the average rate set to 50 Mbps (50,000 Kbps), and a burst size of 64 cells (2 * 32 cells):

```
atm pvc 1 1 1 aal5snap 100000 50000 2
```

## Related Commands

**atm aal aal3/4**
**atm maxvc**
**atm multicast**
**atm rate-queue**
**atm smds**
**mtu**

# atm rate-queue

To create a permanent rate queue for the AIP, use the **atm rate-queue** interface configuration command. The **no** form of this command removes the rate queue.

**atm rate-queue** *queue-number speed*
**no atm rate-queue**

## Syntax Description

| | |
|---|---|
| *queue-number* | Queue number in the range 0 through 7. Queues 0 through 3 are in the high-priority bank and queues 4 through 7 are in the low-priority bank. |
| *speed* | Speed in megabits per second (Mbps) in the range from 1 through 155. The maximum speed is determined by the detected PLIM type on the AIP:<br>• 34 Mbps for E3<br>• 45 Mbps for DS-3 (when available)<br>• 100 Mbps for TAXI<br>• 155 Mbps for SONET |

## Default

No rate-queue is defined.

## Command Mode

Interface configuration

## Usage Guidelines

If you do not create permanent rate queues or if you create PVCs with peak/average rates that are not matched by the rate queues you configure, the software will dynamically create rate queues as necessary to satisfy the requests of the **atm pvc** commands.

You can create multiple rate queues. A warning message appears if all rate queues are deconfigured or if the combined rate-queues exceed the PLIM rate.

## Example

In the following example, rate queue 1 is configured for 100 Mbps:

```
atm rate-queue 1 100
```

## Related Command

**atm pvc**

# atm rawq-size

To define the AIP raw-queue size, use the **atm rawq-size** interface configuration command. The **no** form of this command restores the default value.

**atm rawq-size** *number*
**no atm rawq-size**

## Syntax Description

| | |
|---|---|
| *number* | Maximum number of cells in the raw queue simultaneously, in the range 8 through 256. Default is 32. |

## Default

32 cells

## Command Mode

Interface configuration

## Usage Guidelines

This command is supported on the Cisco 7000, but not on the Cisco 4500.

The raw queue is used for raw ATM cells, which include OAM (F4 and F5) and Interim Local Management Interface (ILMI) cells.

## Example

In the following example, a maximum of 48 cells are allowed in the raw queue:

```
atm rawq-size 48
```

# atm rxbuff

To set the maximum number of Receive buffers for simultaneous packet reassembly, use the **atm rxbuff** interface configuration command. The **no** form of this command restores the default value.

**atm rxbuff** *number*
**no atm rxbuff**

## Syntax Description

| | |
|---|---|
| *number* | Maximum number of packet reassemblies that the AIP can perform simultaneously, in the range 0 through 512. Default is 256. |

## Default

256 packet reassemblies

## Command Mode

Interface configuration

## Example

This command is supported on the Cisco 7000, but not on the Cisco 4500.

In the following example, the AIP can perform a maximum of 300 packet reassemblies simultaneously:

```
atm rxbuff 300
```

# atm smds-address

To assign a unicast E.164 address to the ATM subinterface that supports AAL3/4 and SMDS encapsulation, use the **atm smds-address** interface configuration command.

**atm smds-address** *address*

## Syntax Description

| | |
|---|---|
| *address* | Unicast E.164 address assigned to the subinterface. |

## Default

No E.164 address is assigned.

## Command Mode

Interface configuration

## Usage Guidelines

This command is supported on the Cisco 7000, but not on the Cisco 4500.

Each AAL3/4 subinterface is allowed only one unicast E.164 address.

## Example

The following example assigns a unicast E.164 address to the ATM subinterface that is being configured:

```
atm smds-address c141.555.1212
```

## Related Commands

**atm aal aal3/4**
**atm multicast**
**atm pvc**
**interface atm**

# atm sonet stm-1

To set the proper mode of operation for the SONET PLIM, use the **atm sonet stm-1** interface configuration command. The **no** form of this command restores the default.

**atm sonet stm-1**
**no atm sonet stm-1**

## Syntax Description

This command has no arguments or keywords.

## Default

STS-3C

## Command Mode

Interface configuration

## Usage Guidelines

Use STM-1 in applications where the ATM switch requires "unassigned cells" for rate adaptation. Use the default (STS-3C) in applications where the ATM switch requires "idle cells" for rate adaptation.

## Example

The following example specifies ATM SONET STM-1:

```
atm sonet stm-1
```

# atm txbuff

To set the maximum number of Transmit buffers for simultaneous packet fragmentation, use the **atm txbuff** interface configuration command. The **no** form of this command restores the default value.

**atm txbuff** *number*
**no atm txbuff**

## Syntax Description

| | |
|---|---|
| *number* | Maximum number of packet fragmentations that the AIP can perform simultaneously, in the range 0 through 512. Default is 256. |

## Default

256 packet fragmentations

## Command Mode

Interface configuration

## Usage Guidelines

This command is supported on the Cisco 7000, but not on the Cisco 4500.

## Example

In the following example, the AIP is configured to perform up to 300 packet fragmentations simultaneously:

```
atm txbuff 300
```

# atm vc-per-vp

To set the maximum number of VCIs to support per VPI, use the **atm vc-per-vp** interface configuration command. The **no** form of this command restores the default value.

**atm vc-per-vp** *number*
**no atm vc-per-vp**

## Syntax Description

| | |
|---|---|
| *number* | Maximum number of VCIs to support per VPI. On the Cisco 7000 AIP, valid values are 32, 64, 128, 256, 512, 1024, 2048, or 4096. On the Cisco 4500 NPM, valid values are 32, 64, 128, 256, 512, 1024, 2048, 4096, or 8192. Default is 1024. |

## Default

1024

## Command Mode

Interface configuration

## Usage Guidelines

This command controls the memory allocation in the AIP to deal with the VCI table. It defines the maximum number of VCIs to support per VPI; it does not bound the VCI numbers.

An invalid VCI causes a warning message to be displayed.

## Example

In the following example, the maximum number of VCIs to support per VPI is set to 512:

```
atm vc-per-vp 512
```

## Related Command

**atm pvc**

# atm vp-filter

To set the AIP filter register, use the **atm vp-filter** interface configuration command. The **no** form of this command restores the default value.

**atm vp-filter** *hexvalue*
**no atm vp-filter**

## Syntax Description

| | |
|---|---|
| *hexvalue* | Value in hexadecimal format. Default is 0x7B. |

## Default

0x7B

## Command Mode

Interface configuration

## Usage Guidelines

This command is supported on the Cisco 7000 AIP, but not on the Cisco 4500 NPM.

This command configures the hexadecimal value used in the VP filter register in the reassembly operation. The VP filter comprises 16 bits. The VP Filter Register uses the most significant bits (bits 15 through 8, the left half of the filter) as mask bits and uses bits 7 through 0 (the right half of the filter) as compare bits. When a cell is received, the right half of the filter is exclusively NORed with the binary value of the incoming VPI. The result is then ORed with the left half of the filter (the mask). If the result is all ones, then reassembly is done using the VCI/MID table (AAL3/4 processing). Otherwise, reassembly is done using the VPI/VCI table (AAL5 processing).

In other words, this command allows a way to specify which VPI (or range of VPIs) will be used for AAL3/4 processing; all other VPIs map to AAL5 processing. If only AAL5 processing is desired, the VP filter can default or be set to an arbitrary VPI and AAL5 processing will be performed on all VPIs.

## Examples

In the following example, all incoming cells will be reassembled using AAL3/4 processing:

```
atm vp-filter ff00
```

In the following example, all incoming cells with VP=0 will be reassembled using AAL3/4 processing; all other cells will be reassembled using AAL5 processing:

```
atm vp-filter 0
```

In the following example, all incoming cells with the most significant bit of the VP set will be reassembled using AAL3/4; all other cells will be reassembled using AAL5 processing:

```
atm vp-filter 7f80
```

# atm-nsap

To define an ATM map statement for an SVC, use the **atm-nsap** map-list configuration command in conjunction with the **map-list** global configuration command. The **no** form of this command removes the address.

*protocol protocol-address* **atm-nsap** *atm-nsap-address* [**class** *class-name*] [**broadcast**]
**no** *protocol protocol-address* **atm-nsap** *atm-nsap-address* [**class** *class-name*] [**broadcast**]

## Syntax Description

| | |
|---|---|
| *protocol* | One of the following keywords: **appletalk**, **apollo**, **bridge**, **clns**, **decnet**, **ip**, **ipx**, **vines**, **xns**. |
| *protocol-address* | Destination address that is being mapped to this SVC. |
| *atm-nsap-address* | Destination ATM NSAP address. Must be exactly 40 hexadecimal digits long and in the correct dotted format. |
| **class** | (Optional) Keyword. |
| *class-name* | (Optional) Name of a table that contains encapsulation-specific parameters. Such a table can be shared between maps that have the same encapsulation. |
| **broadcast** | (Optional) Indicates this map entry is to be used when the corresponding *protocol* wants to send broadcast packets to the interface (for example, IGRP updates). |

## Default

No map statements are defined.

## Command Mode

Map-list configuration

## Usage Guidelines

This command is required with the **map-list** command when you are configuring an SVC.

## Example

In the following example, a map list named atmsvc includes one map statement for a destination address being mapped:

```
map-list atmsvc
ip 131.108.97.17 atm-nsap AB.CDEF.01.234567.890A.BCDE.F012.3456.7890.1234.12 class qos
broadcast
```

## Related Command

**map-list**

# atm-vc

To define an ATM map statement for a PVC, use the **atm-vc** map-list configuration command in conjunction with the **map-list** global configuration command. The **no** form of this command removes the address.

*protocol protocol-address* **atm-vc** *vcd* [**broadcast**]
**no** *protocol protocol-address* **atm-vc** *vcd* [**broadcast**]

## Syntax Description

| | |
|---|---|
| *protocol* | One of the following keywords: **appletalk**, **apollo**, **bridge**, **clns**, **decnet**, **ip**, **ipx**, **vines**, **xns**. |
| *protocol-address* | Destination address that is being mapped to this PVC. |
| *vcd* | Virtual circuit descriptor of the PVC. |
| **broadcast** | (Optional) Indicates that this map entry is to be used when the corresponding *protocol* wants to send broadcast packets to the interface (for example, IGRP updates). Provides pseudo-broadcasting support. |

## Default

No map statements are defined.

## Command Mode

Map-list configuration

## Usage Guidelines

When operating in PVC mode, multicast capabilities may not exist in the ATM switch. For this reason, all static maps for a specific protocol should be marked as **broadcast** for multicasting. When a protocol is sending a packet to its multicast address, all static maps marked as **broadcast** will get a copy of that packet. This procedure simulates the multicast environment of a LAN.

Some switches may have point-to-multipoint PVCs that do the equivalent process. If one exists, then that PVC may be used as the sole **broadcast** PVC for all multicast requests.

## Example

In the following example, a map list named atm includes two map statements for protocol addresses being mapped:

```
map-list atm
ip 131.108.168.112 atm-vc 1 broadcast
decnet 10.2 atm-vc 2 broadcast
```

## Related Command

**map-list**

# atmsig close

To disconnect an SVC, use the **atmsig close** EXEC command.

**atmsig close atm** *slot*/**0** *vcd*

## Syntax Description

| | |
|---|---|
| *slot* | Slot of the SVC to close. |
| *vcd* | Virtual circuit descriptor of the signaling PVC to close. |

## Command Mode

EXEC

## Usage Guidelines

Since the AIP does not perform packet-level accounting on a per-virtual circuit basis, the interface does not close an idle SVC automatically. You must execute this command if you want to close a particular SVC. Since virtual circuits are numbered per interface, you must specify which ATM interface by its slot number.

## Example

The following example closes SVC 2 on ATM interface 4/0:

```
atmsig close atm4/0 2
```

# dxi map

To map a protocol address to a given VPI and VCI, use the **dxi map** interface configuration command. Use the **no** form of this command to remove the mapping for that protocol and protocol address.

**dxi map** *protocol protocol-address vpi vci* [**broadcast**]
**no dxi map** *protocol protocol-address*

## Syntax Description

| | |
|---|---|
| *protocol* | The bridging or protocol keyword: **apollo**, **appletalk**, **bridge**, **clns**, **decnet**, **ip**, **novell**, **vines**, or **xns**. |
| *protocol-address* | Protocol-specific address. |
| *vpi* | Virtual path identifier in the range 0 to 15. |
| *vci* | Virtual circuit identifier in the range 0 to 63. |
| **broadcast** | (Optional) Broadcasts should be forwarded to this address. |

## Default

No map definition is established.

## Command Mode

Interface configuration

## Usage Guidelines

This command is used in configurations where the router is intended to communicate with an ATM network through an ATM Data Service Unit (ADSU). Given the circuit identifier parameters (VPI and VCI) for the ATM permanent virtual circuit, the router computes and uses the DXI frame address (DFA) that is used for communication between the router and the ADSU.

The **dxi map** command can be used only on a serial interface or HSSI configured for ATM-DXI encapsulation.

## Example

In the following example, all IP packets intended for the host with IP address 131.108.170.49 are converted into ATM cells identified with a VPI of 2 (binary 0000 0010) and a VCI of 46 (binary 0000 0000 0010 1110) by the ADSU.

```
interface serial 0
dxi map ip 131.108.170.49 2 46 broadcast
```

Using the mapping defined in Annex A of the ATM DXI Specification, the router will use the VPI and VCI information in this example to compute a DFA of 558 (binary 1000101110). The ADSU will use DFA of the incoming frame to extract the VPI and VCI information when formulating ATM cells.

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**dxi pvc**
**encapsulation atm-dxi**
**interface serial** †

# dxi pvc

Use the **dxi pvc** interface configuration command to configure multiprotocol or single protocol ATM-DXI encapsulation. The **no** form of this command disables multiprotocol ATM-DXI encapsulation.

**dxi pvc** *vpi vci* [**snap** | **nlpid** | **mux**]
**no dxi pvc** *vpi vci* [**snap** | **nlpid** | **mux**]

## Syntax Description

| | |
|---|---|
| *vpi* | ATM network virtual path identifier (VPI) of this PVC, in the range from 0 through 255. The VPI is an 8-bit field in the header of the ATM cell. The VPI value is unique only on a single interface, not throughout the ATM network (it has local significance only).<br>Both *vpi* and *vci* cannot be specified as 0; if one is 0, the other cannot be 0. |
| *vci* | ATM network virtual channel identifier (VCI) of this PVC, in the range of 0 through 65535. The VCI is a 16-bit field in the header of the ATM cell. The VCI value is unique only on a single interface, not throughout the ATM network (it has local significance only).<br>Both *vpi* and *vci* cannot be specified as 0; if one is 0, the other cannot be 0. |
| **snap** | (Optional) LLC/SNAP encapsulation based on the protocol used in the packet. This keyword defines a PVC that can carry multiple network protocols. This is the default. |
| **nlpid** | (Optional) RFC 1294/1490 encapsulation. This option is provided for backward compatibility with the default encapsulation in earlier versions of the Cisco IOS. |
| **mux** | (Optional) MUX encapsulation; the carried protocol is defined by the **dxi map** command when the PVC is set up. This keyword defines a PVC that carries only one network protocol. |

## Default

LLC/SNAP encapsulation.

## Command Mode

Interface configuration

## Usage Guidelines

This command can be used only on a serial interface or HSSI that is configured with ATM-DXI encapsulation.

Select the **nlpid** option if software earlier than Release 10.3 was loaded previously on this router and the router was configured for the default encapsulation, which was **nlpid** in pre-10.3 releases.

## Examples

The following example configures ATM-DXI MUX encapsulation on serial interface 1. The PVC identified by a VPI of 10 and a VCI of 10 will carry a single protocol. Then the protocol to be carried on this PVC is defined by the **dxi map** command.

```
interface serial 1
dxi pvc 10 10 mux
dxi map ip 131.108.176.45 10 10 broadcast
```

The following example configures ATM-DXI NLPID encapsulation on serial interface 1. The PVC identified by a VPI of 11 and a VCI of 11 can carry multiprotocol traffic that is encapsulated with a header described in RFC 1294/1490.

```
interface serial 0
dxi pvc 11 11 nlpid
```

## Related Commands

**dxi map**
**encapsulation atm-dxi**
**show dxi pvc**

# loopback plim

To place the AIP into loopback mode, use the **loopback plim** interface configuration command. The **no** form of this command removes the loopback.

**loopback plim**
**no loopback plim**

## Syntax Description

This command has no arguments or keywords.

## Default

Packets go from the AIP to the ATM network.

## Command Mode

Interface configuration

## Usage Guidelines

This command is useful for testing because it loops all packets from the AIP back to the AIP as well as directing the packets to the network.

## Example

The following example places the AIP into loopback mode:

```
loopback plim
```

# map-class

To define quality of service (QOS) parameters that are associated with a static map for an SVC, use the **map-class** global configuration command. The **no** form of this command deletes this class.

**map-class** *encapsulation class-name*
**no map-class** *encapsulation class-name*

## Syntax Description

| | |
|---|---|
| *encapsulation* | Encapsulation type. One of the following: **atm**, **dialer**, **frame-relay**, **smds**, or **x25**. |
| *class-name* | User-assigned name of the QOS parameters table. |

## Default

No QOS parameters are defined.

## Command Mode

Global configuration

## Usage Guidelines

If the map class identified by *class-name* does not already exist, the router creates a new one. In either case, this command specifies the map class to which subsequent encapsulation-specific commands apply. Configuration of a map class is allowed only if the subsystem corresponding to the encapsulation is linked.

It is up to the media-specific routing that uses a static map to ensure that the referenced class exists if parameters are required.

## Example

The following example establishes QOS parameters for map-class atmclass1 and map-class atmclass2:

```
map-list atmlist
ip 131.108.170.21 atm-vc 12
ip 131.108.180.121 atm-nsap 12.3456.7890.abcd.0000.00 broadcast
ip 131.108.190.221 atm-vc 88 class atmclass1
decnet 10.23 atm-vc 33 class atmclass2 broadcast
map-class atm atmclass1
atm forward-peak-cell-rate-clp0 8000
atm backward-peak-cell-rate-clp0 8000
map-class atm atmclass2
atm forward-peak-cell-rate-clp1 7000
atm backward-peak-cell-rate-clp1 7000
atm backward-sustainable-cell-rate-clp0 800
interface atm 2/0
map-group atmlist
```

## Related Commands

**atm backward-peak-cell-rate-clp0**
**atm backward-peak-cell-rate-clp1**
**atm backward-max-burst-size-clp0**
**atm backward-max-burst-size-clp1**
**atm backward-sustainable-cell-rate-clp0**
**atm backward-sustainable-cell-rate-clp1**
**atm forward-peak-cell-rate-clp0**
**atm forward-peak-cell-rate-clp1**
**atm forward-max-burst-size-clp0**
**atm forward-max-burst-size-clp1**
**atm forward-sustainable-cell-rate-clp0**
**atm forward-sustainable-cell-rate-clp1**

# map-group

To associate an ATM map list to an interface or subinterface for either a PVC or SVC, use the **map-group** interface configuration command. The **no** form of this command removes the reference to the map list.

**map-group** *name*
**no map-group** *name*

## Syntax Description

| | |
|---|---|
| *name* | Name of the map list identified by the **map-list** command. |

## Default

No ATM map lists are associated.

## Command Mode

Interface configuration

## Usage Guidelines

More than one map-group can be configured for an interface.

## Example

In the following example, the map list named atm is associated with the ATM interface:

```
interface atm 2/0
map-group atm
```

## Related Command

**map-list**

# map-list

To define an ATM map statement for either a PVC or SVC, use the **map-list** global configuration command. The **no** form of this command deletes this list and all associated map statements.

**map-list** *name*
**no map-list** *name*

## Syntax Description

| | |
|---|---|
| *name* | Name of the map list. |

## Default

No map statements are defined.

## Command Mode

Global configuration

## Usage Guidelines

ATM currently does not provide broadcasting or multicasting capabilities. To allow the router to propagate routing updates and ARP requests, a static map that maps the protocol address and the ATM address of the next-hop ATM station must be configured. The router supports a mapping scheme that identifies the ATM address of remote hosts/routers. This address can be specified either as a virtual circuit descriptor (*vcd*) for a PVC or an NSAP address for an SVC.

The **map-list** command specifies the map list to which the subsequent map-list configuration commands apply. These map-list configuration commands identify destination addresses. One map list can contain multiple map entries. A map-list can be referenced by more than one interface.

## Examples

In the following example for a PVC, a map list named atm is followed by two map statements for protocol addresses being mapped:

```
map-list atm
ip 131.108.168.112 atm-vc 1 broadcast
decnet 10.2 atm-vc 2 broadcast
```

In the following example for an SVC, a map list named atm includes two map statements for protocol addresses being mapped:

```
map-list atm
ip 131.108.97.165 atm-nsap BC.CDEF.01.234567.890A.BCDE.F012.3456.7890.1234.13
ip 131.108.97.166 atm-nsap BC.CDEF.01.234567.890A.BCDE.F012.3456.7890.1234.12
```

## Related Commands

**atm-nsap**
**atm-vc**
**map-group**

# show atm interface atm

To display ATM-specific information about an interface, use the **show atm interface atm** privileged EXEC command.

**show atm interface atm** *slot*/**0**

**show atm interface atm** *number* (Cisco 4500 )

## Syntax Description

| | |
|---|---|
| *slot* | Slot number of the AIP. |

## Command Mode

Privileged EXEC

## Sample Display

The following is sample output from the **show atm interface atm** command to display statistics on slot 4, port 0:

```
Router# show atm interface atm 4/0

ATM interface ATM4/0:
AAL enabled: AAL5, Maximum VCs: 1024, Current VCs: 6
Tx buffers 256, Rx buffers 256, Exception Queue: 32, Raw Queue: 32
VP Filter: 0x7B, VCIs per VPI: 1024, Max Datagram Size:4496, MIDs/VC:16
PLIM Type:4B5B - 100Mbps, No Framing, TX clocking: LINE
4897 input, 2900 output, 0 IN fast, 0 OUT fast
Rate-Queue 1 set to 100Mbps, reg=0x4EA DYNAMIC, 1 VCCs
ATM4/0.1:AAL3/4-SMDS address c111.1111.1111 Multicast e222.2222.222
Config. is ACTIVE
```

Table 7-1 describes the fields shown in the display.

**Table 7-1 Show ATM Interface ATM Field Descriptions**

| Field | Description |
|---|---|
| ATM interface | Slot/port number of the interface. |
| AAL enabled | Type of AAL. If both AAL5 and AAL3/4 are enabled on the interface, the output will include both AAL5 and AAL3/4. |
| Maximum VCs | Maximum number of virtual circuits this interface can support. |
| Current VCs | Number of active virtual circuits. |
| Tx buffers, Rx buffers | Number of buffers configured with the **atm txbuff** or **atm rxbuff** command, respectively. |
| Exception Queue | Number of buffers configured with the **atm exception-queue** command. |
| Raw Queue | Queue size configured with the **atm rawq-size** command. |
| VP Filter | Hexadecimal value of the VP filter as configured by the **atm vp-filter** command. |

| Field | Description |
|---|---|
| VCIs per VPI | Maximum number of VCIs to support per VPI, as configured by the **atm vc-per-vp** command. |
| Max Datagram Size | The configured maximum number of bytes in the largest datagram. |
| MIDs/VC | The configured maximum number of message identifiers allowed per virtual circuit on this interface. |
| PLIM Type | Physical Layer Interface Module (PLIM) type (E3, 4B/5B, or SONET). |
| Framing | For E3, this might be G.804; otherwise, no framing. |
| TX clocking | Clocking on the router. For E3 or SONET, this might be INTERNAL, meaning the AIP generates the clock. Otherwise, LINE indicates that the ATM switch provides the clocking. |
| input | Number of packets received and process switched. |
| output | Number of packets sent from process switch. |
| IN fast | Number of input packets fast-switched. |
| OUT fast | Number of output packets fast-switched. |
| Rate-Queue | List of configured rate queues. |
| reg= | Actual register value passed to the AIP to define a specific rate queue. |
| DYNAMIC | Indicates that the rate queue is dynamic and was created automatically by the software. Dynamic rate queues are created when an **atm pvc** command specifies a peak/average rate that does not match any user configured rate queue. The value PERMANENT indicates that the rate queue was user-configured. |
| VCCs | Number of virtual channel connections (VCCs) dynamically attached to this rate queue. |
| ATM4/0.1 | Indicates that the subinterface supports ATM adaptation layer AAL3/4 and displays the SMDS E.164 unicast address and the SMDS E.164 multicast address assigned to the subinterface. |
| Config. is | ACTIVE or VALID in *n* SECONDS. ACTIVE indicates that the current AIP configuration has been loaded into the AIP and is being used. There is a 5-second window when a user changes a configuration and the configuration is sent to the AIP. |

## Related Command

**atm pvc**

# show atm map

To display the list of all configured ATM static maps to remote hosts on an ATM network, use the **show atm map** privileged EXEC command.

**show atm map**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Privileged EXEC

## Sample Display

The following is sample output from the **show atm map** command:

```
Router# show atm map
Map list atm :

vines 3004B310:0001 maps to VC 4, broadcast
ip 131.108.168.110 maps to VC 1, broadcast
clns 47.0004.0001.0000.0c00.6e26.00 maps to VC 6, broadcast
appletalk 10.1 maps to VC 7, broadcast
decnet 10.1 maps to VC 2, broadcast
```

Table 7-2 describes the fields shown in the display.

**Table 7-2 Show ATM Map Field Descriptions**

| Field | Description |
|---|---|
| Map list | Name of map list. |
| *protocol address* maps to VC *x* | Name of protocol, the protocol address, and the VCD that the address is mapped to. |
| broadcast | Indicates pseudo broadcasting. |

## Related Commands

**atm pvc**
**map-list**

# show atm traffic

To display current, global ATM traffic information to and from all ATM networks connected to the router, use the **show atm traffic** privileged EXEC command.

**show atm traffic**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Privileged EXEC

## Sample Display

The following is sample output from the **show atm traffic** command:

```
Router# show atm traffic

4915 Input packets
0 Output packets
2913 Broadcast packets
0 Packets for non-existent VC
```

Table 7-3 describes the fields shown in the display.

**Table 7-3 Show ATM Traffic Field Descriptions**

| Field | Description |
|---|---|
| Input packets | Total packets input. |
| Output packets | Total packets output (non-broadcast). |
| Broadcast packets | Total broadcast packets output. |
| Packets for non-existent VC | Packets sent to virtual circuits not configured. |

## Related Command

**atm pvc**

# show atm vc

To display all active ATM virtual circuits (PVCs and SVCs) and traffic information, use the **show atm vc** privileged EXEC command.

**show atm vc** [*vcd*]

## Syntax Description

| | |
|---|---|
| *vcd* | (Optional) Specifies which virtual circuit to display information about. |

## Command Mode

Privileged EXEC

## Usage Guidelines

If no *vcd* is specified, the command displays information for all PVCs and SVCs. The output is in summary form (one line per virtual circuit).

## Sample Displays

The following is sample output from the **show atm vc** command when no *vcd* is specified, displaying statistics for all virtual circuits:

```
Router# show atm vc

Intfc.   VCD   VPI   VCI   Type  AAL/Encaps    Peak  Avg.  Burst
ATM4/0.1 1     1     1     PVC   AAL3/4-SMDS   0     0     0
ATM4/0   2     2     2     PVC   AAL5-SNAP     0     0     0
ATM4/0   3     3     3     PVC   AAL5-SNAP     0     0     0
ATM4/0   4     4     4     PVC   AAL5-MUX      0     0     0
ATM4/0   6     6     6     PVC   AAL5-SNAP     0     0     0
ATM4/0   7     7     7     PVC   AAL5-SNAP     0     0     0
```

The following is sample output from the **show atm vc** command when a *vcd is* specified, displaying statistics for that virtual circuit only:

```
Router# show atm vc 8

ATM4/0: VCD: 8, VPI: 8, VCI: 8, etype:0x0, AAL5 - LLC/SNAP, Flags: 0x30
PeakRate: 0, Average Rate: 0, Burst: 0 *32cells, VCmode: 0xE000
InPkts: 181061, OutPkts: 570499, InBytes: 757314267, OutBytes: 2137187609
InPRoc: 181011, OutPRoc: 10, Broadcasts: 570459
InFast: 39, OutFast: 36, InAS: 11, OutAS: 6
```

The following is sample output from the **show atm vc** command when a *vcd* is specified, AAL3/4 is enabled, an ATM SMDS subinterface has been defined, and a range of message identifier numbers (MIDs) has been assigned to the PVC:

```
Router# show atm vc 1

ATM4/0.1: VCD: 1, VPI: 0, VCI: 1, etype:0x1, AAL3/4 - SMDS, Flags: 0x35
PeakRate: 0, Average Rate: 0, Burst: 0 *32cells, VCmode: 0xE200
MID start: 1, MID end: 16
InPkts: 0, OutPkts: 0, InBytes: 0, OutBytes: 0
InPRoc: 0, OutPRoc: 0, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 0, OutAS: 0
```

Table 7-4 describes the fields shown in the displays.

**Table 7-4 Show ATM VC Field Descriptions**

| Field | Description |
|---|---|
| Intfc. | Interface slot/port. |
| VCD | Virtual circuit descriptor (virtual circuit number). |
| VPI | Virtual path identifier. |
| VCI | Virtual channel identifier. |
| Type | Type of virtual circuit, either PVC or SVC. |
| AAL/Encaps | Type of ATM adaptation layer (AAL) and encapsulation. |
| etype | Ether type. |
| Flags | Bit mask describing virtual circuit information. The flag values are summed to result in the displayed value.<br>0x40 SVC<br>0x20 PVC<br>0x10 ACTIVE<br>0x1 AAL5SNAP<br>0x2 AAL5NLPID<br>0x3 AAL5FRNLPID<br>0x4 AAL5MUX<br>0x5 AAL3/4-SMDS<br>0x6 QSAAL |
| PeakRate | Number of packets transmitted at the peak rate. |
| Average Rate | Number of packets transmitted at the average rate. |
| Burst | Value that, when multiplied by 32, equals the maximum number of ATM cells the virtual circuit can transmit at the peak rate of the virtual circuit. |
| Vcmode | AIP-specific register describing the usage of the virtual circuit. Contains values such as rate queue, peak rate, and AAL mode, which are also displayed in other fields. |
| InPkts | Total number of packets received on this virtual circuit. This number includes all silicon-switched, fast-switched, autonomous-switched, and process-switched packets. |
| OutPkts | Total number of packets sent on this virtual circuit. This number includes all silicon-switched, fast-switched, autonomous-switched, and process-switched packets. |
| InBytes | Total number of bytes received on this virtual circuit. This number includes all silicon-switched, fast-switched, autonomous-switched, and process-switched bytes. |
| OutBytes | Total number of bytes sent on this virtual circuit. This number includes all silicon-switched, fast-switched, autonomous-switched, and process-switched bytes. |
| InPRoc | Number of process-switched input packets. |
| OutPRoc | Number of process-switched output packets. |
| Broadcast | Number of process-switched broadcast packets. |

| Field | Description |
|---|---|
| InFast | Number of fast-switched input packets. |
| OutFast | Number of fast-switched output packets. |
| InAS | Number of autonomous-switched or silicon-switched input packets. |
| OutAS | Number of autonomous-switched or silicon-switched output packets. |

## Related Command

**atm pvc**

# show dxi map

To display all the protocol addresses mapped to a serial interface, use the **show dxi map** EXEC command.

**show dxi map**

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show dxi map** command. It displays output for several previously defined ATM-DXI maps that defined Apollo, IP, DECnet, CLNS, and AppleTalk protocol addresses, various encapsulations, and broadcast traffic.

```
Router# show dxi map

Serial0 (administratively down): ipx 123.0000.1234.1234
   DFA 69(0x45,0x1050), static, vpi = 4, vci = 5,
   encapsulation: SNAP
Serial0 (administratively down): appletalk 2000.5
   DFA 52(0x34,0xC40), static, vpi = 3, vci = 4,
   encapsulation: NLPID
Serial0 (administratively down): ip 131.108.177.1
   DFA 35(0x23,0x830), static,
   broadcast, vpi = 2, vci = 3,
   encapsulation: VC based MUX,
   Linktype IP
```

Table 7-5 explains significant fields shown in the display.

**Table 7-5 Show DXI Map Field Descriptions**

| Field | Description |
|---|---|
| DFA | DXI Frame Address, similar to a DLCI for Frame Relay. The DFA is shown in decimal, hexadecimal, and in DXI header format. The router computes this address value from the VPI and VCI values. |
| encapsulation: | Encapsulation type selected by the **dxi pvc** command. Displayed values can be SNAP, NLPID, or VC based MUX. |
| Linktype | Value used only with MUX encapsulation and therefore with only a single network protocol defined for the PVC. Maps configured on a PVC with MUX encapsulation must have the same link type. |

# show dxi pvc

To display the PVC statistics for a serial interface, use the **show dxi pvc** EXEC command.

**show dxi pvc**

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show dxi pvc** command. It displays output for ATM-DXI PVCs previously defined for serial interface 0.

```
Router# show dxi pvc

PVC Statistics for interface Serial0 (ATM DXI)

DFA = 17, VPI = 1, VCI = 1, PVC STATUS = STATIC, INTERFACE = Serial0

  input pkts 0             output pkts 0            in bytes 0
  out bytes 0              dropped pkts 0

DFA = 34, VPI = 2, VCI = 2, PVC STATUS = STATIC, INTERFACE = Serial0

  input pkts 0             output pkts 0            in bytes 0
  out bytes 0              dropped pkts 0

DFA = 35, VPI = 2, VCI = 3, PVC STATUS = STATIC, INTERFACE = Serial0

  input pkts 0             output pkts 0            in bytes 0
  out bytes 0              dropped pkts 0
```

Table 7-6 describes significant fields shown in the display.

**Table 7-6 Show DXI PVC Field Descriptions**

| Field | Description |
|---|---|
| DFA | DXI Frame Address, similar to a DLCI for Frame Relay. The DFA is shown in decimal, hexadecimal, and in DXI header format. The router computes this address value from the VPI and VCI values. |
| PVC STATUS = STATIC | Only static maps are supported. Maps are not created dynamically. |
| input pkts | Number of packets received. |
| output pkts | Number of packets transmitted. |
| in bytes | Number of bytes in all packets received. |
| out bytes | Number of bytes in all packets transmitted. |
| dropped pkts | Should display a zero (0) value. A nonzero value indicates a configuration problem, specifically that a PVC does not exist. |

# show sscop

To show SSCOP details for all ATM interfaces, use the **show sscop** privileged EXEC command.

**show sscop**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

Privileged EXEC

## Sample Display

The following is sample output from the **show sscop** command:

```
Router# show sscop
SSCOP details for interface ATM4/0
   Current State = Data Transfer Ready
   Send Sequence Number: Current = 2,  Maximum = 9
   Send Sequence Number Acked = 3
   Rcv Sequence Number: Lower Edge = 2, Upper Edge = 2, Max = 9
   Poll Sequence Number = 1876, Poll Ack Sequence Number = 2
   Vt(Pd) = 0
   Connection Control:  timer = 1000
   Timer currently Inactive
   Keep Alive Timer = 30000
   Current Retry Count = 0, Maximum Retry Count = 10
      Statistics -
      Pdu's Sent = 0, Pdu's Received = 0, Pdu's Ignored = 0
      Begin = 0/1, Begin Ack = 1/0, Begin Reject = 0/0
      End = 0/0, End Ack = 0/0
      Resync = 0/0, Resync Ack = 0/0
      Sequenced Data = 2/0, Sequenced Poll Data = 0/0
      Poll = 1591/1876, Stat = 0/1591, Unsolicited Stat = 0/0
      Unassured Data = 0/0, Mgmt Data = 0/0, Unknown Pdu's = 0
```

Table 7-7 describes the fields shown in the display. Interpreting this output requires a good understanding of the SSCOP; it is usually displayed by our technicians to help diagnose network problems.

**Table 7-7 Show SSCOP Field Descriptions**

| Field | Description |
|---|---|
| SSCOP details for interface | Interface slot and port. |
| Current State | SSCOP state for the interface. |
| Send Sequence Number | Current and maximum send sequence number. |
| Send Sequence Number Acked | Sequence number of packets already acknowledged. |
| Rcv Sequence Number | Sequence number of packets received. |
| Poll Sequence Number | Current poll sequence number. |
| Poll Ack Sequence Number | Poll sequence number already acknowledged. |

| Field | Description |
|---|---|
| Vt(Pd) | Number of Sd frames sent which triggers a sending of a Poll frame. |
| Connection Control | Timer used for establishing and terminating SSCOP. |
| Keep Alive Timer | Timer used to send keepalives on an idle link. |
| Current Retry Count | Current count of the retry counter. |
| Maximum Retry Count | Maximum value the retry counter can take. |
| Pdu’s Sent | Total number of SSCOP frames sent. |
| Pdu’s Received | Total number of SSCOP frames received. |
| Pdu’s Ignored | Number of invalid SSCOP frames ignored. |
| Begin | Number of Begin frames sent/received. |
| Begin Ack | Number of Begin Ack frames sent/received. |
| Begin Reject | Number of Begin Reject frames sent/received. |
| End | Number of End frames sent/received. |
| End Ack | Number of End Ack frames sent/received. |
| Resync | Number of Resync frames sent/received. |
| Resync Ack | Number of Resync Ack frames sent/received. |
| Sequenced Data | Number of Sequenced Data frames sent/received. |
| Sequenced Poll Data | Number of Sequenced Poll Data frames sent/received. |
| Poll | Number of Poll frames sent/received. |
| Stat | Number of Stat frames sent/received. |
| Unsolicited Stat | Number of Unsolicited Stat frames sent/received. |
| Unassured Data | Number of Unassured Data frames sent/received. |
| Mgmt Data | Number of Mgmt Data frames sent/received. |
| Unknown Pdu’s | Number of Unknown Pdu’s frames sent/received. |

# sscop cc-timer

To change the connection control timer, use the **sscop cc-timer** interface configuration command. The **no** form of this command restores the default value.

**sscop cc-timer** *seconds*
**no sscop cc-timer**

## Syntax Description

| | |
|---|---|
| *seconds* | Number of seconds between Begin messages. Default is 10 seconds. |

## Default

10 seconds

## Command Mode

Interface configuration

## Usage Guidelines

The connection control timer determines the time between transmission of BGN, END, or RS PDUs as long as an acknowledgment has not been received.

## Example

In the following example, the connection control timer is set to 15 seconds:

```
sscop cc-timer 15
```

## Related Command

**sscop max-cc**

# sscop keepalive-timer

To change the keepalive timer, use the **sscop keepalive-timer** interface configuration command. The **no** form of this command restores the default value.

**sscop keepalive-timer** *seconds*
**no sscop keepalive-timer** *seconds*

## Syntax Description

| | |
|---|---|
| *seconds* | Number of seconds the router waits between transmission of POLL PDUs when no SD or SDP PDUs are queued for transmission or are outstanding pending acknowledgments. |

## Default

30 seconds

## Command Mode

Interface configuration

## Example

In the following example, the keepalive timer is set to 15 seconds:

```
sscop keepalive-timer 15
```

# sscop max-cc

To change the retry count of connection control, use the **sscop max-cc** interface configuration command. The **no** form of this command restores the default value.

**sscop max-cc** *retries*
**no sscop max-cc**

## Syntax Description

| | |
|---|---|
| *retries* | Number of times that SSCOP will retry to transmit BGN, END, or RS PDUs as long as an acknowledgment has not been received. Valid range is 1 to 6000. |

## Default

10 retries

## Command Mode

Interface configuration

## Example

In the following example, the retry count of the connection control is set to 20:

```
sscop max-cc 20
```

## Related Command

**sscop cc-timer**

# sscop poll-timer

To change the poll timer, use the **sscop poll-timer** interface configuration command. The **no** form of this command restores the default value.

**sscop poll-timer** *seconds*
**no sscop poll-timer**

## Syntax Description

| | |
|---|---|
| *seconds* | Number of seconds the router waits between transmission of POLL PDUs. |

## Default

10 seconds

## Command Mode

Interface configuration

## Usage Guidelines

The poll timer controls the maximum time between transmission of POLL PDUs when SD or SDP PDUs are queued for transmission or are outstanding pending acknowledgments.

## Example

In the following example, the poll timer is set to 15 seconds:

```
sscop poll-timer 15
```

# sscop rcv-window

To change the receiver window, use the **sscop rcv-window** interface configuration command. The **no** form of this command restores the default value.

**sscop rcv-window** *packets*
**no sscop rcv-window**

## Syntax Description

| | |
|---|---|
| *packets* | Number of packets the interface can receive before it must send an acknowledgment to the ATM switch. Valid range is 1 to 6000. |

## Default

7 packets

## Command Mode

Interface configuration

## Example

In the following example, the receiver's window is set to 10 packets:

```
sscop rcv-window 10
```

# sscop send-window

To change the transmitter window, use the **sscop send-window** interface configuration command. The **no** form of this command restores the default value.

**sscop send-window** *packets*
**no sscop send-window**

## Syntax Description

| | |
|---|---|
| *packets* | Number of packets the interface can send before it must receive an acknowledgment from the ATM switch. Valid range is 1 to 6000. |

## Default

7 packets

## Command Mode

Interface configuration

## Example

In the following example, the transmitter's window is set to 10 packets:

```
sscop send-window 10
```

CHAPTER 8

# DDR Commands

This chapter lists dial-on-demand routing (DDR) commands, explains the command syntax, and provides usage guidelines. For information about configuring DDR and configuration examples, refer to the “Configuring DDR” chapter in the *Router Products Configuration Guide.*

# backup delay

To define how much time should elapse before a secondary line status changes after a primary line status has changed, use the **backup delay** interface configuration command. To return to the default, which means as soon as the primary fails, the secondary is immediately brought up without delay, use the **no** form of this command.

**backup delay** {*enable-delay* | **never**} {*disable-delay* | **never**}
**no backup delay** {*enable-delay* | **never**} {*disable-delay* | **never**}

## Syntax Description

| | |
|---|---|
| *enable-delay* | Number of seconds that elapse after the primary line goes down before the router activates the secondary line. |
| *disable-delay* | Number of seconds that elapse after the primary line goes up before the router deactivates the secondary line. |
| **never** | Prevents the secondary line from being activated or deactivated. |

## Default

0 seconds

## Command Mode

Interface configuration

## Usage Guidelines

For environments in which there are spurious signal disruptions that may appear as intermittent lost carrier signals, it is recommended that some delay be enabled before activating and deactivating a secondary line.

## Example

The following example sets a 10-second delay on deactivating the secondary line (interface serial 0); however, the line is activated immediately:

```
interface serial 0
backup delay 0 10
```

# backup interface

To configure the serial interface as a secondary or dial backup line, use the **backup interface** interface configuration command. To disable this feature, use the **no** form of this command.

**backup interface** *type number*
**no backup interface** *type number*

## Syntax Description

| | |
|---|---|
| *type* | Interface type. It must be **serial**. |
| *number* | Serial port to be set as the secondary line. |

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

The interface you define with this command can only backup one interface.

## Example

The following example sets serial 1 as the backup line:

```
interface serial 1
backup interface serial 1
```

# backup load

To set traffic load threshold for dial backup service, use the **backup load** interface configuration command. To return to the default value, use the **no** form of this command.

**backup load** {*enable-load* | **never**} {*disable-load* | **never**}
**no backup load** {*enable-load* | **never**} {*disable-load* | **never**}

## Syntax Description

| | |
|---|---|
| *enable-load* | Percentage of the primary line's available bandwidth. |
| *disable-load* | Percentage of the primary line's available bandwidth. |
| **never** | Sets the secondary line to never be activated due to traffic load. |

## Default

No threshold is predefined.

## Command Mode

Interface configuration

## Usage Guidelines

When the transmitted or received load on the primary line is greater than the value assigned to the *enable-load* argument, the secondary line is enabled.

The secondary line is disabled when one of the following conditions occur:

- The transmitted load on the primary line plus the transmitted load on the secondary line is less than the value entered for the *disable-load* argument.
- The received load on the primary line plus the received load on the secondary line is less than the value entered for the *disable-load* argument.

If the **never** keyword is used instead of an enable-threshold value, the secondary line is never activated because of a traffic load. If the **never** keyword is used instead of a *disable-load* argument, the secondary line is never activated because of traffic load.

## Example

The following example sets the traffic load threshold to 60 percent of the primary line serial 0. When that load is exceeded, the secondary line is activated, and will not be deactivated until the combined load is less than 5 percent of the primary bandwidth.

```
interface serial 0
backup load 60 5
```

# chat-script

To create a script that will place a call over a modem, use the **chat-script** global configuration command. To disable the specified chat script, use the **no** form of this command.

**chat-script** *script-name expect-send*
**no chat-script** *script-name expect-send*

## Syntax Description

| | |
|---|---|
| *script-name* | Name of the chat script. |
| *expect-send* | Content of the chat script. |

## Default

No chat scripts are defined.

## Command Mode

Global configuration

## Usage Guidelines

Chat scripts are used in dial-on-demand routing to give commands to dial a modem and commands to log on to remote systems. The defined script will be used to place a call over a modem.

Some characteristics of chat scripts are as follows:

- Chat scripts are case sensitive.
- You can have any number of ABORT sequences active at once.
- When a chat script starts, the default timeout is 5 seconds. Changes to the timeout persist until the next time you change them in the script.
- A string within quotation marks is treated as a single entity.

It is recommended that one chat script (a “modem” chat script) be written for placing a call and another chat script (a “system” or “login” chat script) be written to log onto remote systems, where required.

## Suggested Chat Script Naming Conventions

A suggested chat script naming convention is as follows:

*vendor-type-modulation*

In other words, the syntax of the **chat-script** command becomes the following:

**chat-script** *vendor-type-modulation expect send*

For example, if you have a Telebit T3000 modem that uses V.32bis modulation, you would name your chat script as follows:

```
telebit-t3000-v32bis
```

For example, the chat-script command could become the following:

```
chat-script telebit-t3000-v32bis ABORT ERROR ABORT BUSY ABORT "NO ANSWER" "" "AT H" OK
"AT DT \T" DIALING \c TIMEOUT 30 CONNECT \c
```

For example, you could have script names like the following

- telebit-tb-b103
- telebit-tb-v21
- telebit-tb-v22
- codex-326x-b103
- codex-326x-v21
- codex-326x-v22
- codex-326x-v22bis
- codex-326x-v32
- codex-326x-v32bis
- usr-courier-v22bis
- usr-courier-hst
- usr-courier-v32
- usr-courier-v32bis

Adhering to this naming convention allows you to use partial chat script names with regular expressions to specify a range of chat scripts that can be used. This is particularly useful for dialer rotary groups and is explained further in the next section.

## Escape Sequences

Chat scripts are in the form *expect send*, where the send string following the hyphen is executed if the preceding expect string fails. Each send string is followed by a return unless it ends with \c. ^*x* gets translated into the appropriate control character, and *x* gets translated into x if *x* is not one of the special sequences listed in Table 8-1.

See the book entitled *Managing uucp and Usenet* by Tim O'Reilly and Grace Todino for more information about chat scripts.

The escape sequences used in chat scripts are listed in Table 8-1.

**Table 8-1 Chat Script Escape Sequences**

| Escape Sequence | Description |
|---|---|
| “” | Expect a null string. |
| EOT | Send an end-of-transmission character. |
| BREAK | Cause a BREAK. This is sometimes simulated using line speed changes and null characters. May not work on all systems. |
| \c | Suppress new line at the end of the send string. |
| \d | Delay for 2 seconds. |
| \K | Insert a BREAK. |

| Escape Sequence | Description |
|---|---|
| \n | Send a newline or linefeed character. |
| \p | Pause for 1/4 second |
| \r | Send a return. |
| \s | Send a space character. |
| \t | Send a table character |
| \\ | Send a backslash (\) character. |
| \T | Replaced by phone number. |
| \q | Reserved, not yet used |

## Expect-Send Pairs

Sample supported *expect-send* pairs are described in Table 8-2.

**Table 8-2 Sample Supported Expect-Send Pairs**

| Expect and Send Pair | Function |
|---|---|
| **ABORT** *string* | Starts scanning for the string in the input and if it is seen this indicates that the chat script has failed. |
| **TIMEOUT** *time* | Sets the time to wait for input, in seconds. The default is five seconds. |

As an example of how expect-send pairs function, if the modem reports BUSY when the number is busy, you can indicate that you want the attempt stopped at this point by including ABORT BUSY in your chat script.

## Alternate Handlers

ABORT sink instead of ABORT ERROR means that the system will abort when it sees sink instead of when it sees ERROR.

## Missed Characters

After the connection is established and Return is pressed, a second Return is often required before the prompt appears.

You might include the following as part of your chat script:

```
ssword:-/r-ssword
```

This means that after the connection is established you want “ssword” to be displayed. If it is not displayed, send a return again after the timeout passes.

## Example

The following example shows the **chat-script** command being used to create a chat script named t3000:

```
chat-script t3000 ABORT ERROR ABORT BUSY ABORT "NO ANSWER" "" "AT H" OK "AT DT \T" DIALING
\c TIMEOUT 30 CONNECT \c
```

## Related Commands

**dialer map**
**script dialer**

# clear dialer

To clear the values of dialer statistics for one or more serial or BRI interfaces configured for DDR, use the **clear dialer** privileged EXEC command.

**clear dialer** [**interface** *type number*]

**clear dialer** [**interface serial** *slot*/*port*]    (Cisco 7000 series only)

## Syntax Description

| | |
|---|---|
| **interface** | (Optional) Indicates that one interface will be specified. |
| *type* | Interface type, either **serial** or **bri**. |
| *number* | Interface number. |
| *slot*/*port* | On the Cisco 7000 series, specifies the slot and port numbers. |

## Command Mode

Privileged EXEC

## Usage Guidelines

If the **interface** keyword and the arguments are not used, dialer statistics are cleared on all interfaces.

## Example

The following example clears the dialer statistics on serial interface 1:

```
clear dialer interface serial 1
```

# clear snapshot quiet-time

To end the quiet period on a client router within two minutes, use the **clear snapshot quiet-time** EXEC command.

**clear snapshot quiet-time** *interface*

## Syntax Description

| | |
|---|---|
| *interface* | Interface type and number. |

## Command Mode

EXEC

## Usage Guidelines

The **clear snapshot quiet-time** command places the client router in a state to reenter the active period within two minutes. The two-minute hold period ensures a quiet period of at least two minutes between active periods.

## Example

The following example ends the quiet period on dialer interface 1:

```
clear snapshot quiet-time dialer 1
```

## Related Commands

**show snapshot**
**snapshot client**

# dialer caller

To configure caller ID screening, use the **dialer caller** interface configuration command. To disable this feature, use the **no** form of this command.

**dialer caller** *number*
**no dialer caller** *number*

## Syntax Description

| | |
|---|---|
| *number* | Telephone number for which to screen. Specify an x to represent a single "don't-care" character. The maximum length of each number is 25 characters. |

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

This command configures the router to accept calls from the specified number.

Caller ID screening is available on Cisco 7000 series, Cisco 4000 series, Cisco 3000 series, and Cisco 2500 series routers that have dialer interfaces.

The maximum length of each number is 25 characters.

**Note** Caller ID screening requires a local switch that is capable of delivering the caller ID to the router. If you enable caller ID screening but do not have such a switch, no calls will be allowed in.

## Examples

The following example configures the router to accept a call with a delivered caller ID equal to 4155551234:

```
dialer caller 4155551234
```

The following example configures the router to accept a call with a delivered caller ID having 41555512 and any numbers in the last two positions:

```
dialer caller 41555512xx
```

## Related Command

**show dialer**

# dialer dtr

To enable DDR on an interface and specify that the serial line is connected by non-V.25bis modems using EIA signaling only (the data terminal ready [DTR] signal), use the **dialer dtr** interface configuration command. To disable dial-on-demand routing for the interface, use the **no** form of this command.

**dialer dtr**
**no dialer dtr**

## Syntax Description

This command has no keywords or arguments.

## Default

DTR dialing is disabled.

## Command Mode

Interface configuration

## Usage Guidelines

A serial interface configured for DTR dialing can place calls only; it cannot accept them.

When a local interface is configured for DTR dialing, the remote interface (that will be receiving the calls) can be configured for in-band dialing or not configured for anything but encapsulation, depending on the desired behavior. If the remote interface is expected to terminate a call when no traffic is transmitted for some time, it must be configured for in-band dialing (along with access lists and a dummy dialer string). If the remote interface is purely passive, no configuration is necessary.

Rotary groups cannot be configured for DTR dialing.

The **dialer map** and **dialer string** commands have no effect on DTR dialers.

## Example

The following example enables DDR and specifies DTR dialing on an interface:

```
dialer dtr
```

## Related Commands

**dialer in-band**
**dialer map**
**dialer string**

# dialer enable-timeout

To set the length of time an interface stays down after a call has completed or failed, before it is available to dial again, use the **dialer enable-timeout** interface configuration command. To return to the default value, use the **no** form of this command.

**dialer enable-timeout** *seconds*
**no dialer enable-timeout**

## Syntax Description

| | |
|---|---|
| *seconds* | Time in seconds that the router waits before the next call can occur on the specific interface. Acceptable values are positive, nonzero integers. |

## Default

15 seconds

## Command Mode

Interface configuration

## Usage Guidelines

This command applies to inbound and outbound calls.

If your phone lines are busy or down, you might want to enforce a certain period of time before the system repeats an attempt to make a connection with a remote site. Configuring this timeout can prevent outgoing lines and switching equipment from being needlessly loaded down.

## Example

The following example specifies a waiting period of 30 seconds on interface async 1:

```
interface async 1
dialer enable-timeout 30
```

# dialer fast-idle

To specify the amount of time that a line for which there is contention will stay idle before the line is disconnected and the competing call is placed, use the **dialer fast-idle** interface configuration command. To return to the default value, use the **no** form of this command.

**dialer fast-idle** *seconds*
**no dialer fast-idle**

## Syntax Description

| | |
|---|---|
| *seconds* | Idle time, in seconds, that must occur on an interface before the line is disconnected. Acceptable values are positive, nonzero integers. |

## Default

20 seconds

## Command Mode

Interface configuration

## Usage Guidelines

The fast idle timer is activated if there is contention for a line. In other words, if a line is busy, a packet for a different next hop address is received, and the busy line is required to send the competing packet, the dialer fast idle timer is activated.

If the line becomes idle for configured length of time, the current call is disconnected immediately and the new call is placed.

If the line has not yet been idle as long as the fast idle timer, the packet is dropped because there is no way to get through to the destination. After the packet is dropped, the fast idle timer remains active and the current call is disconnected as soon as it has been idle for as long as the fast idle timeout.

If, in the meanwhile, there is another packet transmitted to the currently connected destination, and it is classified as interesting, the fast idle timer will be restarted.

This command applies to inbound and outbound calls.

Combining this command with the **dialer idle-timeout** command allows you to configure lines to stay up for a longer period of time when there is not contention, but to be reused more quickly when there are not enough lines for the current demand.

## Example

The following example specifies a fast idle timeout of 35 seconds on interface async 1:

```
interface async 1
dialer fast-idle 35
```

## Related Commands

**dialer idle-timeout**
**dialer map**

# dialer hold-queue

To allow "interesting" outgoing packets to be queued until a modem connection is established, use the **dialer hold-queue** interface configuration command.

**dialer hold-queue** *packets*
**no dialer hold-queue** [*packets*]

## Syntax Description

| | |
|---|---|
| *packets* | Number of packets, in the range 0 to 100 packets, to hold in the queue. This argument is optional with the **no** form of the command. |

## Default

The outgoing packet queue is disabled.

## Command Mode

Interface configuration

## Usage Guidelines

A dialer hold queue can be configured on any type of dialer, including in-band synchronous, asynchronous, DTR, and ISDN dialers. Rotary groups can be configured with a dialer hold queue. If a rotary group is configured with a hold queue, all members of the group will be configured with a dialer hold queue and no individual member's hold queue can be altered.

## Example

The following command configures a dialer hold queue to hold 10 packets:

```
dialer hold-queue 10
```

## Related Command

**dialer-group**

# dialer idle-timeout

To specify the idle time before the line is disconnected, use the **dialer idle-timeout** interface configuration command. To reset the idle timeout to the default, use the **no** form of this command.

**dialer idle-timeout** *seconds*
**no dialer idle-timeout**

## Syntax Description

| | |
|---|---|
| *seconds* | Idle time, in seconds, that must occur on an interface before the line is disconnected. Acceptable values are positive, nonzero integers. |

## Default

120 seconds

## Command Mode

Interface configuration

## Usage Guidelines

This command is used on lines for which there is no contention. When contention occurs, the dialer fast-idle command is activated. For example, when a busy line is requested to send another packet to a different destination than it is currently connected to, line contention occurs and the **dialer fast-idle** command is activated.

This command applies to inbound and outbound calls. For example, if a receiving system needs to make outgoing calls, you might configure it with a short idle timeout.

## Example

The following example specifies of an idle timeout of 3 minutes (180 seconds) on interface async 1:

```
interface async 1
dialer idle-timeout 180
```

## Related Command

**dialer fast-idle**

# dialer in-band

To specify that DDR is to be supported, use the **dialer in-band** interface configuration command. To disable dial-on-demand routing for the interface, use the **no** form of this command.

**dialer in-band** [**no-parity** | **odd-parity**]
**no dialer in-band**

## Syntax Description

| | |
|---|---|
| **no-parity** | (Optional) Indicates that no parity is to be applied to the dialer string that is sent out to the modem on synchronous interfaces. |
| **odd-parity** | (Optional) Indicates that the dialed number has odd parity (7-bit ASCII characters with the eighth bit the parity bit) on synchronous interfaces. |

## Default

Disabled. By default, no parity is applied to the dialer string.

## Command Mode

Interface configuration

## Usage Guidelines

The **dialer in-band** command specifies that chat scripts will be used on the auxiliary port and V.25bis will be used on synchronous interfaces.

The parity keywords do not apply to asynchronous interfaces.

The parity setting applies to the dialer string that is sent out to the modem. If you do not specify a parity, or if you specify no parity, no parity is applied to the output number. If odd parity is configured, the dialed number will have odd parity (7-bit ASCII characters with the eighth bit, the parity bit.)

If an interface is only accepts calls and does not place calls, the **dialer in-band** interface configuration command is the only command needed to configure it. If an interface is configured in this manner, with no dialer rotary groups, the idle timer never disconnects the line. It is up to the remote end (the end that placed the call) to disconnect the line based on idle time.

## Example

The following example specifies DDR for asynchronous interface 1:

```
interface async 1
dialer in-band
```

## Related Commands

**dialer map**
**dialer string**

# dialer load-threshold

To configure bandwidth on demand by setting the maximum load before the dialer places another call to a destination, use the **dialer load-threshold** interface command. To disable the setting, use the **no form** of this command.

**dialer load-threshold** *load*
**no dialer load-threshold**

## Syntax Description

| | |
|---|---|
| *load* | Interface load beyond which the dialer will initiate another call to the destination. This argument is a number between 1 and 255. |

## Default

No maximum load is predefined.

## Command Mode

Interface configuration

## Usage Guidelines

This command applies to dialer rotary groups only.

If a packet is transmitted on a dialer interface, there is a call established, and the transmit load on the interface exceeds the specified load threshold, the dialer will initiate another call to the destination. The dialer will make additional calls as necessary to expand bandwidth but will never interrupt an existing call to another destination.

The argument *load* is the calculated weighted average load value for the interface; 1 is unloaded, 255 is fully loaded. The load is calculated by the system dynamically, based on bandwidth. You must set the bandwidth for an interface in kilobits per second, using the **bandwidth** command.

The load calculation determines how much of the total bandwidth you are using, where 255 means that you are using one hundred percent of the bandwidth.

See the interface configuration chapter for a full description of the **bandwidth** command.

## Example

In the following example, if the load to a particular destination on an interface in dialer rotary group 5 exceeds interface load 200, the dialer will initiate another call to the destination.

```
interface dialer 5
dialer load-threshold 200
```

## Related Commands

A dagger (**†**) indicates that the command is documented in another chapter.

**bandwidth †**
**interface dialer**
**dialer rotary-group**

# dialer map

To configure a serial interface or Integrated Services Digital Network (ISDN) interface to call one or multiple sites, use a form of the **dialer map** interface configuration command; all options are shown in the first form of the command. To configure a serial interface or ISDN interface to place a call to multiple sites and to authenticate calls from multiple sites, use the second form of the **dialer map** command. To configure a serial interface or ISDN interface to support bridging, use the third form of the command. To configure an asynchronous interface to place a call to a single site that has no modem script assigned or that requires a system script, or to multiple sites on a single line, on multiple lines, or on a dialer rotary group, use the fourth form of the **dialer map** command.To delete a particular dialer map entry, use a **no** form of this command.

**dialer map** *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [**modem-script** *modem-regexp*] [**system-script** *system-regexp*] [*dial-string*[**:***isdn-subaddress*]]
**no dialer map** *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [**modem-script** *modem-regexp*] [**system-script** *system-regexp*] [*dial-string*[**:***isdn-subaddress*]]

**dialer map** *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [*dial-string*[**:***isdn-subaddress*]]
**no dialer map** *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [*dial-string*[**:***isdn-subaddress*]]

**dialer map bridge** [**name** *hostname*] [**spc**] [**broadcast**] [*dial-string*[**:***isdn-subaddress*]]
**no dialer map bridge** [**name** *hostname*] [**spc**] [**broadcast**] [*dial-string*[**:***isdn-subaddress*]]

**dialer map** *protocol next-hop-address* [**name** *hostname*] [**broadcast**] [**modem-script** *modem-regexp*] [**system-script** *system-regexp*] [*dial-string*]
**no dialer map** *protocol next-hop-address* [**name** *hostname*] [**broadcast**] [**modem-script** *modem-regexp*] [**system-script** *system-regexp*] [*dial-string*]

## Syntax DescriptionDefault

| | |
|---|---|
| *protocol* | Protocol keyword. See Table 8-3 for a list of supported protocols and their keywords. |
| *next-hop-address* | Protocol address used to match against addresses to which packets are destined. This argument is not used with the **bridge** protocol keyword. |
| **name** | (Optional) Indicates the remote system with which the local router communicates. |
| *hostname* | (Optional) Case-sensitive name or ID of the remote device (usually the host name). For routers with ISDN interfaces, if calling line identification (CLI/ANI/caller ID) is provided, the *hostname* field can contain the number that the calling line ID provides. |
| **spc** | Specifies a semipermanent connection between customer equipment and the exchange; used only in Germany to configure connections between an ISDN BRI and a 1TR6 ISDN switch type. |

| | |
|---|---|
| **speed 56 \| 64** | Keyword and value indicating the line speed to use. Used for ISDN only. |
| **broadcast** | Indicates that broadcasts should be forwarded to this protocol address. |
| **modem-script** | (Optional) Indicates the modem script to be used for the connection (for asynchronous interfaces). |
| *modem-regexp* | (Optional) Regular expression to which a modem script will be matched (for asynchronous interfaces). |
| **system-script** | (Optional) Indicates the system script to be used for the connection (for asynchronous interfaces). |
| *system-regexp* | (Optional) Regular expression to which a system script will be matched (for asynchronous interfaces). |
| *dial-string* | Telephone number sent to the dialing device when it recognizes packets with the specified next-hop-address that matches the access lists defined.<br>*The dial string must be the last item in the command line.* |
| **:***isdn-subaddress* | (Optional) Subaddress number used for ISDN multipoint connections. |

No dialer map is configured. The default speed is 64. No scripts are defined for placing calls.

## Command Mode

Interface configuration

## Usage Guidelines

Table 8-3 lists the protocols supported by the **dialer map** command.

**Table 8-3 Dialer Map Command Supported Protocols**

| Keyword | Protocol |
|---|---|
| **appletalk** | AppleTalk |
| **bridge** | Bridging |
| **clns** | ISO CLNS |
| **decnet** | DECnet |
| **ip** | IP |
| **ipx** | Novell IPX |
| **novell** | Novell IPX |
| **snapshot** | Snapshot Routing |
| **vines** | Banyan VINES |
| **xns** | Xerox Network Services |

## Synchronous and ISDN Interfaces

Use the **dialer map** command with the **name** keyword in configurations in which remote sites are calling a central site, but the central site is not calling the remote site. With this command, the local device will authenticate the remote site using CHAP or PAP, which will transmit the remote site's host name to the central site. The central site will then use this name to authenticate the caller, and will use the next hop address to transmit packets to the remote site. Because there is no dialer string specified, the central site cannot call the remote router.

For ISDN interfaces only, you can specify an optional speed parameter for **dialer map** commands if you also specify a dial string. This option informs the ISDN software whether it should place a call at 56 or 64 kbps. If you omit the ISDN speed parameter, the default is 64 kbps.

For routers with ISDN interfaces, if calling line identification (CLI/ANI/caller ID) is provided, the *hostname* field may contain the number that calling line id provides.

## Asynchronous Interfaces

Specify chat scripts for a physical interface that is not part of a dialer rotary group if no chat script is specified for the line or an additional (system) chat script is required to log on to the remote system.

Configure a **dialer map** command for each remote destination for that interface.

You do not need to specify a system script under the following conditions:

- The modem script can be used to dial and log on to the remote system.
- You are calling a system that does not require a login script; that is, a system that answers and immediately goes into protocol mode.

If you adhere to the chat script naming convention suggested in this publication, use the form [**modem-script** **modulation-type*] in the **dialer map** command; for example, ".*-v32bis." This allows you to specify the modulation type that is best for the system you are calling, and allows the modem type for the line to be specified by the **modem chat-script** command.

The period (.) is a wildcard that matches any character, and the asterisk (*) indicates that the preceding character can be duplicated multiple times. For more information about regular expressions, see the "Regular Expressions" appendix.

If there is a **modem-script** specified in the **dialer map** interface configuration command and a modem script specified in the **modem chat-script** line configuration command, the first chat script that matches both will be used. If no script matches both, an error message is logged and the connection is not established. If there is no modem chat script specified for the line, the first chat script (that is, the one specified using the **chat-script** global configuration command) that matches the modem script regular expression will be used. If there is a system script specified in the **dialer map** interface configuration command, the first chat script to match the regular expression will be used.

The **modem-script** and **system-script** keywords and corresponding arguments are optional. They are ignored on synchronous interfaces.

If you have named your chat script according to the type of modem and modulation (for example, codex-v32 or telebit v32), your regular expression could be codex-.* in the **modem chat-script** line configuration command, and *-v32bis in the modem script specified in the **dialer map** command for a system that you wish to connect to using v32bis modulation.

The modem lines (specified by the argument *regexp* in the **modem chat-script** line configuration command) would be set to one of the following regular expressions to match patterns, depending on what kind of modem you have:

- codex-.*
- telebit-.*
- usr-.*

With an interface configured for Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) and configured with the **name** *hostname* keyword and argument pair, the local device authenticates the remote site using CHAP, which transmits the remote site's host name to the central site. The central site then uses this name to authenticate the caller and uses the next hop address to transmit packets to the remote site. Because no dialer string is specified, the central site cannot call the remote router.

For routers with ISDN interfaces, if calling line identification (CLI/ANI/caller id) is provided, the *hostname* field can contain the number that calling line id provides.

## Examples

In the following example, the dialer speed is set at 56 kbps to call a remote site at 131.108.2.5.

```
interface async 1
encapsulation ppp
ppp authentication chap
dialer map ip 131.108.2.5 speed 56
```

The following example shows a dialing chat script and a login chat script. The **dialer in-band** command enables DDR on asynchronous interface 10 and the **dialer map** command looks for the specified dialing and the login scripts, and then uses those scripts to dial 96837890.

```
chat-script dial ABORT ERROR "" "AT Z" OK "ATDT \T" TIMEOUT 30 CONNECT \c
chat-script login ABORT invalid TIMEOUT 15 name: billw word: wewpass ">"
                "slip default"
interface async 10
dialer in-band
dialer map ip 10.55.0.1 modem-script dial system-script login 96837890
```

In the following example, the remote site is calling the central site, and the central site is calling the remote site. The central router can use the name, ZZZ, to authenticate the remote router when they connect and also can use the dialer string 14155553434 to call the remote router if it is not currently connected.

```
interface async 1
dialer map ip 131.108.2.5 name ZZZ 14155553434
```

In the following example, a remote site is calling a central site, but the central site is not calling the remote site. The local device will authenticate the site that is calling in using CHAP. CHAP will cause the remote site's name, YYY, to be transmitted to the site it is calling. The central site will then use this name to authenticate the remote site.

```
interface async 1
encapsulation ppp
ppp authentication chap
dialer map ip 131.108.2.5 name YYY
```

## Related Commands

A dagger (**†**) indicates that the command is documented in another chapter.

**chat-script**
**ppp authentication chap** †
**ppp authentication pap** †
**username**

# dialer map snapshot

To define a dialer map for Cisco's snapshot routing protocol on a client router connected to a DDR interface, use the **dialer map snapshot** interface configuration command. To delete one or more previously defined snapshot routing dialer maps, use the **no** form of this command.

**dialer map snapshot** *sequence-number dial-string*
**no dialer map snapshot** [*sequence-number*]

## Syntax Description

| | |
|---|---|
| *sequence-number* | An number in the range from 1 to 254, inclusive, that uniquely identifies a dialer map. |
| *dial-string* | Telephone number of a remote snapshot server to be called during an active period. |

## Default

No snapshot routing dialer map is defined.

## Command Mode

Interface configuration

## Usage Guidelines

Enter a command for each remote snapshot server router the client router should call during an active period.

Use the **no dialer map snapshot** form of this command to remove all previously defined snapshot dialer maps on the client router; use the **no dialer map snapshot** *sequence-number* form of this command to delete a specified dialer map.

## Example

The following examples define snapshot dialer maps on a client router:

```
dialer map snapshot 12 4151231234
dialer map snapshot 13 4151231245
```

The following example removes one of the previously defined snapshot routing dialer maps on the client router:

```
no dialer map snapshot 13
```

## Related Commands

**dialer rotary-group**
**interface dialer**
**snapshot client**

# dialer priority

To set the priority of an interface in a dialer rotary group, use the **dialer priority** interface configuration command. Use the **no** form of the command to revert to the default setting.

**dialer priority** *number*
**no dialer priority**

## Syntax Description

| | |
|---|---|
| *number* | Priority of an interface in a dialer rotary group; the highest number indicates the highest priority. This is a number from 0 through 255. The default value is 0. |

## Default

No priority is predefined. When priority is defined, the default value is 0.

## Command Mode

Interface configuration

## Usage Guidelines

The value 0 indicates the lowest priority and 255 indicates the highest priority. The **dialer priority** command controls which interfaces within a dialer rotary group will be used first. Higher priority interfaces (configured with higher *n* value) are used first. This command is only meaningful for interfaces that are part of dialer rotary groups.

The **priority** command gives the administrator the ability to tell the dialer rotary group which free interface (and by extension which modem) to use first. This command applies to outgoing calls only.

## Examples

In the following example, interface async 3 will be used after interfaces with higher and before interfaces with lower priority.

```
interface async 3
dialer priority 5
```

For example, a router has a selection of many modems on it. Some of them are perceived to be better performers than others. You also have two 4800-bps, three 1200-bps, and one 300-bps modem. They are all on interfaces that are in a dialer rotary group. You do not want the router to make the call on the 300-baud modem if any of the faster modems are free. You want the router to use the highest-performance modems first, and the slowest modems last.

## Related Commands

**interface dialer**
**dialer rotary-group**

# dialer rotary-group

To include an interface in a dialer rotary group, use the **dialer rotary-group** interface configuration command.

**dialer rotary-group** *number*

## Syntax Description

| | |
|---|---|
| *number* | Number of the previously defined dialer interface in whose rotary group this interface is to be included. A number from 0 to 255. The dialer interface is defined by the **interface dialer** command. |

## Default

No interfaces are included in a dialer rotary group.

## Command Mode

Interface configuration

## Example

The following example places async interfaces1 and 2 into dialer rotary group 1, defined by the **interface dialer 1** command:

```
hostname central-site
! PPP encapsulation is enabled for interface dialer 1.
interface dialer 1
encapsulation ppp
dialer in-band
ip address 131.108.2.1 255.255.255.0
ip address 131.126.4.1 255.255.255.0 secondary

! The first dialer map command allows the central site and remote site YYY
! and to call each other and allows the central site to authenticate site YYY
! when it calls in. The second dialer map command, with no! dialer string,
! allows the central site to authenticate remote site ZZZ when it calls in, but
! the central site cannot call remote site ZZZ (no phone number).
dialer map ip 131.108.2.5 name YYY 14155553434
dialer map ip 131.126.4.5 name ZZZ

! The DTR pulse signals for three seconds on the interfaces in dialer
! group 1. This holds the DTR low so the modem can recognize that DTR has been
! dropped.
pulse-time 3

! Interfaces async 1 and async 2 are placed in dialer rotary group 1.
! All of the interface configuration commands (the encapsulation and dialer
! map commands shown earlier in this example) applied to interface
! dialer 1 apply to the physical interfaces assigned to the dialer group.
interface async 1
dialer rotary-group 1
interface async 2
dialer rotary-group 1
```

## Related Command

**interface dialer**

# dialer string

To specify the string (telephone number) to be called for interfaces calling a single site, use the **dialer string** interface configuration command. To delete the dialer string specified for the interface, use the **no** form of this command.

**dialer string** *dial-string*
**no dialer string**

## Syntax Description

| | |
|---|---|
| *dial-string* | String of characters to be sent to a DCE. |

## Default

No strings are predefined.

## Command Mode

Interface configuration

## Usage Guidelines

To use this command on an asynchronous interface, a modem chat script must be defined for the associated line, by using the **script dialer** command. A script must be used to implement dialing.

Dialers configured as **in-band** pass the string to the external dialing device. Specify one **dialer string** command per interface.

To specify multiple strings, use the **dialer map** command. In general, you include a **dialer string** or **dialer map** command if you intend to use a specific interface to initiate a DDR call.

---

**Note** If a **dialer string** command is specified without a **dialer-group** command with access lists defined, dialing never will be initiated. If debug dialer is enabled, an error message will be displayed indicating that dialing never will occur.

---

The string of characters specified for the *dial-string* argument is the default number used under the following conditions:

- A **dialer map** command is not included in the interface configuration.
- The next-hop-address specified in a packet is not included in any of the **dialer map** interface configuration commands recorded—assuming that the destination address passes any access lists specified for DDR with the **dialer-list** command.

## ITU-T V.25bis Options

On synchronous interfaces, depending on the type of modem you are using, International Telecommunication Union Telecommunication (ITU-T) Standardization Sector V.25bis options might be supported as *dial-string* parameters of the **dialer string** command. Supported options are listed in Table 8-4. The functions of the parameters are nation specific, and they may have different implementations in your country. These options apply only if you have enabled DDR with the **dialer in-band** command. Refer to the operation manual for your modem for a list of supported options.

**Table 8-4 ITU-TV.25bis Options**

| Option | Description |
|---|---|
| **:** | Wait tone. |
| < | Pause.<br>Usage and duration of this parameter vary by country. |
| = | Separator 3.<br>For national use. |
| > | Separator 4<br>For national use. |
| **P** | Dialing to be continued in pulse mode.<br>Optionally accepted parameter. |
| **T** | Tone (Dialing to be continued in Dual Tone Multifrequency, DTMF, mode).<br>Optionally accepted parameter. |
| **&** | Flash. (The flash duration varies by country.)<br>Optionally accepted parameter. |

**Note** The ITU-T carries out the functions of the former Consultative Committee for International Telegraph and Telephone (CCITT).

## Example

The following example specifies a DDR telephone number to be tone dialed on interface async 1 using the **dialer string** command:

```
interface async 1
dialer string T14085553434
```

## Related Commands

**dialer-group**
**dialer in-band**
**dialer map**
**script dialer**

# dialer wait-for-carrier-time

To specify how long to wait for a carrier, use the **dialer wait-for-carrier-time** interface configuration command. To reset the carrier wait time value to the default, use the **no** form of this command.

**dialer wait-for-carrier-time** *seconds*
**no dialer wait-for-carrier-time**

## Syntax Description

| | |
|---|---|
| *seconds* | Number of seconds that the interface waits for the carrier to come up when a call is placed. Acceptable values are positive, nonzero integers. |

## Default

30 seconds

## Command Mode

Interface configuration

## Usage Guidelines

On asynchronous interfaces, the **dialer wait-for-carrier-time** command sets the total time allowed for the chat script to run.

If a carrier signal is not detected in this amount of time, the interface is disabled until the enable timeout occurs (configured with the **dialer enable-timeout** command).

## Example

The following example specifies a carrier wait time of 45 seconds on interface async 1:

```
interface async 1
dialer wait-for-carrier-time 45
```

## Related Command

**dialer enable-timeout**

# dialer-group

To control access, use the **dialer-group** interface configuration command. To remove an interface from the specified dialer access group, use the **no** form of this command.

**dialer-group** *group-number*
**no dialer-group**

## Syntax Description

| | |
|---|---|
| *group-number* | Number of the dialer access group to which the specific interface belongs. This access group is defined using the **dialer-list** command. Acceptable values are nonzero, positive integers between 1 and 10. |

## Default

No access is predefined.

## Command Mode

Interface configuration

## Usage Guidelines

An interface can only be associated with a single dialer access group; multiple **dialer-group** assignment is not allowed. A second dialer access group assignment will override the first. A dialer access group is defined with the **dialer-group** command. The **dialer-list** command associates an access list with a dialer access group.

## Example

The following example specifies dialer access group number 1.

The destination address of the packet is evaluated against the access list specified in the associated **dialer-list** command. If it passes, a call is initiated (if no connection has already been established) or the idle timer is reset (if a call is currently connected).

```
interface async 1
dialer-group 1
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
```

## Related Command

**dialer-list**

# dialer-list list

To group access lists, use the **dialer-list list** global configuration command. To disable automatic dialing, use the **no** form of this command.

**dialer-list** *dialer-group* **list** *access-list-number*
**no dialer-list** *dialer-group* **list** *access-list-number*

## Syntax Description

| | |
|---|---|
| *dialer-group* | Specifies the number of a dialer access group identified in any **dialer-group** interface configuration command. |
| *access-list-number* | Specifies the access list number specified in any IP or Novell IPX access lists including Novell IPX extended, Service Access Point (SAP) access lists and bridging type. See the "Dialer-List List Command Access List Types and Numbers" table for the supported access list types and numbers. |

## Default

None

## Command Mode

Global configuration

## Usage Guidelines

The **dialer-list list** command applies access lists to dialer access groups to control dialing using DDR. This command applies access lists to dialer access groups defined with the **dialer-group** command. See the *Router Products Configuration Guide* for more information about configuring access lists.

To specify additional protocols and access control with a finer granularity, see the **dialer-list protocol** command.

Table 8-5 lists the access list types and numbers that the **dialer-group** command supports.

**Table 8-5 Dialer-List List Command Access List Types and Numbers**

| Access List Type | Access List Number Range |
|---|---|
| Standard IP | 1-99 |
| Extended IP | 100-199 |
| Transparent Bridging | 200-299 |
| Standard Novell IPX | 800-899 |
| Extended Novell IPX | 900-999 |

## Example

In the following example, dialing occurs when an interesting packet (one that matches access list specifications) needs to be output on an interface. Using the standard access list method, packets can be classified as interesting or uninteresting. To specify that IGRP TCP/IP routing protocol updates are not interesting (relative to DDR automatic dialing), the following access list would be defined:

```
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
```

To permit all other IP traffic, the preceding would be modified as follows:

```
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
```

Then the following command would be used to place list 101 into dialer access group 1:

```
dialer-list 1 list 101
```

## Related Command

**dialer-group**

# dialer-list protocol

To define a DDR dialer list to control dialing by protocol or by a combination of protocol and access list, use the **dialer-list protocol** global configuration command. To delete a dialer list, use the **no** form of this command.

**dialer-list** *dialer-group* **protocol** *protocol-name* {**permit** | **deny** | **list** *access-list-number* | *access-group*}
**no dialer-list** *dialer-group* [**protocol** *protocol-name* [**list** *access-list-number* | *access-group*]]

## Syntax Description

| | |
|---|---|
| *dialer-group* | Number of a dialer access group identified in any **dialer-group** interface configuration command. |
| *protocol-name* | One of the following protocol keywords: **appletalk**, **bridge**, **clns**, **clns_es**, **clns_is**, **decnet**, **decnet_router-L1**, **decnet_router-L2**, **decnet_node**, **ip**, **ipx**, **vines**, or **xns**. |
| **permit** | (Optional) Permits access to an entire protocol. |
| **deny** | (Optional) Denies access to an entire protocol. |
| **list** | Specifies that an access list will be used for defining a granularity finer than an entire protocol. |
| *access-list-number* | Access list number. Access list numbers include any DECnet, Banyan VINES, IP, Novell IPX, or XNS standard or extended access lists, Novell IPX extended, Service Access Point (SAP) access lists and bridging types. See "Table 8-6" in the "Usage Guidelines" section for the supported access list types and numbers. |
| *access-group* | Filter list name used in the **clns filter-set** and **clns access-group** commands. |

## Default

No dialer lists are defined.

## Command Mode

Global configuration

## Usage Guidelines

The various **no** forms of this command have the following effects:

- The **no dialer-list 1** command deletes all lists configured with list 1, regardless of the keyword previously used (**permit**, **deny**, **protocol**, or **list**).

- The **no dialer-list 1 protocol** *protocol-name* command deletes all lists configured with list 1 and protocol *protocol-name*.
- The **no dialer-list 1 protocol** *protocol-name* **list** *access-list-number* command deletes the specified list.

The **dialer-list protocol** form of this command permits or denies access to an entire protocol. The **dialer-list protocol list** form of this command provides a finer permission granularity and also supports protocols that were not previously supported.

The **dialer-list protocol list** form of this command applies protocol access lists to dialer access groups to control dialing using DDR. The dialer access groups are defined with the **dialer-group** command. See the *Router Products Configuration Guide* for more information about configuring access lists for protocols.

Although the **dialer-list list** command is still supported for IP, IPX, DECnet, AppleTalk, XNS, and bridging, the new **dialer-list protocol list** form of this command should be used for all protocols. The **dialer-list protocol list** form of this command is supported for all those protocols and also for Banyan VINES and ISO CLNS.

Table 8-6 lists the access list types and numbers that the **dialer-list protocol list** command supports. The table does not include ISO CLNS because that protocol uses filter names instead of predefined access list numbers.

**Table 8-6 Dialer-List Supported Access List Types and Numbers**

| Access List Type | Access List Number Range (decimal) |
|---|---|
| AppleTalk | 600-699 |
| Banyan VINES (standard) | 1-100 |
| Banyan VINES (extended) | 101-200 |
| DECnet | 300-399 |
| IP (standard) | 1-99 |
| IP (extended) | 100-199 |
| Novell IPX (standard) | 800-899 |
| Novell IPX (extended) | 900-999 |
| Transparent Bridging | 200-299 |
| XNS | 500-599 |

## Examples

In the following example, dialing occurs when an interesting packet (one that matches access list specifications) needs to be output on an interface. Using the standard access list method, packets can be classified as interesting or uninteresting. To specify that IGRP TCP/IP routing protocol updates are not interesting (relative to DDR automatic dialing), the following access list would be defined:

```
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
```

To permit all other IP traffic, the preceding example would be modified as follows:

```
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
```

Then the following command would be used to place list 101 into dialer access group 1:

```
dialer-list 1 protocol ip list 101
```

In the following example, DECnet access lists allow any DECnet packets with source area 10 and destination area 20 to trigger calls:

```
access-list 301 permit 10.0 0.1023 10.0 0.1023
access-list 301 permit 10.0 0.1023 20.0 0.1023
```

Then the following command would be used to place list 301 into dialer access group 1:

```
dialer-list 1 protocol decnet list 301
```

In the following example, both IP and VINES access lists are defined. The IP access lists define IGRP packets as uninteresting, but permits other IP packets to trigger calls. The VINES access lists do not allow RTP routing updates to trigger calls, but allow any other data packets to trigger calls.

```
access-list 101 deny igrp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!
vines access-list 107 deny RTP 00000000:0000 FFFFFFFF:FFFF 00000000:0000 FFFFFFFF:FFFF
vines access-list 107 permit IP 00000000:0000 FFFFFFFF:FFFF 00000000:0000 FFFFFFFF:FFFF
```

Then the following two commands place the IP and VINES access lists into dialer access group 1:

```
dialer-list 1 protocol ip list 101
dialer-list 1 protocol vines list 107
```

In the following example, a CLNS filter is defined, then the filter is placed in dialer access group 1:

```
clns filter-set ddrline permit 47.0004.0001....
!
dialer-list 1 protocol clns list ddrline
```

## Related Commands

A dagger (**†**) indicates that the command is documented in another chapter.

**access-list**†
**clns filter-set**†
**dialer-group**
**dialer-list list**
**vines access-list**†

# interface dialer

To define a dialer rotary group, use the **interface dialer** global configuration command.

**interface dialer** *number*

## Syntax Description

| | |
|---|---|
| *number* | Number of the dialer rotary group. It can be number in the range 0 through 255. |

## Default

No dialer rotary groups are predefined.

## Command Mode

Global configuration

## Usage Guidelines

Dialer rotary groups allow you to apply a single interface configuration to a set of physical interfaces. This allows a group of interfaces to be used as a pool of interfaces for calling many destinations.

Once the interface configuration is propagated to a set of interfaces, those interfaces can be used to place calls using the standard DDR criteria. When multiple destinations are configured, any of these interfaces can be used for outgoing calls.

Dialer rotary groups are useful in environments that require multiple calling destinations. Only the rotary group needs to be configured with all of the **dialer map** commands. The only configuration required for the interfaces is the **dialer rotary-group** command indicating that each interface is part of a dialer rotary group.

Although a dialer rotary group is configured as an interface, it is not a physical interface. Instead it represents a group of interfaces. Interface configuration commands entered after the **interface dialer** command will be applied to all physical interfaces assigned to specified rotary groups. Individual interfaces in a dialer rotary group do not have individual addresses. The dialer interface has an address, and that address is used by all interfaces in the dialer rotary group.

## Example

The following example identifies interface dialer 1 as the dialer rotary group leader. Interface dialer 1 is not a physical interface, but represents a group of interfaces. The interface configuration commands that follow apply to all interfaces included in this group.

```
interface dialer 1
encapsulation ppp
authentication chap
dialer in-band
ip address 1.2.3.4
dialer map ip 1.2.2.5 name YYY 14155553434
dialer map ip 1.3.2.6 name ZZZ
```

# ppp authentication chap

To enable Challenge Handshake Authentication Protocol (CHAP) on a serial interface, use the **ppp authentication chap** interface configuration command. To disable this encapsulation, use the **no** form of this command.

**ppp authentication chap** [**if-needed**]
**no ppp authentication chap**

## Syntax Description

| | |
|---|---|
| **if-needed** | (Optional) CHAP authentication is not done on this line if the user has already authenticated. |

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

Once you have enabled CHAP, the local router requires a password from remote devices. If the remote device does not support CHAP, no traffic will be passed to that device.

The **if-needed** option affects only lines that run EXEC and have teletype devices (TTYs) associated with them. This option affects the router AUX port.

## Example

The following example enables CHAP on serial interface 4:

```
interface serial 4
encapsulation ppp
ppp authentication chap
```

## Related Commands

**encapsulation ppp**
**ppp authentication pap**

# ppp authentication pap

To enable Password Authentication Protocol (PAP) on a serial interface, use the **ppp authentication pap** interface configuration command. To disable this encapsulation, use the **no** form of this command.

**ppp authentication pap** [**if-needed**]
**no ppp authentication pap**

## Syntax Description

| | |
|---|---|
| **if-needed** | (Optional) PAP authentication is not done on this line if the user has already authenticated. |

## Default

Disabled

## Command Mode

Interface configuration

## Usage Guidelines

When PAP is enabled, the remote router attempting to connect to the local router is required to send an authentication request. If the username and password specified in the authentication request are accepted, the router sends an authentication acknowledgment.

The **if-needed** option affects only lines that run EXEC and have teletype devices (TTYs) associated with them. This option affects the router AUX port.

## Example

The following example enables CHAP on serial interface 4:

```
interface serial 4
encapsulation ppp
ppp authentication pap
```

## Related Commands

**encapsulation ppp**
**ppp authentication chap**

# script dialer

To specify a default modem chat script, use the **script dialer** line configuration command. Use the **no** form of this command to disable this feature.

**script dialer** *regexp*
**no script dialer**

## Syntax Description

| | |
|---|---|
| *regexp* | Specifies the set of modem scripts that might be executed. The first script that matches the argument *regexp* will be used. |

## Default

No chat script is defined.

## Command Mode

Line configuration

## Usage Guidelines

This command is used by dial-on-demand routing modules to provide modem dialing commands and commands to log in to remote systems.

The argument *regexp* is used to specify the name of the modem script that is to be executed. The first script that matches the argument in this command and the dialer map command will be used. For more information about regular expressions, refer to the "Regular Expressions" appendix in the this publication.

If you adhered to the recommended naming convention for chat scripts, the modem lines (the argument *regexp* in the **script dialer** command) would be set to one of the following regular expressions to match patterns, depending on the kind of modem you have:

- codex-.*
- telebit-.*
- usr-.*
- xyz-.*

In the **dialer map** command, you could specify the modulation but leave the type of modem unspecified, as in ".*-v32bis."

## Example

The following example shows line chat scripts being specified for lines connected to Telebit and US Robotics modems:

```
! Some lines have telebit modems
line 1 6
dialer script telebit.*
! Some lines have US robotics modems
line 7 12
```

```
dialer script usr.*
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**chat-script**
**dialer map modem-script system-script**
**dialer map modem-script system-script name**
**script activation**†
**script connection**†
**script reset**†
**script startup**†
**start-chat** †

# show dialer

To obtain a general diagnostic display for serial interfaces configured for DDR, use the **show dialer** EXEC command.

**show dialer** [**interface** *type number*]

## Syntax Description

| | |
|---|---|
| **interface** | (Optional) Information for the interface specified by the arguments *type* and *number* is to be displayed. |
| *type* | (Optional) Interface type. |
| *number* | (Optional) Interface number. |

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show dialer** command for an asynchronous interface:

```
Router# show dialer interface async 1
Async1 - dialer type = IN-BAND NO-PARITY
Idle timer (900 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Time until disconnect 838 secs
Current call connected 0:02:16
Connected to 8986

 Dial String      Successes   Failures    Last called   Last status
 8986                   0         0        never                     Default
 8986                   8         3        0:02:16      Success      Default
```

Table 8-7 describes significant fields shown in the display.

**Table 8-7 Show Dialer Field Descriptions for In-Band Dialers**

| Field | Description |
|---|---|
| Async 1 | Name of an asynchronous interface. |
| dialer type = IN-BAND | Indicates that DDR is enabled. |
| Idle timer (900 secs) | Idle timeout specification (in seconds). |
| Fast idle timer (20 secs) | Fast idle timer specification (in seconds). |
| Wait for carrier (30 secs) | Wait for carrier timer specification (in seconds). |
| Re-enable (15 secs) | Enable timeout specification (in seconds). |
| Time until disconnected | Time until line is configured to disconnect. |
| Current call connected | Time at which the current call was connected. |
| Connected to | Dial string to which line is currently connected. |

| Field | Description |
|---|---|
| Dial string | Dial strings of logged calls (telephone numbers). On ISDN BRI interfaces, if you have specified a subaddress number in the **dialer string** or **dialer map** command, this number is included in the dial string after a colon. |
| Successes | Successful connections (even if no data is passed). |
| Failures | Failed connections; call not successfully completed. |
| Last called | Time that last call occurred to specific dial string. |
| Last status | Status of last call to specific dial string (successful or failed). |
| Default | If the DDR facility is using the dial string specified with the **dialer string** command, the word Default is appended to the Last status entry. |

When the **show dialer** EXEC command is issued for a synchronous serial interface configured for DTR dialing, output similar to the following is displayed:

```
Serial 0 - dialer type = DTR SYNC
Idle timer (120 secs), Fst idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)

 Dial String   Successes   Failures    Last called   Last status
 ----                1         0        1:04:47      Success     DTR dialer
 8986                0         0        never                    Default
```

Table 8-8 describes new fields shown in the display.

**Table 8-8 Show Dialer Field Descriptions for DTR Dialers**

| Field | Description |
|---|---|
| DTR SYNC | Indicates that DDR is enabled and that DTR dialing is enabled on this synchronous interface. |
| Last status: Success | Indicates that the last call was successful and that DTR dialing was used. |
| DTR dialer | Phrase appended to the Last status entry to indicate that this is a DTR dialer. |

If an interface is connected to a destination, a display is provided that indicates the idle time before the line is disconnected (decrements each second). Then the duration of the current connection is shown. The following shows an example of this display; it would appear after the third line in the **show dialer** display.

```
Time until disconnect 596 secs
Current call connected 0:00:25
```

After a call disconnects, the system displays the time remaining before being available to dial again. The following is an example of this display; it would appear after the third line in the **show dialer** display:

```
Time until interface enabled 8 secs
```

If the **show dialer** command is issued for an interface on which DDR is not enabled, the system displays an error message. The following is a sample error message:

```
Async 1 - Dialing not enabled on this interface.
```

If an interface is configured for DDR, the **show interfaces** command displays the following message:

```
Async1 is up, line protocol is up (spoofing)
Hardware is Async Serial
```

The word *spoofing* indicates that the line really is not up, but the dialer is forcing the line to masquerade as "up" so that upper level protocols will continue to operate as expected. (Spoofing is a state added to allow DDR to work. Basically, the interface "dials on demand" in response to packets being routed to it. No packets are routed to down interfaces, so the router interface must pretend to be up [spoof] so packets will be routed to it when it's not connected. It's the normal idle state on a dial-on-demand interface.)

If caller ID screening is configured on an ISDN BRI, the **show dialer** command display includes a line similar to the following:

```
1 incoming call(s) have been screened.
```

This line reports the number of calls that have been screened by the router.

# show snapshot

To display snapshot routing parameters associated with an interface, use the **show snapshot** EXEC command.

**show snapshot** [*interface*]

## Syntax Description

| | |
|---|---|
| *interface* | (Optional) Interface type and number. |

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show snapshot** command:

```
Router# show snapshot serial 1

Serial1 is up, line protocol is up, snapshot up
Options: dialer support
Length of each activation period: 3 minutes
Period between activations:       10 minutes
Retry period on connect failure:  10
For dialer address 240
 Current queue: active, remaining active time: 3 minutes
 Updates received this cycle: ip, ipx, appletalk
For dialer address 1
 Current queue: client quiet, time until next activation: 7 minutes
```

Table 8-9 describes the fields shown in the display.

**Table 8-9 Show Snapshot Fields**

| Field | Description |
|---|---|
| Serial1 is up, line protocol is up | Indicates whether the interface hardware is currently active (whether carrier detect is present) and if it has been taken down by an administrator. |
| snapshot up | Indicates whether the snapshot protocol is enabled on the interface. |
| Options: | Options configured on the **snapshot client** or **snapshot server** interface configuration command. It can be one of the following:<br>• dialer support—Snapshot routing is configured with the **dialer** keyword.<br>• stay asleep on carrier up—Snapshot routing is configured with the **suppress-statechange-update** keyword. |
| Length of each activation period | Length of the active period. |
| Period between activations | Length of the quiet period. |
| Retry period on connect failure | Length of the retry period. |
| For dialer address | Displays information about each dialer rotary group configured with the **dialer map** command. |

| Field | Description |
|---|---|
| Current queue: | Indicates which period snapshot routing is currently in. It can be one of the following:<br>• active—Routing updates are being exchanged.<br>• client quiet—The client router is in a quiet period and routing updates are not being exchanged.<br>• server quiet—The server router is in a quiet period, awaiting an update from the client router before awakening, and routing updates are not being exchanged.<br>• post active—Routing updates are not being exchanged. If the server router receives an update from the client router, it processes it but does not begin an active period. This allows time for resynchronization of active periods between the client and server routers.<br>• no queue—This is a temporary holding queue for new snapshot routing interfaces and for interfaces being deleted. |
| remaining active time<br>time until next activation | Time remaining in the current period. |
| Updates received this cycle | Protocols from which routing updates have been received in the current active period. This line is displayed only if the router is in an active period. |

# snapshot client

To configure a client router for snapshot routing, use the **snapshot client** interface configuration command. To disable a client router, use the **no** form of this command.

**snapshot client** *active-time quiet-time* [**suppress-statechange-updates**] [**dialer**]
**no snapshot client** *active-time quiet-time* [**suppress-statechange-updates**] [**dialer**]

## Syntax Description

| | |
|---|---|
| *active-time* | Amount of time, in minutes, that routing updates are regularly exchanged between the client and server routers. This can be an integer in the range 5 to 100. There is no default value. A typical value would be 5 minutes. |
| *quiet-time* | Amount of time, in minutes, that routing entries are frozen and remain unchanged between active periods. Routes are not aged during the quiet period, so they remain in the routing table as if they were static entries. This argument can be an integer from 8 to 100000. There is no default value. The minimum quiet time is generally the active time plus 3. |
| **suppress-statechange-updates** | (Optional) Disables the exchange of routing updates each time the line protocol goes from "down" to "up" or from "dialer spoofing" to "fully up." |
| **dialer** | (Optional) Allows the client router to dial up the remote router in the absence of regular traffic. |

## Default

Snapshot routing is disabled.

The *active-time* and *quiet-time* arguments have no default values.

## Command Mode

Interface configuration

## Usage Guidelines

The value of the *active-time* argument must be the same for the client and server routers.

To specify the remote server routers to be called by this client router during each active period, use the **dialer map snapshot** command.

## Example

The following example configures a client router for snapshot routing:

```
interface dialer 1
snapshot client 5 600 suppress-statechange-updates dialer
```

### Related Commands

**clear snapshot quiet-time**
**dialer map**
**show snapshot**
**snapshot server**

# snapshot server

To configure a server router for snapshot routing, use the **snapshot server** interface configuration command. To disable a server router, use the **no** form of this command.

**snapshot server** *active-time* [**dialer**]
**no snapshot server** *active-time* [**dialer**]

## Syntax Description

| | |
|---|---|
| *active-time* | Amount of time, in minutes, that routing updates are regularly exchanged between the client and server routers. This can be an integer in the range 5 to 100. There is no default value. A typical value would be 5 minutes. |
| **dialer** | (Optional) Allows the client router to dial up the remote router in the absence of regular traffic. |

## Default

Snapshot routing is disabled.

The *active-time* argument has no default value.

## Command Mode

Interface configuration

## Usage Guidelines

The value of the *active-time* argument must be the same for the client and server routers.

## Example

The following example configures a server router for snapshot routing:

```
interface dialer 1
snapshot server 5
```

## Related Commands

**show snapshot**
**snapshot client**

# username

To specify the password to be used in Challenge Handshake Authentication Protocol (CHAP) caller identification and Password Authentication Protocol (PAP), use the **username** command.

**username** *name* **password** *secret*

## Syntax Description

| | |
|---|---|
| *name* | Host name, server name, user ID, or command name. |
| **password** | Possibly an encrypted password for this username. |
| *secret* | For CHAP authentication: specifies the secret for the local router or the remote device. The secret is encrypted when it is stored on the local router. This prevents the secret from being stolen. The secret can consist of any string of up to 11 printable ASCII characters. There is no limit to the number of username/password combinations that can be specified, allowing any number of remote devices to be authenticated. |

## Default

No password is predefined.

## Command Mode

Global configuration

## Usage Guidelines

Add a **name** entry for each remote system that the local router requires authentication from.

The **username** command is required as part of the configuration for authentication protocols, such as CHAP and PAP. For each remote system that the local router communicates with from which it requires authentication, you add a **username** entry.

---

**Note** To enable the local router to respond to remote CHAP challenges, one **username** *name* entry must be the same as the **hostname** *name* entry that has already been assigned to your router.

---

If there is no secret specified and **debug serial-interface** is enabled, an error is displayed when a link is established and the authentication protocol challenge is not implemented. Debugging information about authentication protocols is available using the **debug serial-interface** and **debug serial-packet** commands. See the *Debug Command Reference* publication for more information.

## Example

The following example configuration enables CHAP on interface serial 0. It also defines a password for the local server, Adam, and a remote server, Eve.

```
hostname Adam
interface serial 0
encapsulation ppp
ppp authentication chap
username Eve password theirsystem
```

When you look at your configuration file, the passwords will be encrypted and the display will look similar to the following:

```
hostname Adam
interface serial 0
encapsulation ppp
ppp authentication chap
username Eve password 7 121F0A18
```

## Related Command

A dagger (†) indicates that the command is documented in another chapter.

**hostname**†

CHAPTER 9

# Frame Relay Commands

Use the commands described in this chapter to configure access to Frame Relay networks.

For Frame Relay configuration information and examples, refer to the “Configuring Frame Relay” chapter in the *Router Products Configuration Guide*.

# clear frame-relay-inarp

To clear dynamically created Frame Relay maps, which are created by the use of Inverse ARP, use the **clear frame-relay-inarp** EXEC command.

**clear frame-relay-inarp**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Example

The following example clears dynamically created Frame Relay maps:

```
clear frame-relay-inarp
```

## Related Commands

**frame-relay inverse-arp**
**show frame-relay map**

# encapsulation frame-relay

To enable Frame Relay encapsulation, use the **encapsulation frame-relay** interface configuration command. To disable Frame Relay encapsulation, use the **no** form of this command.

**encapsulation frame-relay** [**cisco** | **ietf**]
**no encapsulation frame-relay** [**ietf**]

## Syntax Description

| | |
|---|---|
| **cisco** | (Optional) Uses Cisco's own encapsulation, which is a four-byte header, with two bytes to identify the DLCI and two bytes to identify the packet type. This is the default. |
| **ietf** | (Optional) Sets the encapsulation method to comply with the IETF standard (RFCs 1294 and 1490). Use this keyword when connecting to another vendor's equipment across a Frame Relay network. |

## Default

Enabled

## Command Mode

Interface configuration

## Usage Guidelines

Use this command with no keywords to restore the default Cisco encapsulation.

## Examples

The following example configures Cisco Frame Relay encapsulation on interface serial 1:

```
interface serial 1
encapsulation frame-relay
```

Use the **ietf** keyword if your router is connected to another vendor's equipment across a Frame Relay network to conform with RFCs 1294 and 1490:

```
interface serial 1
encapsulation frame-relay ietf
```

# frame-relay broadcast-queue

To create a special queue for a specified interface to hold broadcast traffic that has been replicated for transmission on multiple DLCIs, use the **frame-relay broadcast-queue** interface configuration command.

**frame-relay broadcast-queue** *size byte-rate packet-rate*

## Command Syntax

| | |
|---|---|
| *size* | Number of packets to hold in the broadcast queue. The default is 64 packets. |
| *byte-rate* | Maximum number of bytes to be transmitted per second. The default is 256000 bytes per second. |
| *packet-rate* | Maximum number of packets to be transmitted per second. The default is 36 packets per second. |

## Default

The default values are as follows:

*size*—64 packets
*byte-rate*—256000 bytes per second
*packet-rate*—36 packets per second

## Command Mode

Interface configuration

## Usage Guidelines

For purposes of the Frame Relay broadcast queue, broadcast traffic is defined as packets that have been replicated for transmission on multiple DLCIs, but it does not include the original routing packet or SAP packet, which passes through the normal queue. Due to timing sensitivity, bridged broadcasts and spanning tree packets are sent through the normal queue.

The Frame Relay broadcast queue is managed independently of the normal interface queue. It has its own buffers and a configurable service rate.

A broadcast queue is given a maximum transmission rate (throughput) limit measured in bytes per second and packets per second. The queue is serviced to ensure that only this maximum is provided. The broadcast queue has priority when transmitting at a rate below the configured maximum, and hence has a guaranteed minimum bandwidth allocation. The two transmission rate limits are intended to avoid flooding the interface with broadcasts. The actual limit in any second is the first rate limit that is reached.

Given the transmission rate restriction, additional buffering will be required to store broadcast packets. The broadcast queue is configurable to store large numbers of broadcast packets.

The queue size should be set to avoid loss of broadcast routing update packets. The exact size will depend on the protocol being used and the number of packets required for each update. To be safe, set the queue size so that one complete routing update from each protocol and for each DLCI can be stored. As a general rule, start with 20 packets per DLCI.

As a general rule, the byte rate should be less than both of the following:

- N/4 times the minimum remote access rate (measured in *bytes* per second), where N is the number of DLCIs to which the broadcast must be replicated
- 1/4 the local access rate (measured in *bytes* per second)

The packet rate is not critical if you set the byte rate conservatively. As a general rule, set the packet rate assuming 250-byte packets.

## Example

The following example specifies a broadcast queue to hold 80 packets, to have a maximum byte transmission rate of 240,000 bytes per second, and to have a maximum packet transmission rate of 160 packets per second:

```
frame-relay broadcast-queue 80 240000 160
```

# frame-relay de-group

To specify the discard eligibility (DE) group number to be used for a specified DLCI, use the **frame-relay de-group** interface configuration command. To disable a previously defined group number assigned to a specified DLCI, use the **no** form of the command with the relevant keyword and arguments.

**frame-relay de-group** *group-number dlci*
**no frame-relay de-group** [*group-number*] [*dlci*]

## Syntax Description

| | |
|---|---|
| *group-number* | DE group number to apply to the specified DLCI number, in the range from 1 through 10. |
| *dlci* | DLCI number. |

## Default

No DE group is defined.

## Command Mode

Interface configuration

## Usage Guidelines

To disable all previously defined group numbers, use the **no** form of this command with no arguments.

This command requires that Frame Relay software be enabled.

The DE bit is not set or recognized by the Frame Relay switching code, but must be recognized and interpreted by the Frame Relay network.

## Example

The following example specifies that group number 3 will be used for DLCI 170:

```
frame-relay de-group 3 170
```

## Related Command

**frame-relay de-list**

# frame-relay de-list

To define a discard eligibility (DE) list specifying which packets will have the DE bit set and thus will be eligible for discarding when congestion is experienced on the Frame Relay switch, use the **frame-relay de-list** global configuration command. To delete a portion of a previously defined DE list, use the **no** form of this command.

**frame-relay de-list** *list-number* {**protocol** *protocol* | **interface** *type number*} *characteristic*
**no frame-relay de-list** *list-number* {**protocol** *protocol* | **interface** *type number*} *characteristic*

## Syntax Description

| | |
|---|---|
| *list-number* | Number of the DE list. |
| *protocol* | One of the following keywords corresponding to a supported protocol or device:<br>**arp**—Address Resolution Protocol.<br>**apollo**—Apollo Domain.<br>**appletalk**—AppleTalk.<br>**bridge**—bridging device.<br>**clns**—ISO Connectionless Network Service.<br>**clns_es**—CLNS end systems.<br>**clns_is**—CLNS intermediate systems.<br>**compressedtcp**—Compressed TCP.<br>**decnet**—DECnet.<br>**decnet_node**—DECnet end node.<br>**decnet_router-L1**—DECnet Level 1 (intra-area) router.<br>**decnet_router-L2**—DECnet Level 2 (interarea) router.<br>**ip**—Internet Protocol.<br>**ipx**—Novell Internet Packet Exchange.<br>**vines**—Banyan VINES.<br>**xns**—Xerox Network Systems. |
| *type* | One of the following interface types: **serial**, **null**, or **ethernet**. |
| *number* | Interface number. |
| *characteristic* | You must supply one of the following:<br>**fragments**—Classify fragmented IP packets.<br>**tcp** *port*—TCP packets to or from a specified port.<br>**udp** *port*—UDP packets to or from a specified port.<br>**list** *access-list-number*—Previously defined access list number.<br>**gt** *bytes*—Packets larger than the specified number of bytes will have the DE bit set.<br>**lt** *bytes*—Packets smaller than the specified number of bytes will have the DE bit set. |

## Default

Discard eligibility is not defined.

## Command Mode

Global configuration

## Usage Guidelines

To remove an entire DE list, use the **no** form of this command with no options and arguments.

This prioritization feature requires that the Frame Relay network be able to interpret the DE bit as indicating which packets can be dropped first in case of congestion or which packets are less time sensitive or both.

## Example

The following example specifies that IP packets larger than 512 bytes will have the discard eligibility bit set.

```
frame-relay de-list 1 protocol ip gt 512
```

# frame-relay interface-dlci

To assign a DLCI to a specified Frame Relay subinterface on the router, use the **frame-relay interface-dlci** interface configuration command. To remove this assignment, use the **no** form of this command.

**frame-relay interface-dlci** *dlci* [*option*]
**no frame-relay interface-dlci** *dlci* [*option*]

**frame-relay interface-dlci** *dlci* [**protocol ip** *ip-address*]

## Syntax Description

| | |
|---|---|
| *dlci* | A DLCI number to be used on the specified subinterface. |
| *option* | (Optional) Broadcast or encapsulation keyword, as defined in the "Frame Relay Interface-DLCI Option Keywords" table. |
| **protocol ip** *ip-address* | Indicates the IP address of the serial interface of a new router onto which a router configuration file is to be autoinstalled over a Frame Relay network. Use this option only when this router will act as the BOOTP server for autoinstallation over Frame Relay. |

## Default

No DLCI is assigned.

## Command Mode

Interface configuration

## Usage Guidelines

Use this command only for subinterfaces on a router. Use of the command on an interface, rather than a subinterface, will prevent the router from forwarding packets intended for that DLCI.

Subinterfaces are logical interfaces associated with a physical interface. To use this command, you must be in subinterface configuration mode. This requires making the logical subinterface assignment before assigning any DLCIs and any encapsulation or broadcast options. See the "Example" section for the sequence of commands.

Use the **protocol ip** *ip-address* option only when this router will act as the BOOTP server for autoinstallation over Frame Relay.

For more information about autoinstalling router configuration files over a Frame Relay network, see the "Loading System Images, Microcode Images, and Configuration Files" chapter in the *Router Products Configuration Guide*.

Table 9-1 lists the **frame-relay interface-dlci** option keywords.

**Table 9-1 Frame Relay Interface-DLCI Option Keywords**

| Keyword | Option |
|---|---|
| **broadcast** | Broadcasts should be forwarded out through this interface. |
| **ietf** | IETF Frame Relay encapsulation. |
| **cisco** | Cisco Frame Relay encapsulation. |

## Example

The following example assigns DLCI 100 to subinterface serial 5.17:

```
! Enter interface configuration and begin assignments on interface serial 5
interface serial 5
! Enter subinterface configuration by assigning subinterface 17
interface serial 5.17
! Now assign a DLCI number to subinterface 5.17
frame-relay interface-dlci 100
```

# frame-relay intf-type

Use the **frame-relay intf-type** interface configuration command to configure a Frame Relay switch type. Use the **no** form of this command to disable the switch.

**frame-relay intf-type** [**dce** | **dte** | **nni**]
**no frame-relay intf-type** [**dce** | **dte** | **nni**]

## Syntax Description

| | |
|---|---|
| **dce** | (Optional) Router functions as a switch connected to a router. |
| **dte** | (Optional) Router is connected to a Frame Relay network. |
| **nni** | (Optional) Router functions as a switch connected to a switch (supports NNI connections). |

## Default

**dte**

## Command Mode

Interface configuration

## Usage Guidelines

This command can be used only if Frame Relay switching has previously been enabled globally by use of the **frame-relay switching** command.

## Example

The following example configures a DTE switch type:

```
frame-relay switching
!
interface serial 2
frame-relay intf-type dte
```

# frame-relay inverse-arp

If the Inverse Address Resolution Protocol (InvARP) was previously disabled on a router configured for Frame Relay, use the **frame-relay inverse-arp** interface configuration command to reenable InvARP. Use the **no** form of this command to disable this feature.

**frame-relay inverse-arp** *protocol dlci*
**no frame-relay inverse-arp** *protocol dlci*

## Syntax Description

| | |
|---|---|
| *protocol* | Supported protocols: **appletalk**, **decnet**, **ip**, **ipx**, **vines**, and **xns**. |
| *dlci* | One of the DLCI numbers used on the interface. Acceptable numbers are integers in the range 16 through 1007. |

## Default

Enabled.

## Command Mode

Interface configuration

## Usage Guidelines

This implementation of Inverse ARP is based on RFC 1293. It allows a router running Frame Relay to discover the protocol address of a device associated with the virtual circuit.

In Frame Relay, permanent virtual circuits are identified by a DLCI, which is the equivalent of a hardware address. By exchanging signaling messages, a network announces a new virtual circuit, and with Inverse ARP, the protocol address at the other side of the circuit can be discovered.

The **show frame-relay map** command displays the word "dynamic" to flag virtual circuits that are created dynamically by Inverse ARP.

## Example

The following example sets Inverse ARP on an interface running AppleTalk:

```
interface serial 0
frame-relay inverse-arp appletalk 100
```

## Related Commands

**clear frame-relay-in1arp**
**show frame-relay map**

# frame-relay ip tcp header-compression

To configure an interface to ensure that the associated PVC will always carry outgoing TCP/IP headers in compressed form, use the **frame-relay ip tcp header-compression** interface configuration command. To disable compression of TCP/IP packet headers on the interface, use the **no** form of this command.

**frame-relay ip tcp header-compression** [**passive**]
**no frame-relay ip tcp header-compression**

## Syntax Description

| | |
|---|---|
| **passive** | (Optional) Compresses the outgoing TCP/IP packet header only if an incoming packet had a compressed header. |

## Default

Active TCP/IP header compression; all outgoing TCP/IP packets are subjected to header compression.

## Command Mode

Interface configuration

## Usage Guidelines

This command applies to interfaces that support Frame Relay encapsulation, specifically serial ports and HSSI.

Frame Relay must be configured on the interface before this command can be used.

TCP/IP header compression and IETF encapsulation are mutually exclusive. If an interface is changed to IETF encapsulation, all encapsulation and compression characteristics are lost.

When you use this command to enable TCP/IP header compression, every IP map will inherit the compression characteristics of the interface, unless header compression is explicitly rejected or modified by using the **frame-relay map ip header compression** command.

## Example

The following example configures serial interface 1 to use the default encapsulation (**cisco**) and passive TCP header compression:

```
interface serial 1
encapsulation frame-relay
frame-relay ip tcp header-compression passive
```

## Related Command

**frame-relay map ip tcp header-compression**

# frame-relay keepalive

To enable the Local Management Interface (LMI) mechanism for serial lines using Frame Relay encapsulation, use the **frame-relay keepalive** interface configuration command. Use the **no** form of this command to disable this capability.

**frame-relay keepalive** *number*
**no frame-relay keepalive**

## Syntax Description

| | |
|---|---|
| *number* | An integer that defines the keepalive interval. The interval must be set and must be less than the interval set on the switch; see the **frame-relay lmi-t392dce** command description. |

## Default

10 seconds

## Command Mode

Interface configuration

## Usage Guidelines

The **frame-relay keepalive** and **keepalive** commands perform the same function; both commands enable the keepalive sequence. The keepalive sequence is part of the Local Management Interface (LMI) protocol, so these commands also control the enabling and disabling of the LMI.

When viewing the configuration information using the **show configuration** command, only the **keepalive** command setting is included; you will not see the **frame-relay keepalive** setting.

---

**Note** When netbooting over Frame Relay, it might be necessary to disable keepalives.

---

## Example

The following example sets the keepalive timer on the server for a period that is two or three seconds faster (shorter interval) than the interval set on the keepalive timer of the Frame Relay switch. The difference in keepalive intervals ensures proper synchronization between the Cisco server and the Frame Relay switch.

```
interface serial 3
frame-relay keepalive 8
```

## Related Commands

A dagger (†) indicates that the command is documented in another chapter.

**keepalive** †
**frame-relay lmi-t392dce**

# frame-relay lmi-n391dte

To set a full status polling interval, use the **frame-relay lmi-n391dte** interface configuration command. To restore the default interval value, assuming an LMI has been configured, use the **no** form of this command.

**frame-relay lmi-n391dte** *keep-exchanges*
**no frame-relay lmi-n391dte** *keep-exchanges*

## Syntax Description

| | |
|---|---|
| *keep-exchanges* | Number of keep exchanges to be done before requesting a full status message. Acceptable value is a positive integer in the range 1 through 255. |

## Default

6 keep exchanges

## Command Mode

Interface configuration

## Usage Guidelines

Use this command when the interface is configured as data terminal equipment (DTE) or network-to-network interface (NNI) as a means of setting the full status message polling interval.

## Example

In the following example, one out of every four status inquiries generated by the router will request a full status response from the switch. The other three status inquiries will request keepalive exchanges only.

```
interface serial 0
frame-relay intf-type DTE
frame-relay lmi-n391dte 4
```

# frame-relay lmi-n392dce

To set the DCE and NNI error threshold, use the **frame-relay lmi-n392dce** interface configuration command. To remove the current setting, use the **no** form of this command.

**frame-relay lmi-n392dce** *threshold*
**no frame-relay lmi-n392dce** *threshold*

## Syntax Description

| | |
|---|---|
| *threshold* | Error threshold value. Acceptable value is a positive integer in the range 1 through 10. |

## Default

2

## Command Mode

Interface configuration

## Usage Guidelines

In Cisco's implementation, N392 errors must occur within the number defined by the N393 event count in order for the link to be declared down. Therefore, the threshold value for this command must be less than the count value defined in the **frame-relay lmi-n393dce** command.

## Example

In the following example, the LMI failure threshold is set to three. The router acts as a Frame Relay DCE or NNI switch.

```
interface serial 0
frame-relay intf-type DCE
frame-relay lmi-n392dce 3
```

## Related Command

**frame-relay lmi-n393dce**

# frame-relay lmi-n392dte

To set the error threshold on a DTE or NNI interface, use the **frame-relay lmi-n392dte** interface configuration command. To remove the current setting, use the **no** form of this command.

**frame-relay lmi-n392dte** *threshold*
**no frame-relay lmi-n392dte** *threshold*

## Syntax Description

| | |
|---|---|
| *threshold* | Error threshold value. Acceptable value is a positive integer in the range 1 through 10. |

## Default

2

## Command Mode

Interface configuration

## Example

In the following example, the LMI failure threshold is set to three. The router acts as a Frame Relay DTE or NNI switch.

```
interface serial 0
frame-relay intf-type DTE
frame-relay lmi-n392dte 3
```

# frame-relay lmi-n393dce

To set the DCE and NNI monitored events count, use the **frame-relay lmi-n393dce** interface configuration command. To remove the current setting, use the **no** form of this command.

**frame-relay lmi-n393dce** *events*
**no frame-relay lmi-n393dce** *events*

## Syntax Description

| | |
|---|---|
| *events* | Monitored events count value. Acceptable value is a positive integer in the range 1 through 10. |

## Default

2

## Command Mode

Interface configuration

## Usage Guidelines

This command and the **frame-relay lmi-n392dce** command define the condition that causes the link to be declared down. In Cisco's implementation, N392 errors must occur within the *events* count in order for the link to be declared down. Therefore, the *events* value defined in this command must be greater than the threshold value defined in the **frame-relay lmi-n392dce** command.

## Example

In the following example, the LMI monitored events count is set to three. The router acts as a Frame Relay DCE or NNI switch.

```
interface serial 0
frame-relay intf-type DCE
frame-relay lmi-n393dce 3
```

## Related Command

**frame-relay lmi-n392dce**

# frame-relay lmi-n393dte

To set the monitored event count on a DTE or NNI interface, use the **frame-relay lmi-n393dte** interface configuration command. To remove the current setting, use the **no** form of this command.

**frame-relay lmi-n393dte** *events*
**no frame-relay lmi-n393dte** *events*

## Syntax Description

| | |
|---|---|
| *events* | Monitored events count value. Acceptable value is a positive integer in the range 1 through 10. |

## Default

2

## Command Mode

Interface configuration

## Example

In the following example, the LMI monitored events count is set to three. The router acts as a Frame Relay DTE or NNI switch.

```
interface serial 0
frame-relay intf-type DTE
frame-relay lmi-n393dte 3
```

# frame-relay lmi-t392dce

To set the polling verification timer on a DCE or NNI interface, use the **frame-relay lmi-t392dce** interface configuration command. To remove the current setting, use the **no** form of this command.

**frame-relay lmi-t392dce** *timer*
**no frame-relay lmi-t392dce** *timer*

## Syntax Description

| | |
|---|---|
| *timer* | Polling verification timer value (in seconds). Acceptable value is a positive integer in the range 5 through 30. |

## Default

15

## Command Mode

Interface configuration

## Usage Guidelines

The value for the timer must be greater than the DTE or NNI keepalive timer.

## Example

The following example indicates a polling verification timer on a DCE or NNI interface set to 20:

```
interface serial 3
frame-relay intf-type DCE
frame-relay lmi-t392dce 20
```

## Related Command

**frame-relay keepalive**

# frame-relay lmi-type

To select the Local Management Interface (LMI) type, use the **frame-relay lmi-type** interface configuration command. To return to the default LMI type, use the **no** form of this command.

**frame-relay lmi-type** {**ansi** | **cisco** | **q933a**}
**no frame-relay lmi-type** {**ansi** | **q933a**}

## Syntax Description

| | |
|---|---|
| **ansi** | Annex D defined by ANSI standard T1.617. |
| **cisco** | LMI type defined jointly by Cisco and three other companies. |
| **q933a** | ITU-T Q.933 Annex A. |

---

**Note** The International Telecommunication Union Telecommunication Standardization Sector (ITU-T) carries out the functions of the former Consultative Committee for International Telegraph and Telephone (CCITT).

---

## Default

Cisco LMI

## Command Mode

Interface configuration

## Usage Guidelines

Cisco's implementation of Frame Relay supports three LMI types: Cisco, ANSI Annex D, and ITU-T Q.933 Annex A.

The LMI type is set on a per-interface basis and is shown in the output of the **show interfaces** EXEC command.

## Example

The following is an example of the commands you enter to select the ANSI Annex D LMI type:

```
interface Serial1
encapsulation frame-relay
frame-relay lmi-type ansi
```

# frame-relay local-dlci

To set the source DLCI for use when the LMI is not supported, use the **frame-relay local-dlci** interface configuration command . To remove the DLCI number, use the **no** form of this command.

**frame-relay local-dlci** *number*
**no frame-relay local-dlci**

---

**Note** The **frame-relay local-dlci** command is provided mainly to allow testing of the Frame Relay encapsulation in a setting where two servers are connected back to back. This command is not required in a live Frame Relay network.

---

## Syntax Description

| | |
|---|---|
| *number* | Local (source) DLCI number to be used. |

## Default

No source DLCI is set.

## Command Mode

Interface configuration

## Usage Guidelines

If LMI is supported and the multicast information element is present, the network server sets its local DLCI based on information provided via the LMI.

## Example

The following example specifies 100 as the local DLCI:

```
interface serial 4
frame-relay local-dlci 100
```

# frame-relay map

Use the **frame-relay map** interface configuration command to define the mapping between a destination protocol address and the DLCI used to connect to the destination address. Use the **no** form of this command to delete the map entry.

**frame-relay map** *protocol protocol-address dlci* [**broadcast**] [**ietf** | **cisco**]
**no frame-relay map** *protocol protocol-address*

## Syntax Description

| | |
|---|---|
| *protocol* | Supported protocol, bridging, or logical link control keywords: **appletalk**, **decnet**, **ip**, **ipx**, **llc2**, **rsrb**, **vines** and **xns**. |
| *protocol-address* | Destination protocol address. |
| *dlci* | DLCI number used to connect to the specified protocol address on the interface. |
| **broadcast** | (Optional) Broadcasts should be forwarded to this address when multicast is not enabled (see the **frame-relay multicast-dlci** command for more information about multicasts). This keyword also simplifies the configuration of OSPF (see the "Usage Guidelines" section for more detail). |
| **ietf** | (Optional) IETF form of Frame Relay encapsulation. Use when the router is connected to another vendor's equipment across a Frame Relay network. |
| **cisco** | (Optional) Cisco encapsulation method. |

## Default

No mapping is defined.

## Command Mode

Interface configuration

## Usage Guidelines

There can be many DLCIs known by a router that can send data to many different places, but they are all multiplexed over one physical link. The Frame Relay map tells the router how to get from a specific protocol and address pair to the correct DLCI.

The optional **ietf** and **cisco** keywords allow flexibility in the configuration. If no keywords are specified in the configuration, the map inherits the attributes set with the **encapsulation frame-relay** command. You can also use the encapsulation options to specify that, for example, all interfaces use IETF encapsulation except one, which needs the original Cisco encapsulation method, and it can be defined using the **cisco** keyword with the **frame-relay map** command.

The **broadcast** keyword provides two functions: It forwards broadcasts when multicasting is not enabled, and it simplifies the configuration of OSPF for nonbroadcast networks that will use Frame Relay.

OSPF treats a nonbroadcast, multiaccess network such as Frame Relay much the same way it treats a broadcast network in that it requires selection of a designated router. In previous releases, this required manual assignment in the OSPF configuration using the **neighbor interface** router command. When the **frame-relay map** command is included in the configuration with the **broadcast**, and **the ip ospf network** command (with the **broadcast** keyword) is configured, there is no need to configure any neighbors manually. OSPF will now automatically run over the Frame Relay network as a broadcast network. (Refer to the **ip ospf network** interface command for more detail.)

---

**Note** The OSPF broadcast mechanism assumes that IP class D addresses are never used for regular traffic over Frame Relay.

---

## Example

The following example maps the destination IP address 131.108.123.1 to DLCI 100:

```
interface serial 0
frame-relay map IP 131.108.123.1 100 broadcast
```

OSPF will use DLCI 100 to broadcast updates.

# frame-relay map bridge

Use the **frame-relay map bridge** interface configuration command to specify that broadcasts should be forwarded when bridging. Use the **no** form of this command to delete the map entry.

**frame-relay map bridge** *dlci* [**broadcast**]
**no frame-relay map bridge** *dlci*

## Syntax Description

| | |
|---|---|
| *dlci* | DLCI number to be used for bridging on the specified interface or subinterface. |
| **broadcast** | (Optional) Broadcasts should be forwarded when multicast is not enabled. |

## Default

No broadcasts are forwarded.

## Command Mode

Interface configuration

## Examples

The following example uses DLCI 144 for bridging:

```
interface serial 0
frame-relay map bridge 144 broadcast
```

The following example sets up separate point-to-point links over a subinterface and runs transparent bridging over it:

```
interface serial 0
bridge-group 1
encapsulation frame-relay
interface serial 0.1
bridge-group 1
frame-relay map bridge 42 broadcast
interface serial 0.2
bridge-group 1
frame-relay map bridge 64 broadcast
interface serial 0.3
bridge-group 1
frame-relay map bridge 73 broadcast
```

DLCI 42 is used as the link; see the section “Frame Relay Configuration Examples” in the *Router Products Configuration Guide* for more examples of subinterfaces.

# frame-relay map clns

Use the **frame-relay map clns** interface configuration command to specify that broadcasts should be forwarded when routing using ISO CLNS. Use the **no** form of this interface configuration command to delete the map entry.

**frame-relay map clns** *dlci* [**broadcast**]
**no frame-relay map clns** *dlci*

## Syntax Description

| | |
|---|---|
| *dlci* | DLCI number to which CLNS broadcasts should be forwarded on the specified interface. |
| **broadcast** | (Optional) Broadcasts should be forwarded when multicast is not enabled. |

## Default

No broadcasts are forwarded.

## Command Mode

Interface configuration

## Example

The following example uses DLCI 125 for ISO CLNS routing:

```
interface serial 0
frame-relay map clns 125 broadcast
```

# frame-relay map ip tcp header-compression

To assign header compression characteristics to an IP map that differ from the compression characteristics of the interface with which the IP map is associated, use the **frame-relay map ip tcp header-compression** interface configuration command. To remove the IP map, use the **no** form of this command.

**frame-relay map ip** *ip-address dlci* [**broadcast**] [**cisco** | **ietf**] [**nocompress**]
**tcp header-compression** {**active** | **passive**}
**no frame-relay map ip** *ip-address dlci*

## Syntax Description

| | |
|---|---|
| *ip-address* | IP address. |
| *dlci* | DLCI number. |
| **broadcast** | (Optional) Forwards broadcasts to the specified IP address. |
| **cisco** | (Optional) Uses Cisco’s proprietary encapsulation. This is the default. |
| **ietf** | (Optional) Uses RFC 1294 encapsulation. No TCP/IP header compression is done if IETF encapsulation is chosen for the IP map or the associated interface. |
| **nocompress** | (Optional) Disables TCP/IP header compression for this map. |
| **active** | Compresses the header of every outgoing TCP/IP packet. |
| **passive** | Compresses the header of an outgoing TCP/IP packet only if an incoming TCP/IP packet had a compressed header. |

## Default

The default encapsulation is **cisco**.

## Command Mode

Interface configuration

## Usage Guidelines

To disable TCP/IP header compression on the IP map, use the **nocompress** form of the command.

IP maps inherit the compression characteristics of the associated interface unless this command is used to provide different characteristics. This command can also be used to reconfigure an IP map that existed before TCP header compression was configured on the associated interface.

When IP maps at both ends of a connection inherit passive compression, the connection will never transfer compressed traffic because neither side will generate a packet with a compressed header.

If you change the encapsulation characteristics of the interface to IETF, you lose the TCP header compression configuration of the associated IP map.

The command **frame-relay map ip** *ip-address dlci* **tcp header-compression active** can also be entered as **frame-relay map ip** *ip-address dlci* **active tcp header-compression**.

## Example

The following example illustrates a command sequence configuring an IP map associated with serial interface 1 to enable active TCP header compression:

```
interface serial 1
encapsulation frame-relay
ip address 131.108.177.170 255.255.255.0
frame-relay map ip 131.108.177.180 190 cisco tcp header-compression active
```

## Related Command

**frame-relay ip tcp header-compression**

# frame-relay multicast-dlci

Use the **frame-relay multicast-dlci** interface configuration command to define the DLCI to be used for multicasts. Use the **no** form of this command to remove the multicast group.

**frame-relay multicast-dlci** *number*
**no frame-relay multicast-dlci**

---

**Note** The **frame-relay multicast-dlci** command is provided mainly to allow testing of the Frame Relay encapsulation in a setting where two servers are connected back to back. This command is not required in a live Frame Relay network.

---

## Syntax Description

| | |
|---|---|
| *number* | Multicast DLCI. (Note that this is *not* the multicast group number, which is an entirely different value.) |

## Default

No DLCI is defined.

## Command Mode

Interface configuration

## Usage Guidelines

Use this command when the multicast facility is not supported. Network transmissions (packets) sent to a multicast DLCI are delivered to all network servers defined as members of the multicast group.

## Example

The following example specifies 1022 as the multicast DLCI:

```
interface serial 0
frame-relay multicast-dlci 1022
```

# frame-relay route

Use the **frame-relay route** interface configuration command to specify the static route for PVC switching. Use the **no** form of this command to remove a static route.

**frame-relay route** *in-dlci out-interface out-dlci*
**no frame-relay route** *in-dlci out-interface out-dlci*

## Syntax Description

| | |
|---|---|
| *in-dlci* | DLCI on which the packet is received on the interface. |
| *out-interface* | Interface the router uses to transmit the packet. |
| *out-dlci* | DLCI the router uses to transmit the packet over the specified *out-interface*. |

## Default

No static route is specified.

## Command Mode

Interface configuration

## Examples

The following example configures a static route that allows packets in DLCI 100 and transmits packets out over DLCI 200 on interface serial 2:

```
frame-relay route 100 interface Serial2 200
```

The following example illustrates the commands you enter for a complete configuration that includes two static routes for PVC switching between interface serial 1 and interface serial 2:

```
interface Serial1
no ip address
encapsulation frame-relay
keepalive 15
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 100 interface Serial2 200
frame-relay route 101 interface Serial2 201
clockrate 2000000
```

# frame-relay short-status

To instruct the network server to request the short status message from the switch (see Version 2.3 of the joint *Frame Relay Interface* specification), use the **frame-relay short-status** interface configuration command. Use the **no** form of this command to override the default

**frame-relay short-status**
**no frame-relay short-status**

## Syntax Description

These commands have no keywords or arguments.

## Default

To request the full status message

## Command Mode

Interface command

## Example

The following example returns the interface to the default state of requesting full status messages.

```
interface serial 0
no frame-relay short-status
```

# frame-relay switching

Use the **frame-relay switching** global configuration command to enable PVC switching on a Frame Relay DCE or an NNI. Use the **no** form of this command to disable switching.

**frame-relay switching**
**no frame-relay switching**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Global configuration

## Usage Guidelines

This command must be added to the configuration file before configuring the routes.

## Example

The following example shows the simple command that is entered in the configuration file before the Frame Relay configuration commands to enable switching:

```
frame-relay switching
```

# show frame-relay ip tcp header-compression

To display statistics and TCP/IP header compression information for the interface, use the **show frame-relay ip tcp header-compression** EXEC command.

**show frame-relay ip tcp header-compression**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show frame-relay ip tcp header-compression** command:

```
DLCI 200          Link/Destination info: ip 131.108.177.200
Interface Serial0:
Rcvd:     40 total, 36 compressed, 0 errors
          0 dropped, 0 buffer copies, 0 buffer failures
Sent:     0 total, 0 compressed
          0 bytes saved, 0 bytes sent
Connect:  16 rx slots, 16 tx slots, 0 long searches, 0 misses, 0% hit ratio
          Five minute miss rate 0 misses/sec, 0 max misses/sec
```

Table 9-2 describes the fields shown in the display.

**Table 9-2 Show Frame-Relay IP TCP Header-Compression Field Descriptions**

| Field | Description |
|---|---|
| **Rcvd** | |
| total | Sum of compressed and uncompressed packets received. |
| compressed | Number of compressed packets received. |
| errors | Number of errors caused by errors in the header fields (version, total length, or IP checksum). |
| dropped | Number of packets discarded. Seen only after line errors. |
| buffer copies | Number of times that a new buffer was needed to put the uncompressed packet in. |
| buffer failures | Number of times that a new buffer was needed but was not obtained. |
| **Sent** | |
| total | Sum of compressed and uncompressed packets sent. |
| compressed | Number of compressed packets sent. |
| bytes saved | Number of bytes reduced because of the compression. |
| bytes sent | Actual number of bytes transmitted. |

| Field | Description |
|---|---|
| **Connect** | |
| rx slots, tx slots | Number of states allowed over one TCP connection. A state is recognized by a source address, a destination address, and an IP header length. |
| long searches | Number of times that the connection ID in the incoming packet was not the same as the previous one that was processed. |
| misses | Number of times that a matching entry was not found within the connection table and a new entry had to be entered. |
| hit ratio | Percentage of times that a matching entry was found in the compression tables and the header was compressed. |
| Five minute miss rate | Miss rate computed over the most recent 5 minutes and the maximum per-second miss rate during that period. |

# show frame-relay lmi

Use the **show frame-relay lmi** EXEC command to display statistics about the Local Management Interface (LMI).

**show frame-relay lmi** [*type number*]

## Syntax Description

| | |
|---|---|
| *type* | (Optional) Interface type; serial only. |
| *number* | (Optional) Interface number. |

## Command Mode

EXEC

## Usage Guidelines

Enter the command without arguments to obtain statistics about all Frame Relay interfaces.

## Sample Displays

The following is sample output from the **show frame-relay lmi** command when the interface is a DTE:

```
Router# show frame-relay lmi

LMI Statistics for interface Serial1 (Frame Relay DTE) LMI TYPE = ANSI
  Invalid Unnumbered info 0             Invalid Prot Disc 0
  Invalid dummy Call Ref 0              Invalid Msg Type 0
  Invalid Status Message 0              Invalid Lock Shift 0
  Invalid Information ID 0              Invalid Report IE Len 0
  Invalid Report Request 0              Invalid Keep IE Len 0
  Num Status Enq. Sent 9                Num Status msgs Rcvd 0
  Num Update Status Rcvd 0              Num Status Timeouts 9
```

The following is sample output from the **show frame-relay lmi** command when the interface is an NNI:

```
Router# show frame-relay lmi

LMI Statistics for interface Serial3 (Frame Relay NNI) LMI TYPE = CISCO
  Invalid Unnumbered info 0             Invalid Prot Disc 0
  Invalid dummy Call Ref 0              Invalid Msg Type 0
  Invalid Status Message 0              Invalid Lock Shift 0
  Invalid Information ID 0              Invalid Report IE Len 0
  Invalid Report Request 0              Invalid Keep IE Len 0
  Num Status Enq. Rcvd 11               Num Status msgs Sent 11
  Num Update Status Rcvd 0              Num St Enq. Timeouts 0
  Num Status Enq. Sent 10               Num Status msgs Rcvd 10
  Num Update Status Sent 0              Num Status Timeouts 0
```

Table 9-3 describes significant fields shown in the output.

**Table 9-3 Show Frame-Relay LMI Field Descriptions**

| Field | Description |
|---|---|
| LMI TYPE = | Signaling or LMI specification: CISCO, ANSI, or ITU-T. |
| Invalid Unnumbered info | Number of received LMI messages with invalid unnumbered information field. |
| Invalid Prot Disc | Number of received LMI messages with invalid protocol discriminator. |
| Invalid dummy Call Ref | Number of received LMI messages with invalid dummy call references. |
| Invalid Msg Type | Number of received LMI messages with invalid message type. |
| Invalid Status Message | Number of received LMI messages with invalid status message. |
| Invalid Lock Shift | Number of received LMI messages with invalid lock shift type. |
| Invalid Information ID | Number of received LMI messages with invalid information identifier. |
| Invalid Report IE Len | Number of received LMI messages with invalid Report IE Length. |
| Invalid Report Request | Number of received LMI messages with invalid Report Request. |
| Invalid Keep IE Len | Number of received LMI messages with invalid Keep IE Length. |
| Num Status Enq. Rcvd | Number of LMI status inquiry messages received. |
| Num Status msgs Sent | Number of LMI status messages sent. |
| Num Status Update Sent | Number of LMI update status messages sent. |
| Num Status Enq. Sent | Number of LMI status inquiry messages sent. |
| Num Status msgs Received | Number of LMI status messages received. |
| Num Status Update Rcvd | Number of LMI update status messages received. |
| Num Status Timeouts | Number of times the status message was not received within the keepalive timer. |
| Num Status Enq. Timeouts | Number of times the status enquiry message was not received within the T392 DCE timer. |

# show frame-relay map

To display the current map entries and information about the connections, use the **show frame-relay map** EXEC command.

**show frame-relay map**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show frame-relay map** command:

```
Router# show frame-relay map

Serial 1 (administratively down): ip 131.108.177.177
dlci 177 (0xB1,0x2C10), static,
broadcast,
CISCO
TCP/IP Header Compression (inherited), passive (inherited)
```

Table 9-4 describes significant fields shown in the display.

**Table 9-4 Show Frame-Relay Map Field Descriptions**

| Field | Description |
|---|---|
| Serial 1 (administratively down) | Identifies a Frame Relay interface and its status (up or down). |
| ip 131.108.177.177 | Destination IP address. |
| dlci 177 (0xB1,0x2C10) | DLCI that identifies the logical connection being used to reach this interface. This value is displayed in three ways: its decimal value (177), its hexadecimal value (0xB1), and its value as it would appear on the wire (0x2C10). |
| static | Indicates whether this is a static or dynamic entry. |
| CISCO | Indicates the encapsulation type for this map; either CISCO or IETF. |
| TCP/IP Header Compression (inherited), passive (inherited) | Indicates whether the TCP header compression characteristics were inherited from the interface or were explicitly configured for the IP map. |

## Related Command

**show frame-relay pvc**

# show frame-relay pvc

To display statistics about PVCs for Frame Relay interfaces, use the **show frame-relay pvc** EXEC command.

**show frame-relay pvc** [*type number* [*dlci*]]

## Syntax Description

| | |
|---|---|
| *type* | (Optional) Interface type. |
| *number* | (Optional) Interface number. |
| *dlci* | (Optional) One of the specific DLCI numbers used on the interface. Statistics for the specified PVC display when a DLCI is also specified. |

## Command Mode

EXEC

## Usage Guidelines

To obtain statistics about PVCs on all Frame Relay interfaces, use this command with no arguments.

When the interface is configured as a DCE and the DLCI usage is SWITCHED, the value displayed in the PVC STATUS field is determined by the status of outgoing interfaces (up or down) and status of the outgoing PVC. (The status of the outgoing PVC is updated in the local management interface (LMI) message exchange). PVCs terminated on a DCE interface use the status of the interface to set the PVC STATUS.

If the outgoing interface is a tunnel, the PVC status is determined by what is learned from the tunnel.

If an LMI status report indicates that a PVC is not active, then it is marked as inactive. A PVC is marked as deleted if it is not listed in a periodic LMI status message.

In the case of a hybrid DTE switch, the PVC status on the DTE side is determined by the PVC status reported by the external Frame Relay network through the LMI.

Congestion control mechanisms are currently not supported, but the switch passes Forward Explicit Congestion Notification (FECN) bits, Backward Explicit Congestion Notification (BECN) bits, and Discard Eligibility (DE) bits unchanged from ingress to egress points in the network.

## Sample Display

The following is sample output from the **show frame-relay pvc** command:

```
Router# show frame-relay pvc

   PVC Statistics for interface Serial1 (Frame Relay DCE)

   DLCI = 100, DLCI USAGE = SWITCHED, PVC STATUS = ACTIVE

     input pkts 0             output pkts 0            in bytes 0
     out bytes 0              dropped pkts 0           in FECN pkts 0
     in BECN pkts 0           out FECN pkts 0          out BECN pkts 0
     in DE pkts 0             out DE pkts 0
     pvc create time 0:03:03 last time pvc status changed 0:03:03
     Num Pkts Switched 0
```

```
DLCI = 101, DLCI USAGE = SWITCHED, PVC STATUS = INACTIVE

  input pkts 0             output pkts 0            in bytes 0
  out bytes 0              dropped pkts 0           in FECN pkts 0
  in BECN pkts 0           out FECN pkts 0          out BECN pkts 0
  in DE pkts 0             out DE pkts 0
  pvc create time 0:02:58 last time pvc status changed 0:02:58
  Num Pkts Switched 0

DLCI = 102, DLCI USAGE = SWITCHED, PVC STATUS = DELETED
  input pkts 0             output pkts 0            in bytes 0
  out bytes 0              dropped pkts 0           in FECN pkts 0
  in BECN pkts 0           out FECN pkts 0          out BECN pkts 0
  in DE pkts 0             out DE pkts 0
  pvc create time 0:02:58 last time pvc status changed 0:02:58
  Num Pkts Switched 0
```

Table 9-5 describes the fields shown in the display.

**Table 9-5 Show Frame-Relay PVC Field Descriptions**

| Field | Description |
|---|---|
| DLCI | One of the Data Link Connection Identifier (DLCI) numbers for the PVC. |
| DLCI USAGE | Lists SWITCHED when the router is used as a switch, or LOCAL when the router is used as a DTE. |
| PVC STATUS | Status of the PVC: ACTIVE, INACTIVE, or DELETED. |
| input pkts | Number of packets received on this PVC. |
| output pkts | Number of packets sent on this PVC. |
| in bytes | Number of bytes received. |
| out bytes | Number of bytes sent. |
| dropped pkts | Number of packets dropped by the router. |
| in FECN pkts | Number of packets received with the FECN bit set. |
| in BECN pkts | Number of packets received with the BECN bit set. |
| out FECN pkts | Number of packets sent with the FECN bit set. |
| out BECN pkts | Number of packets sent with the BECN bit set. |
| in DE pkts | Number of DE packets received. |
| out DE pkts | Number of DE packets sent. |
| pvc create time | Time the PVC was created. |
| last time pvc status changed | Time the PVC changed status (active to inactive). |
| Num Pkts Switched | Number of packets switched within the router; this PVC is the source PVC. |

# show frame-relay route

Use the **show frame-relay route** EXEC command to display all configured Frame Relay routes, along with their status.

**show frame-relay route**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show frame-relay route** command:

```
Router# show frame-relay route

    Input Intf      Input Dlci      Output Intf     Output Dlci  Status
    Serial1         100             Serial2         200          active
    Serial1         101             Serial2         201          active
    Serial1         102             Serial2         202          active
    Serial1         103             Serial3         203          inactive
    Serial2         200             Serial1         100          active
    Serial2         201             Serial1         101          active
    Serial2         202             Serial1         102          active
    Serial3         203             Serial1         103          inactive
```

Table 9-6 describes significant fields shown in the output.

**Table 9-6 Show Frame-Relay Route Field Descriptions**

| Field | Description |
|---|---|
| Input Intf | Input interface and unit. |
| Input Dlci | Input DLCI number. |
| Output Intf | Output interface and unit. |
| Output Dlci | Output DLCI number. |
| Status | Status of the connection: active or inactive. |

# show frame-relay traffic

Use the **show frame-relay traffic** EXEC command to display the router's global Frame Relay statistics since the last reload.

**show frame-relay traffic**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show frame-relay traffic** command:

```
Router# show frame-relay traffic

Frame Relay statistics:
ARP requests sent 14, ARP replies sent 0
ARP request recvd 0, ARP replies recvd 10
```

Information shown in the display is self-explanatory.

# show interfaces serial

Use the **show interfaces serial** EXEC command to display information about a serial interface. When using the Frame Relay encapsulation, use the **show interfaces serial** command to display information about the multicast DLCI, the DLCIs used on the interface, and the LMI DLCI used for the Local Management Interface.

**show interfaces serial** *number*

## Syntax Description

| | |
|---|---|
| *number* | Interface number. |

## Command Mode

EXEC

## Usage Guidelines

The multicast DLCI and the local DLCI can be set using the **frame-relay multicast-dlci** and the **frame-relay local-dlci** commands, or provided through the Local Management Interface. The status information is taken from the LMI, when active.

Sample Displays

The following is sample output from the **show interfaces serial** command for a serial interface with the CISCO LMI enabled:

```
Router# show interface serial 1

Serial1 is up, line protocol is down
  Hardware is MCI Serial
  Internet address is 131.108.174.48, subnet mask is 255.255.255.0
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 246/255, load 1/255
  Encapsulation FRAME-RELAY, loopback not set, keepalive set (10 sec)
  LMI enq sent  2, LMI stat recvd 0, LMI upd recvd 0, DTE LMI down
  LMI enq recvd 266, LMI stat sent  264, LMI upd sent  0
  LMI DLCI 1023  LMI type is CISCO  frame relay DTE
  Last input 0:00:04, output 0:00:02, output hang never
  Last clearing of "show interface" counters 0:44:32
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
     307 packets input, 6615 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 input packets with dribble condition detected
     266 packets output, 3810 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets, 0 restarts
     178 carrier transitions
```

The display shows the statistics for the LMI as the number of status inquiry messages sent (LMI sent), the number of status messages received (LMI recvd), and the number of status updates received (upd recvd). See the *Frame Relay Interface* specification for additional explanations of this output.

The following is sample output from the **show interfaces** command for a serial interface with the ANSI LMI enabled:

```
Router# show interface serial 1
Serial1 is up, line protocol is down
  Hardware is MCI Serial
  Internet address is 131.108.174.48, subnet mask is 255.255.255.0
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 249/255, load 1/255
  Encapsulation FRAME-RELAY, loopback not set, keepalive set (10 sec)
  LMI enq sent  4, LMI stat recvd 0, LMI upd recvd 0, DTE LMI down
  LMI enq recvd 268, LMI stat sent  264, LMI upd sent  0
  LMI DLCI 0  LMI type is ANSI Annex D  frame relay DTE
  Last input 0:00:09, output 0:00:07, output hang never
  Last clearing of "show interface" counters 0:44:57
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
Five minute output rate 0 bits/sec, 0 packets/sec
     309 packets input, 6641 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 input packets with dribble condition detected
     268 packets output, 3836 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets, 0 restarts
     180 carrier transitions
```

Each display provides statistics and information about the type of LMI configured, either CISCO for the Cisco LMI type, ANSI for the ANSI T1.617 Annex D LMI type, or ITU-T for the ITU-T Q.933 Annex A LMI type. See the description for the **show interfaces** command for a description of the other fields displayed by this command.

## Related Command

A dagger (†) indicates that the command is documented in another chapter.

**show interfaces**†

# Index

# INDEX

## Symbols



## Numerics



## A

# B

## C

## D

## E

## F

## G



## H

# I

## K



## L

## M

## N

## O

## P

## Q



## R

# S

# T

## U



## V

## Z