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 are described in the "System Image, Microcode Image, and Configuration File Load Commands" 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.

  — You can assign passwords (and encrypt them) to restrict access to router terminal lines, login connections, or privileged EXEC mode.

  — You can establish Terminal Access Controller Access Control System (TACACS) protection for network servers that have shared access.

  — You can restrict login connections to specific users with a username authentication system.

  — You can control access on serial interfaces with Challenge Handshake Authentication Protocol (CHAP).

  — You can create access lists to filter traffic to and from specific destinations. Subsequent chapters that describe the routing protocols define access lists.

  — You can create security labels for Internet Protocol (IP) datagrams using the Internet Protocol Security Option (IPSO), as described in the " IP Commands" 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 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 detail on all **debug** commands, see the *Debug Command Reference* publication.

- Performance management

  To manage system performance, you need to monitor and set utilization thresholds 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 with such Cisco features as priority and custom queuing.

- Accounting management

  Accounting management allows you to track individual and group user utilization of network resources. You can then reallocate resources as needed.

For system management configuration tasks and examples, refer to the "Managing the System" chapter of the *Router Products Configuration Guide*.

# access-class

To apply a basic IP access list to a line, use the **access-class** line configuration command. To remove an access class, use the **no** form of the command.

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

## Syntax Description

| | |
|---|---|
| *list* | Identifies a specific standard IP access list (1 to 99). |
| **in** | Indicates an incoming connection, such as a virtual terminal connection. |
| **out** | Indicates an outgoing Telnet connection. |

## Default

None

## Command Mode

Line configuration

## Example

In the following example, the standard IP access list number 5 is assigned to incoming connections for virtual terminal line 3:

```
line vty 3
access-class 5 in
```

## Related Command

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

**access-list** †

# 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 buffers** command to return the buffers to their default size.

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

## Syntax Description

| | |
|---|---|
| **small** | Small buffer size. |
| **middle** | Medium buffer size. |
| **big** | Big buffer size. |
| **large** | Large buffer size. |
| **huge** | Huge buffer size. |
| **permanent** | Number of permanent buffers that the system tries to allocate. Permanent buffers are normally not deallocated 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 should be allocated when the system is reloaded. This 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 the 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

It is normally not necessary to adjust these parameters; do so only after consulting with technical support personnel. Improper settings could adversely impact system performance.

On the Cisco 4000, when building the receive rings for the serial and Ethernet interfaces, if a buffer request fails (that is, there isn't enough of that buffer size left in the pool), the interface is marked as down and the initialization is abandoned at that point. The interface will later initialize as more buffers are created to fill the demand. The configuration where this problem is most noticeable is the 1E4T configuration. The Serial 3 interface could take as long as 5 minutes before that interface would be usable.

However, buffer pool allocation is a user tunable parameter. The buffer pool to tune depends on the type of encapsulation used by the interfaces. Correspondingly, the ring size changes with the size of the buffer required. The mapping between buffer and ring size on the Cisco 4000 listed in Table 1-1.

**Table 1-1 Mapping between Buffer and Ring Size**

| Maximum Transmission Unit (MTU) | Receive Ring Size |
|---|---|
| MTU < 1524 | 32 |
| 1524 < MTU < 5024 | 8 |
| 5024 < MTU < 18024 | 4 |

See the examples that follow for specific examples of allocating buffer sizes for the Cisco 4000.

## Examples

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

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

On a Cisco 4000 1E4T box using HDLC encapsulation, there are five receive rings, each of 32 entries. The cache size is 32 buffers. The MTU for this sort of encapsulation is below 1524 bytes (the same as for Ethernet) which means that you must use buffers from the “big” pool. The basic number of “big” buffers required is (5 + 1) * 32 = 192. Adding a bit of “comfort” space, the following command can then be used:

```
buffers big permanent 200
```

This will increase the permanent buffer pool allocation for big buffers to 200.

On a Cisco 4000 6T box, using X.25 encapsulation, there are six receive rings, each with eight entries, plus a cache ring of eight entries. The MTU for this sort of encapsulation is below 5024 bytes but above 1524, so you must use buffers from the “large” pool. The basic number of “large” buffers required is (6 + 1) * 8 = 56. Adding a bit of “comfort” space, the following command can then be used:

```
buffers large permanent 60
```

This will increase the permanent buffer pool allocation for big buffers to 60.

A general guideline is to boot the box, check for whichever buffer pool is depleted, and increase that one. The above examples are just approximate figures for the various configurations.

## 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

## 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 Cisco 7000 system calendar, use the **calendar se**t 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). |

## Default

None

## Command Mode

EXEC

## Usage Guidelines

Once you set the Cisco 7000 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:

```
Router# calendar set 13:32:00 23 July 1993
```

## Related Commands

**clock read-calendar**
**clock set**
**clock summer-time**
**clock timezone**
**clock update-calendar**

# clock calendar-valid

To configure the Cisco 7000 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

None

## 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 the Cisco 7000 system clock, use the **clock read-calendar** EXEC command.

**clock read-calendar**

## Syntax Description

This command has no arguments or keywords.

## Default

None

## Command Mode

EXEC

## Usage Guidelines

When the Cisco 7000 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). |

## Default

None

## 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 pm on July 23, 1993:

```
Router# 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** *name* **recurring** [*week day month hh:mm week day month hh:mm* [*offset*]]
**clock summer-time** *name* **date** *date month year hh:mm date month year hh:mm* [*offset*]
**clock summer-time** *name* **date** *month date year hh:mm month date year hh:mm* [*offset*]
**no clock summer-time**

## Syntax Description

| | |
|---|---|
| *name* | 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** *name* **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** *name hours* [*minutes*]
**no clock timezone**

## Syntax Description

| | |
|---|---|
| *name* | 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:

```
router(config)# clock timezone PST -8
```

## Related Commands

**calendar set**
**clock set**
**clock summer-time**
**show clock**

# clock update-calendar

To set the Cisco 7000 calendar from the system clock, use the **clock update-calendar** EXEC command.

**clock update-calendar**

## Syntax Description

This command has no arguments or keywords.

## Default

None

## 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 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**

# 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 enable last-resort** global configuration 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

## 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 assign a password for the privileged command level, use the **enable password** global configuration command. The commands **enable password** and **enable-password** are synonymous.

**enable password** *password*

## Syntax Description

| | |
|---|---|
| *password* | Case-sensitive character string that specifies the line password prompted for in response to the EXEC command **enable**. The first character cannot be a number. The string can contain any alphanumeric characters, including spaces, up to 80 characters. You cannot specify the *password* in the format *number-space-anything*. The space after the number causes problems. |

## Default

None

## Command Mode

Global configuration

## Usage Guidelines

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

If the **enable** password is not set and the line 0 (console) password is set, it is possible to enter privileged command mode in two ways: either without having to enter a password at the console terminal, or if you are using any other line, by entering the console line password when prompted.

## Example

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

```
enable password secretword
```

## Related Commands

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

**login** †
**login tacacs** †
**password** †

# 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 enable use-tacacs** command to disable TACACS verification.

**enable use-tacacs**
**no enable use-tacacs**

## 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 the Extended TACACS, it also will pass any already-existing UNIX user identification code to the server.

**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.

## 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
```

# 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 Internet address of the host to be used as a syslog server. |

## Default

None

## 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

The default logging device is the console; all messages are displayed on the console unless otherwise specified. To log messages to an internal buffer, use the **logging buffered** global configuration command. The **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

Display 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 1-2 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.

The EXEC command **show logging** displays the addresses and levels associated with the current logging setup, as well as any other logging statistics.

**Table 1-2 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 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* | See Table 1-3 for the *facility-type* keywords. |

## Default

**local7**

## Command Mode

Global configuration

## Usage Guidelines

**Table 1-3 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 1-2. |

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

**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. 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

Log messages to the console

## Command Mode

Global configuration

## Example

The following example shows how to direct error messages to the console terminal only:

```
no logging on
```

# 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 1-2. |

## 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 1-2 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**

# 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**} *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. |
| *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 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 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

None

## 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.

## Example

The following example sets authentication key 10 to aNiceKey:

```
ntp authentication-key 10 md5 aNiceKey
```

**Note** When this command is written to NVRAM, the key is encrypted so that it is not displayed when the configuration is viewed.

## Related Commands

**ntp authenticate**
**ntp peer**
**ntp server**
**ntp trusted-key**

# 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:

```
router(config)# ntp broadcastdelay 5000
```

## Related Commands

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

**ntp broadcast** †
**ntp broadcast client** †

# ntp clock-period

As NTP compensates for the error in the system clock, it keeps track of the correction factor for this error. The system will automatically save 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

Do not enter this command; it is documented for informational purposes only. The system will automatically generate it as NTP determines the clock error and compensates.

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 disable NTP on a specific interface, use the **ntp disable** interface configuration command.

**ntp disable**

### Syntax Description

This command has no arguments or keywords.

### Default

NTP is enabled

### Command Mode

Interface configuration

### Example

The following example disables all NTP services on interface serial 0:

```
interface serial 0
ntp disable
```

# ntp master

To configure the router as an 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 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 server**
**ntp source**
**ntp authentication-key**

# 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 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 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 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

None

## 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 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 update-calendar**
**clock read-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 1-4 describes the test characters that the ping facility sends.

**Table 1-4 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 1-5 describes the default **ping** fields shown in the display.

**Table 1-5 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

User 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 1-6 describes the test characters that the ping facility sends.

**Table 1-6 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)

# 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* **default** {**high** | **medium** | **normal** | **low** }
**no priority-list** *list* **default** {**high** | **medium** | **normal** | **low**}

## Syntax Description

| | |
|---|---|
| *list* | 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

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

**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* **interface** *interface-type interface-number* {**high** | **medium** | **normal** | **low**}
**no priority-list** *list* **interface** *interface-type interface-number* {**high** | **medium** | **normal** | **low**}

## Syntax Description

| | |
|---|---|
| *list* | 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

None

## Command Mode

Global configuration

## Example

The following example sets any packet type entering on interface Ethernet 0 to a medium priority:

```
priority-list 3 interface ethernet 0 medium
```

## Related Commands

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

**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** command with the appropriate list number to remove an entry from the list.

**priority-list** *list* **protocol** *protocol-name* {**high** | **medium** | **normal** | **low**} *queue-keyword keyword-value*
**no priority-list** *list* **protocol**

## Syntax Description

| | |
|---|---|
| *list* | 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**, **ip**, **ipx**, **pad**, **rsrb**, **stun**, **vines**, **xns**, and **x25**. |
| **high** \| **medium** \| **normal** \| **low** | Priority queue level. |
| *queue-keyword keyword-value* | Possible keywords are **gt**, **lt**, **list**, **tcp**, and **udp**. See Table 1-7. |

## Default

None

## 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.

Use Table 1-7, Table 1-8, and Table 1-9 to configure the queuing priorities for your system.

**Table 1-7 Protocol Priority Queue Keywords and Values**

| Option | Description |
|---|---|
| **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 5-9 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 1-9 lists common UDP services and their port numbers. |

**Table 1-8 Common TCP Services and Their Port Numbers**

| Service | Port |
|---|---|
| Telnet | 23 |
| SMTP | 25 |

**Table 1-9 Common UDP Services and Their Port Numbers**

| Service | Port |
|---|---|
| Time service | 37 |
| IEN-116 name service | 42 |
| TACACS service | 49 |
| Domain Name Service | 53 |
| BOOTP server | 67 |
| BOOTP client | 68 |
| TFTP initial transfer | 69 |
| NetBIOS name service | 137 |
| BetBIOS datagram service | 138 |

**Note** The TCP and UDP ports listed in Table 1-8 and Table 1-9 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* 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

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

**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* **queue-limit** *high-limit medium-limit normal-limit low-limit*
**no priority-list** *list* **queue-limit**

## Syntax Description

| | |
|---|---|
| *list* | 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 1-10.

**Table 1-10 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

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

**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* **stun** {**high** | **medium** | **normal** | **low**} **address** *group-number address-number*
**no priority-list** *list* **stun** {**high** | **medium** | **normal** | **low**} **address** *group-number address-number*

## Syntax Description

| | |
|---|---|
| *list* | 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

None

## 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** †

# 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* **default** *queue-number*
**no queue-list** *list* **default** *queue-number*

## Syntax Description

| | |
|---|---|
| *list* | 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:

```
router(config)# queue-list 10 default 2
```

## Related Commands

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

**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* **interface** *interface-type interface-number queue-number*
**no queue-list** *list* **interface** *queue-number*

## Syntax Description

| | |
|---|---|
| *list* | 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

None

## 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.

```
router(config)# queue-list 4 interface tunnel 3 10
```

## Related Commands

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

**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* **protocol** *protocol-name queue-number queue-keyword keyword-value*
**no queue-list** *list* **protocol** *protocol-name*

## Syntax Description

| | |
|---|---|
| *list* | 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_router**, **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 1-7. |

## Default

None

## 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.

Use Tables 5-8, 5-9, and 5-10 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

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

**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* **queue** *queue-number* **byte-count** *byte-count-number*
**no queue-list** *list* **queue** *queue-number* **byte-count** *byte-count-number*

## Syntax Description

| | |
|---|---|
| *list* | 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:

```
router(config)# queue-list 9 queue 10 byte-count 1400
```

## Related Commands

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

**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* **queue** *queue-number* **limit** *limit-number*
**no queue-list** *list* **queue** *queue-number* **limit** *limit-numbe***r**

## Syntax Description

| | |
|---|---|
| *list* | 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:

```
router(config)# queue-list 5 queue 10 limit 40
```

## Related Commands

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

**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* **stun address** *group-number address-number*
**no queue-list** *list* **stun address** *group-number address-number*

## Syntax Description

| | |
|---|---|
| *list* | Number of the queue list. An integer 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

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

**custom-queue-list** †
**show queueing** †
**stun schema** †

# 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

The default is to allow high-priority operations 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

**no service exec-wait**

## 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 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

**no service telnet-zero-idle**

## 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**] or
**service timestamps** *type* **datetime** [**msec**] [**localtime**] [**show-timezone**]

**no service timestamps** [*type*]

## Syntax Description

| | |
|---|---|
| *type* | (Optional.) Type of message to timestamp: **debug** or **log**. |
| **uptime** | Timestamp with time since the system was rebooted. |
| **datetime** | Timestamp with the date and time. |
| **msec** | (Optional.) Add milliseconds to the date and time. |
| **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, 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 access-lists

To show the configured access lists for the system, use the **show access-lists** EXEC command.

**show access-lists**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Sample Display

The following sample output shows that standard IP, XNS, and Novell SAP access lists have been configured:

```
Router# show access-lists
Standard IP access list 99
    permit 0.0.0.55, wildcard bits 255.255.255.0
XNS access list 501
    permit 4 160.0800.0903.9906 0x0000 -1 0x0000
Novell SAP access list 1003
    deny  11.5500.2000.8014 4
Novell SAP access list 1004
    deny  11.5500.2000.8014 0
```

## Related Command

**access-list**

# show buffers

Use the **show buffers** EXEC command to display statistics for the buffer pools on the network server.

The network server has one pool of queuing elements and five pools of packet buffers of different sizes. For each pool, the network server keeps counts of the number of buffers outstanding, the number of buffers in the free list, and the maximum number of buffers allowed in the free list.

**show buffers** [*interface*]

## Syntax Description

| | |
|---|---|
| *interface* | (Optional.) Causes a search of all buffers that have been associated with that interface for longer than one minute. The contents of these buffers are printed to the screen. This option is useful in diagnosing problems where the input queue count on an interface is consistently nonzero. |

## Command Mode

EXEC

## Sample Display

The following is sample output from the **show buffers** command when the optional interface argument was omitted:

```
Router# show buffers

Buffer elements:
      250 in free list (250 max allowed)
      10816 hits, 0 misses, 0 created
Small buffers, 104 bytes (total 120, permanent 120):
      120 in free list (0 min, 250 max allowed)
      26665 hits, 0 misses, 0 trims, 0 created
Middle buffers, 600 bytes (total 90, permanent 90):
      90 in free list (0 min, 200 max allowed)
      5468 hits, 0 misses, 0 trims, 0 created
Big buffers, 1524 bytes (total 90, permanent 90):
      90 in free list (0 min, 300 max allowed)
      1447 hits, 0 misses, 0 trims, 0 created
Large buffers, 5024 bytes (total 0, permanent 0):
      0 in free list (0 min, 100 max allowed)
      0 hits, 0 misses, 0 trims, 0 created
Huge buffers, 12024 bytes (total 0, permanent 0):
      0 in free list (0 min, 30 max allowed)
      0 hits, 0 misses, 0 trims, 0 created

0 failures (0 no memory)
```

Table 1-11 describes significant fields shown in the display.

**Table 1-11 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. |
| 250 in free list (250 max allowed) | Maximum number of buffers that are available for allocation. |
| 10816 hits | Count of successful attempts to allocate a buffer when needed. |
| 0 misses | Count of buffer allocation attempts that resulted in growing the buffer pool in order to allocate a buffer. |
| 0 created | Count of new buffers created to satisfy buffer allocation attempts when the available buffers in the pool have already been allocated. |
| Small buffers | Blocks of memory used to hold network packets. The sizes of these buffers can vary as follows: small, middle, big, large and huge. |
| 104 bytes | Size of this type of buffer. |
| (total 120, permanent 120) | Total number of this type of buffer, and the number of these buffers that are permanent. |
| 0 trims | Count of buffers released to the system because they were not being used. |
| 0 created | Count of new buffers created in response to misses. |
| 0 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. |
| (0 no memory) | Number of failures because no memory was available to create a new buffer. |

# show calendar

To display the calendar hardware setting for the Cisco 7000, 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 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:

* indicates not authoritative

blank indicates authoritative

. 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 7000 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 1-12 describes significant fields shown in the display.

**Table 1-12 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). |
| 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 env
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 1-13 describes the **show environment** display fields on the Cisco 7000.

**Table 1-13 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 7000 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 when there are no warning conditions in the system:

```
env-chassis> show env 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)

Airflowtemperature 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.

```
Router# show env 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 1-14 describes the **show environment all** display fields.

**Table 1-14 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. |
| `Power Supply` | Type of power supply installed and its status. |
| `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. |

# 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) 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 12:40 p.m., so the environmental statistics at that time are displayed.

```
Router# show env 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 1-15 describes the **show environment last** display fields.

**Table 1-15 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. |

# 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.

```
target> show env 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 1-16 describes the **show environment table** display fields.

**Table 1-16 Show Environment Table Field Descriptions**

| Field | Description |
|---|---|
| SENSE (Voltage Parameters) | Voltage specification for DC line. |
| 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 1-17 describes significant fields shown in the display.

**Table 1-17 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 1-18 describes significant fields shown in the first section of the display.

**Table 1-18 Show Memory Field Descriptions—First Section**

| Field | Description |
|---|---|
| Head | Hexadecimal address of the head of the memory allocation chain. |
| Free List | 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 1-19 describes significant fields shown in the second section of the display.

**Table 1-19 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. |
| 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 IGS and 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 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 1-20 describes significant fields shown in the display.

**Table 1-20 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 1-21 describes significant fields shown in the display.

**Table 1-21 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 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
Router#
```

Table 1-22 shows the significant fields in the display.

**Table 1-22 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 processes

Use the **show processes** EXEC command to see 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: 0%/0%; one minute: 0%; five minutes: 0%
  PID Q T      PC Runtime (ms)    Invoked   uSecs   Stacks  TTY Process
    1 M T   40FD4         1736         58   29931  910/1000   0 Check heaps
    2 H E   9B49C           68        585     116  790/900    0 IP Input
    3 M E   AD4E6            0        737       0  662/1000   0 TCP Timer
    4 L E   AEBB2            0          2       0  896/1000   0 TCP Protocols
    5 M E   A2F9A            0          1       0  852/1000   0 BOOTP Server
    6 L E   4D2A0           16        127     125  876/1000   0 ARP Input
    7 L E   50C76            0          1       0  936/1000   0 Probe Input
    8 M E   63DA0            0          7       0  888/1000   0 MOP Protocols
    9 M E   86802            0          2       0 1468/1500   0 Timers
   10 M E   7EBCC          692         64   10812  794/1000   0 Net Background
   11 L E   83BBC            0          5       0  870/1000   0 Logger
   12 M T  11C454            0         38       0  574/1000   0 BGP Open
   13 H E   7F0E0            0          1       0  446/500    0 Net Input
   14 M T   436EA          540       3435     157  737/1000   0 TTY Background
   15 M E  11BA9C            0          1       0  960/1000   0 BGP I/O
   16 M E  11553A         5100       1367    3730 1250/1500   0 IGRP Router
   17 M E  11B76C           88       4200      20 1394/1500   0 BGP Router
   18 L T  11BA64          152      14650      10  942/1000   0 BGP Scanner
   19 M *       0          192         80    2400 1714/2000   0 Exec
```

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 1-23 describes significant fields shown in the two displays.

**Table 1-23 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 1-24 describes significant fields shown in the display.

**Table 1-24 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. |
| *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*.

# 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 1-25 describes significant fields shown in the display.

**Table 1-25 Show RIF Cache Display Field Description**

| 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. |

# 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
Router#
```

## 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-list

To set up an access list that determines which hosts can send requests to the network server, use the **snmp-server access-list** global configuration command. Use the **no snmp-server access-list** command to remove the specified access list.

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

## Syntax Description

| | |
|---|---|
| *list* | Integer from 1 to 99 that specifies an IP access list number. |

## Default

None

## Command Mode

Global configuration

## Usage Guidelines

The server ignores packets from hosts that the access list denies. The access list applies only to the global read-only SNMP agent configured with the command **snmp-server community**.

## Example

The following example allows the router to process only those packets from hosts passing access list 21:

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

## Related Command

**snmp-server community**

# snmp-server chassis-id

To provide a message line identifying the SNMP server serial number, use the **snmp-server chassis-id** global configuration command.

**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

None

## Command Mode

Global configuration

## Usage Guidelines

With Software Release 9.21, the Cisco MIB provides a new 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, use the **snmp-server community** global configuration command. This command enables SNMP server operation on the router. The **no snmp-server community** command removes the specified community string or access list.

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

## Syntax Description

| | |
|---|---|
| *string* | (Optional.) Community string that acts like a password and permits access to the SNMP protocol. |
| **RO** | (Optional.) Specifies read-only access. |
| **RW** | (Optional.) Specifies read-write access. |
| *list* | (Optional.) Integer from 1 to 99 that specifies an access list of Internet addresses that may use the community string. |

## Default

By default, an SNMP community string permits read-only access.

## Command Mode

Global configuration

## Example

The following example assigns the string comaccess to the SNMP allowing read-only access and specifies that Internet access list 4 can use the community string.

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

## Related Command

**snmp-server access-list**

# snmp-server contact

To set the system contact (syscontact) string, use the **snmp-server contact** global configuration command.

**snmp-server contact** *text*

## Syntax Description

| | |
|---|---|
| *text* | String that describes the system contact information. |

## Default

None

## 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 host

To specify the recipient of an SNMP trap operation, use the **snmp-server host** global configuration command. The **no snmp-server host** command removes the specified host.

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

## Syntax Description

| | |
|---|---|
| *address* | Name or Internet address of the host. |
| *community-string* | Password-like community string to send with the trap operation. |
| **snmp** | (Optional.) Enables the SNMP traps defined in RFC 1157. |
| **tty** | (Optional.) Enables Cisco enterprise-specific traps when a TCP connection closes. |

## Default

If neither the **snmp** or **tty** keywords are supplied, the default is to enable both 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.

**snmp-server location** *text*

## Syntax Description

| | |
|---|---|
| *text* | String that describes the system location information. |

## Default

None

## 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.

**snmp-server packetsize** *bytes*

## Syntax Description

| | |
|---|---|
| *bytes* | 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 queue-length

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

**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

## 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 snmp-server system-shutdown** option 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

By 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. This command enables the network server to send a trap message when it receives a packet with an incorrect community string. Use the **no snmp-server trap-authentication** command to remove message authentication.

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

## Syntax Description

This command has no arguments or keywords.

## Default

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

## Command Mode

Global configuration

## Usage Guidelines

The community string is checked before any access list that may be set, so it is possible to get spurious trap messages. 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.

## Example

The following example illustrates how to enter the command that establishes trap message authentication:

```
snmp-server trap-authentication
```

# 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

None

## 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**

# 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

The **tacacs-server authenticate** global configuration command requires a response from the network or router to indicate whether the user may perform the indicated action. Enter one of the keywords to specify the action (when a user makes TCP connection, for example).

**tacacs-server authenticate** {**connection** | **enable**}

## 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. |

## Default

None

## Command Mode

Global configuration

## Usage Guidelines

**Caution** If you use the **enable use-tacacs** command, you must also use **tacacs-server authenticate enable**, or else you will be locked out of the router.

## Example

The following example illustrates how to configure TACACS logins that authenticate user TCP connections:

```
tacacs-server authenticate connect
```

## Related Command

**enable use-tacacs**

# tacacs-server extended

To enable an extended TACACS mode, use the **tacacs-server extended** global configuration command. Use the **no tacacs-server extended** 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

## Example

The following is an example of how to enable 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 tacacs-server host** command deletes the specified name or address.

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

## Syntax Description

| | |
|---|---|
| *name* | Name or Internet address of the host. |

## Default

None

## Command Mode

Global configuration

## Example

The following example illustrates how to specify 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 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.

## Example

The following example illustrates how to force 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. The terminal user, however, 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).

**tacacs-server notify** {**connection** | **enable** | **logout**}

## Syntax Description

| | |
|---|---|
| **connection** | Specifies that a message be transmitted when a user makes a TCP connection. |
| **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. |

## Default

None

## Command Mode

Global configuration

## 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.

**tacacs-server optional-passwords**

## Syntax Description

This command has no arguments or keywords.

## Default

None

## 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.

## Example

The following example illustrates how to configure 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 tacacs-server retransmit** 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 tacacs-server timeout** 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:

```
Router# 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:

```
Router# 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**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

**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.

## Example

The following example illustrates how to begin the memory test:

```
Router# 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 1-26 describes the fields shown in the display.

**Table 1-26 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 1-27 describes the fields that are unique to the extended trace sequence, as shown in the display.

**Table 1-27 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. |

| Field | Description |
|---|---|
| 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 1-28 describes the characters that can appear in **trace** output.

**Table 1-28 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

*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

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 1-29 describes the fields shown in the display.

**Table 1-29 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 1-30 describes the characters that can appear in **trace** output.

**Table 1-30 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

Networks that cannot support a TACACS service still may wish to use a username-based authentication system at login. The software supports these needs by providing a local **username** global configuration command.

**username** *name* [**nopassword** | **password** *encryptiontype* **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. |
| **nopassword** | (Optional.) No password is required for this user to log in. This is usually most useful in combination with the **autocommand** keyword. |
| **password** | Specifies a possibly encrypted password for this username. |
| *encryptiontype* | (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**