CHAPTER 5

System Management Commands

This chapter describes the commands used to manage the communication server and its performance on the network. In general, system or network management falls into the following categories.

- 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.

- 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 communication server 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, create access lists to filter traffic to and from specific destinations, and create security labels for Internet Protocol (IP) datagrams using the Internet Protocol Security Option (IPSO). As well, you can restrict login connections to specific users with user authentication support.

- 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*.

- 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 *Communication Server 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

access-list

access-list

To create an access list providing restriction specifications, use the **access-list** global configuration command. To remove the access list, use the **no** form of the command.

access-list
no access-list

Syntax Description

See the appropriate protocol-specific **access-list** command in this manual. See Table 1-1 for a list of protocols.

Default

None

Command Mode

Global configuration

Usage Guidelines

You can create access lists for the protocols listed in Table 1-1. See the appropriate chapter in this manual for protocol-specific **access-list** commands.

Table 1-1 Summary of Numerical Ranges

Protocol	Range
IP	1–99
Extended IP	100–199
Ethernet type code	200–299
Ethernet address	700–799
Novell	800–899
Extended Novell	900–999
Novell SAP	1000–1099

Example

The following example displays the global configuration access list ranges:

```
cs(config)# access-list ?
  <1-99>       IP standard access list
  <100-199>    IP extended access list
  <1000-1099>  IPX SAP access list
  <1100-1199>  Extended 48-bit MAC address access list
  <200-299>    Protocol type-code access list
 <700-799>    48-bit MAC address access list
  <800-899>    IPX standard access list
  <900-999>    IPX extended access list
```

Related Command

access-class

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 {**big | huge | large | middle | small**} {**initial | max-free | min-free | permanent**} *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.

Example

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

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

Related Command

show buffers

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 source, 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:

```
cs# 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 the **clock summer-time** global configuration command.

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.
recurring	Recurring summer time.
date	Absolute summer time.
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

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:

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

Related Commands

clock set
clock summer-time
show clock

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 the user to enter provileged EXEC mode by entering the privileged command level password.
succeed	Allows the user to enter privileged EXEC mode without requiring a password.

Default

Default action is to not allow the user to enter privileged EXEC mode if the TACACS server does not respond.

Command Mode

Global configuration

Example

In the following example, the communication server is configured to allow access to a user even if the TACACS server does not respond.

```
enable last-resort succeed
```

Related Command

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

enable †

enable password

To assign a password for the privileged EXEC 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.

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 †
password †

enable use-tacacs

To enable use of the TACACS to determine whether a user can access the privileged EXEC 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.

When using TACACS to authenticate enable requests, you are prompted for both a username and a password. Both of these are sent to the TACACS server in an XTA_ENABLE packet. The TACACS server must then decide how to process the three pieces of information (username, password, and request-type=XTA_ENABLE). The server may verify the username and password and then check whether that user is authorized to "enable," or the server may use a single specially designed enable account, verify the password, and ignore the username. The latter behavior is implemented in the TACACS server that is provided by using source code form. Our server uses the account name "$enable$".

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 communication server.

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

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** 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 communication server.

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.

The host name is used in prompts and default configuration filenames. The **setup** facility also prompts for a host name at startup.

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

logging *host*
no logging *host*

Syntax Description

host	Name or IP address of the host to be used as a syslog server.

Default

None

Command Mode

Global configuration

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 names.

Default

debugging

Command Mode

Global configuration

Usage Guidelines

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 emergencies, emergencies only:

```
logging console emergencies
```

Related Command

logging facility

logging facility

To configure the syslog facility, 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 shows the keywords that can be used for the argument *facility-type*.

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

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 names listed in Table 1-2.

Default

debugging

Command Mode

Global configuration

Example

The following example specifies that only errors or higher 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 names listed in Table 1-2.

Default

debugging

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

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).
md5	MD5 authentication.
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.

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

Example

The following example sets authentication key 10 to aNiceKey.

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

Related Commands

ntp authenticate
ntp peer
ntp server
ntp trusted-key

ntp broadcast

To specify that a specific interface should send Network Time Protocol (NTP) broadcast packets, use the **ntp broadcast** interface configuration command. Use the **no** form of the command to disable this capability.

ntp broadcast [**version** *number*]
no ntp broadcast

Syntax Description

version *number*	Number from 1 to 3 indicating the NTP version.

Default

Disabled

Command Mode

Interface configuration

Examples

In the following example, interface Ethernet0 is configured to send NTP version 2 packets:

```
cs(config)# interface ethernet0
cs(config-if)# ntp broadcast version 2
```

Related Commands

ntp broadcast client
ntp broadcastdelay

ntp broadcast client

To allow the system to receive NTP broadcast packets on an interface, use the **ntp broadcast client** command. Use the **no** form of the command to disable this capability.

ntp broadcast client
no ntp broadcast client

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Interface configuration

Usage Guidelines

Use this command to allow the system to listen to broadcast packets on an interface-by-interface basis.

Example

In the following example, the communication server synchronizes to NTP packets broadcasted on interface Ethernet1:

```
cs(config)# interface ethernet1
cs(config-if)# ntp broadcast client
```

Related Commands

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

ntp broadcast
ntp broadcastdelay

ntp broadcastdelay

To set the estimated round-trip delay between the communication server 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 communication server 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 communication server and the broadcast client is set to 5000 microseconds:

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

Related Commands

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 communication server 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 communication server 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 communication server is configured as an NTP master clock to which peers can synchronize:

```
ntp master 10
```

ntp peer

To configure the communication server'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 communication server 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 communication server'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 communication server 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 communication server 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 *number*
no ntp trusted-key *number*

Syntax Description

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

ping (privileged)

Use the **ping** (packet internet groper) privileged EXEC command to diagnose basic network connectivity on IP and Novell networks.

ping [*protocol*] {*host* | *address*}

Syntax Description

protocol	(Optional.) Protocol keyword, one of **ip** and **novell.**
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 communication server.

Example

After you enter the **ping** command in privileged mode, the system prompts for one of the following protocol keywords—**ip** and **novell**. 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.

```
cs# 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 **ip and novell**. 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 communication server. 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 IP and Novell networks.

ping [*protocol*] {*host* | *address*}

Syntax Description

protocol	(Optional.) Protocol keyword, one of **ip** and **novell**.
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 communication server 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:

```
cs> 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: **arp**, **compressedtcp**, **ip**, **ipx**, **pad**, 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 for a description of the keywords and values.

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.

Option	Description
list *list-number*	Assigns traffic priorities according to a specified list when used with IP and IPX. The *list-number* argument is the IP access list number as specified by the **access-list** global configuration command.
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 1-8 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
TFTP	69
NFS	2049
SNMP	161
RPC	111
DNS	53

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 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
!
```

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 †

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:

```
cs(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.

```
cs(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: **arp**, **compressedtcp**, **ip**, **ipx**, **pad**, 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 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
!
```

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:

```
cs(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:

```
cs(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 †

service decimal-tty

To specify that line numbers be displayed and interpreted as decimal numbers rather than octal numbers, use the **service decimal-tty** global configuration command. The **no service decimal-tty** command restores the default.

service decimal-tty
no service decimal-tty

Syntax Description

These commands have no arguments or keywords.

Default

Octal line numbers on the ASM-CS. Decimal numbers on the 500-CS.

Command Mode

Global configuration

Example

The following example shows how to display decimal rather than octal line numbers.

```
service decimal-tty
```

service exec-wait

To delay the startup of the EXEC on noisy lines, use the **service exec-wait** global configuration command. Use the **no service exec-wait** command to disable this feature.

service exec-wait
no service exec-wait

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Global configuration

service nagle

To set 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

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-zeroidle

To set the TCP window to zero (0) when the connection is idle, use the **service telnet-zeroidle** global configuration command. Use the **no service telnet-zeroidle** command to disable this feature.

service telnet-zeroidle
no service telnet-zeroidle

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Global configuration

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 and Novell SAP access lists have been configured:

```
cs# 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.

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

Usage Guidelines

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.

Sample Display

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

```
cs# 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 the specified 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 clock

To display the system clock, use the **show clock** EXEC command:

show clock [**detail**]

Syntax Description

detail (Optional.) Indicates the clock source (NTP) 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 (NTP), 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 communication server from causing peers to synchronize to itself when the communication servertime 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:

```
cs# show clock detail
15:29:03.158 PST Mon Mar 1 1993
Time source is NTP
communication server#
```

Related Commands

clock set
show calendar

show logging

Use the **show logging** EXEC command to display the state of logging (syslog).

show logging

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Usage Guidelines

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.

Sample Display

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

```
communication server# 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 1-12 describes significant fields shown in the display.

Table 1-12 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 communication server's memory, including memory free pool statistics.

show memory [**free**]

Syntax Description

free	(Optional.) Displays free memory statistics.

Command Mode

EXEC

Sample Displays

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

```
cs# 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*
cs#
```

The following is sample output from the **show memory free** command:

```
cs# 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 1-13 describes significant fields shown in the first section of the display.

Table 1-13 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)	Total amount of system memory.
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 1-14 describes significant fields shown in the second section of the display.

Table 1-14 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).

Field	Description
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.

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:

```
cs# 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
cs#
```

Table 1-15 describes significant fields shown in the display.

Table 1-15 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:

```
cs# 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 1-16 describes significant fields shown in the display.

Table 1-16 Show NTP Associatioins 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 timestamp.
rcv time	Receive timestamp.
xmt time	Transmit timestamp.
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:

```
cs# 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
cs#
```

Table 1-17 shows the significant fields in the display.

Table 1-17 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** | **memory**]

Syntax Description

cpu	(Optional.) Displays detailed CPU utilization statistics.
memory	(Optional.) Displays detailed memory utilization statistics.

Command Mode

EXEC

Sample Displays

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

```
cs# 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:

```
cs# 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 1-18 describes significant fields shown in the two displays.

Table 1-18 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.
5Sec	CPU utilization by task in last 5 seconds.
1Min	CPU utilization by task in last minute.
5Min	CPU utilization by task in last 5 minutes.

Following is a description of first line in the sample display: 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:

```
cs# 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 1-19 describes significant fields shown in the display.

Table 1-19 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 and Novell.

show protocols

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

For more information on the parameters or protocols displayed using this command, see the *Communication Server Configuration Guide*.

show rif

Use the **show rif** EXEC command to display the current contents of the Routing Information Field (RIF) cache.

show rif

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Sample Display

The following is a sample display from the **show rif** command:

```
cs# 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 communication server'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 1-20 describes significant fields shown in the display.

Table 1-20 Show RIF Cache Display Field Descriptions

Field	Description
Hardware Addr	Lists the MAC-level addresses.
How	Describes how the RIF has been learned. Possible values include a ring group (rg), or interface (TR).
Idle (min)	Indicates how long, in minutes, since the last response was received directly from this node.
Routing Information Field	Lists the RIF.

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:

```
cs# 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
cs#
```

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:

```
cs# 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
```

show tcp

Use the **show tcp** EXEC command to display the status of TCP Telnet connections.

show tcp [*line-number*]

Syntax Description

line-number	(Optional.) Absolute line number of the line for which you want to display Telnet connection status.

Command Mode

EXEC

Sample Display

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

```
cs# show tcp

con0 (console terminal), connection 1 to host MATHOM
Connection state is ESTAB, I/O status: 1, unread input bytes: 1
Local host: 192.31.7.18, 33537  Foreign host: 192.31.7.17, 23
Enqueued packets for retransmit: 0, input: 0, saved: 0
Event Timers (current time is 2043535532):
Timer:       Retrans   TimeWait    AckHold    SendWnd   KeepAlive
Starts:           69          0         69          0           0
Wakeups:           5          0          1          0           0
Next:     2043536089          0          0          0           0
iss: 2043207208 snduna: 2043211083  sndnxt: 2043211483    sndwnd: 1344
irs: 3447586816 rcvnxt: 3447586900  rcvwnd:       2144 delrcvwnd:   83
RTTO: 565 ms, RTV: 233 ms, KRTT: 0 ms, minRTT: 68 ms, maxRTT: 1900 ms
ACK hold: 282 ms
Datagrams (max data segment is 536 bytes):
Rcvd: 106 (out of order: 0), with data: 71, total data bytes: 83
Sent: 96 (retransmit: 5), with data: 92, total data bytes: 4678
```

Table 1-21 describes the following lines of output shown in the display:

```
con0 (console terminal), connection 1 to host MATHOM
Connection state is ESTAB, I/O status: 1, unread input bytes: 1
Local host: 192.31.7.18, 33537  Foreign host: 192.31.7.17, 23
Enqueued packets for retransmit: 0, input: 0, saved: 0
```

Table 1-21 Show TCP Field Descriptions—First Section of Output

Field	Description
con0	Identifying number of the line.
(console terminal)	Location string.
connection 1	Number identifying the TCP connection.
to host MATHOM	Name of the remote host to which the connection has been made.

Field	Description
Connection state is ESTAB	A connection progresses through a series of states during its lifetime. These states follow in the order in which a connection progresses through them. • LISTEN—Waiting for a connection request from any remote TCP and port. • SYNSENT—Waiting for a matching connection request after having sent a connection request. • SYNRCVD—Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTAB—Indicates an open connection; data received can be delivered to the user. This is the normal state for the data transfer phase of the connection. • FINWAIT1—Waiting for a connection termination request from the remote TCP or an acknowledgment of the connection termination request previously sent. • FINWAIT2—Waiting for a connection termination request from the remote TCP host. • CLOSEWAIT—Waiting for a connection termination request from the local user. • CLOSING—Waiting for a connection termination request acknowledgment from the remote TCP host. • LASTACK—Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP host. • TIMEWAIT—Waiting for enough time to pass to be sure the remote TCP host has received the acknowledgment of its connection termination request. • CLOSED—Indicates no connection state at all. For more information, see RFC 793, *Transmission Control Protocol Functional Specification.*
I/O status: 1	Number describing the current internal status of the connection.
unread input bytes: 1	Number of bytes that the lower-level TCP processes have read, but the higher level TCP processes have not yet processed.
Local host: 192.31.7.18	IP address of the network server.
33537	Local port number, as derived from the following equation: *line-number* + (512 * *random-number*). (The line number uses the lower nine bits; the other bits are random.)
Foreign host: 192.31.7.17	IP address of the remote host to which the TCP connection has been made.
23	Destination port for the remote host.

Field	Description
Enqueued packets for retransmit: 0	Number of packets waiting on the retransmit queue. These are packets on this TCP connection that have been sent but have not yet been acknowledged by the remote TCP host.
input: 0	Number of packets that are waiting on the input queue to be read by the user.
saved: 0	Number of received out-of-order packets that are waiting for all packets comprising the message to be received before they enter the input queue. For example, if packets 1, 2, 4, 5, and 6 have been received, packets 1 and 2 would enter the input queue, and packets 4, 5, and 6 would enter the saved queue.

The following line of output shows the current time according to the system clock of the local host.

```
Event Timers (current time is 2043535532):
```

The time shown is the number of milliseconds since the system started.

The following lines of output display the number of times that various local TCP timeout values were reached during this connection. In this example, the communication server retransmitted 69 times because it received no response from the remote host, and it transmitted an ACK many more times because there was no data on which to piggyback.

```
Timer:       Retrans   TimeWait    AckHold    SendWnd   KeepAlive
Starts:           69          0         69          0           0
Wakeups:           5          0          1          0           0
Next:     2043536089          0          0          0           0
```

Table 1-22 describes the fields in the preceding lines of output.

Table 1-22 Show TCP Field Descriptions—Second Section of Output

Field	Description
Timer:	This line of output indicates the names of the timers in the display.
Starts:	The number of times the timer has been started during this connection.
Wakeups:	The wakeups row of the KeepAlives column shows how many keepalives have been transmitted without receiving any response. (This field is reset to zero when a response is received.)
Next:	The system clock setting that will trigger the next time this timer will go off.
Retrans	The Retransmission timer is used to time TCP packets which have not been acknowledged and are waiting for retransmission.
TimeWait	The TimeWait timer is used to ensure that the remote system receive a request to disconnect a session.
AckHold	The Acknowledgment timer is used to delay the sending of acknowledgments to the remote TCP in an attempt to reduce network use.
SendWnd	The Send Window is used to ensure that there is no closed window due to a lost TCP acknowledgment.
KeepAlive	The KeepAlive timer is used to control the transmission of test messages to the remote TCP to ensure that the link has not been broken without the local TCP's knowledge.

The following lines of output display the sequence numbers that TCP uses to ensure sequenced, reliable transport of data. The communication server and remote host each use these sequence numbers for flow control and to acknowledge receipt of datagrams. Table 1-23 describes the specific fields in these lines of output.

```
iss: 2043207208 snduna: 2043211083  sndnxt: 2043211483    sndwnd: 1344
irs: 3447586816 rcvnxt: 3447586900  rcvwnd:       2144 delrcvwnd:   83
```

Table 1-23 Show TCP Field Descriptions—Sequence Number

Field	Description
iss: 2043207208	Initial send sequence number.
snduna: 2043211083	Last send sequence number the communication server has sent but has not received an acknowledgment for.
sndnxt: 2043211483	Sequence number the communication server will send next.
sndwnd: 1344	TCP window size of the remote host.
irs: 3447586816	Initial receive sequence number.
rcvnxt: 3447586900	Last receive sequence number the communication server has acknowledged.
rcvwnd: 2144	Communication server's TCP window size.
delrcvwnd: 83	Delayed receive window—data the communication server has read from the connection, but has not yet subtracted from the receive window the communication server has advertised to the remote host. The value in this field gradually increases until it is larger than a full-sized packet, at which point it is applied to the rcvwnd field.

The following lines of output display values that the communication server uses to keep track of transmission times so that TCP can adjust to the network it is using. Table 1-24 describes the fields in the following line of output.

```
RTTO: 565 ms, RTV: 233 ms, KRTT: 0 ms, minRTT: 68 ms, maxRTT: 1900 ms
ACK hold: 282 ms
```

Table 1-24 Show TCP Field Descriptions—Line Beginning with RTTO

Field	Description
RTTO: 565 ms	Round trip timeout.
RTV: 233 ms	Variance of the round trip time.
KRTT: 0 ms	New round trip timeout (using the Karn algorithm). This field separately tracks the round trip time of packets that have been retransmitted.
minRTT: 68 ms	Smallest recorded round trip timeout (hard wire value used for calculation).
maxRTT: 1900 ms	Largest recorded round trip timeout.
ACK hold: 282 ms	Time the communication server will delay an acknowledgment in order to piggyback data on it.

For more information on these fields, refer to "Round Trip Time Estimation," P. Karn & C. Partridge, ACM SIGCOMM-87, August 1987.

Table 1-25 describes the fields in the following lines of output.

```
Datagrams (max data segment is 536 bytes):
Rcvd: 106 (out of order: 0), with data: 71, total data bytes: 83
Sent: 96 (retransmit: 5), with data: 92, total data bytes: 4678
```

Table 1-25 Show TCP Field Descriptions—Last Section of Output

Field	Description
Rcvd: 106 (out of order: 0)	Number of datagrams the local host has received during this connection (and the number of these datagrams that were out of order).
with data: 71	Number of these datagrams that contained data.
total data bytes: 83	Total number of bytes of data in these datagrams.
Sent: 96 (retransmit: 5)	Number of datagrams the local host sent during this connection (and the number of these datagrams that had to be retransmitted).
with data: 92	Number of these datagrams that contained data.
total data bytes: 4678	Total number of bytes of data in these datagrams.

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 communication server 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 communication server. 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 IP 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 string (syscontact), 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 IP 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 cisco.com
```

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 communication server 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-type interface-number*
no snmp-server trap-source

Syntax Description

interface-type	Interface from which the SNMP trap originates.
interface-number	Interface number from which the SNMP trap originates.

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.

Example

The following example specifies that the IP address for interface Ethernet 0 is the source for all traps on the communication server:

```
snmp-server trap-source ethernet 0
```

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 communication server 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

Use the **tacacs-server authenticate** global configuration command to determine if the network or communication server may perform the indicated action.

tacacs-server authenticate {**connection** [**always**] | **enable** | **slip** [**always**] [**access-lists**]}

If you use the **enable use-tacacs** command, you must also usethe **tacacs-server authenticate enable** command; otherwise, you will be locked out of the communication server.

Syntax Description

connection	Configures a required response when a user makes a TCP connection.
enable	Configures a required response when a user enters the **enable** command.
slip	Configures a required response when a user starts a SLIP or PPP session.
always	(Optional.) Performs authentication even when a user is not logged in. This option can be used with the **connection** or **slip** keywords.
access-lists	(Optional.) Requests and installs SLIP and PPP access lists. This option only applies to SLIP or PPP sessions, and can be used only with the **slip** keyword.

Command Mode

Global configuration

Usage Guidelines

When using TACACS to authenticate enable requests, you are prompted for both a username and a password. Both of these are sent to the TACACS server in an XTA_ENABLE packet. The TACACS server must then decide how to process the three pieces of information (username, password, and request-type=XTA_ENABLE). The server may verify the username and password and then check whether that user is authorized to “enable,” or the server may use a single specially designed enable account, verify the password, and ignore the username. The latter behavior is implemented in the TACACS server that is provided by using source code form. Our server uses the account name “$enable$”.

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 communication server.

Example

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

```
tacacs-server authenticate connect always
```

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 IP 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

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

login tacacs†

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 five 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 communication server software will search the list of TACACS server hosts before giving up, use the **tacacs-server retransmit** global configuration command. The communication server 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
```

trace (privileged level)

Use the **trace** EXEC command to discover the routes the communication server's packets will actually take when traveling to their destination.

trace [*destination*]

Syntax Description

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. IP is the protocol that can be used.

Type

EXEC

Usage Guidelines

The **trace** command works by taking advantage of the error messages generated by communication servers 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 communication server 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 communication server 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 communication servers, the IP **trace** command might 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:

```
communication server# 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 communication server in the path to the host.
DEBRIS.CISCO.COM	Host name of this communication server.
131.108.1.61	Internet address of this communication server.
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.

```
communication server# 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 communication server to use as a source address for the probes. The communication server will normally pick what it feels is the best source address to use.
Numeric display	The default is to have both a symbolic and numeric display; however, you can suppress the symbolic display.
Timeout in seconds	The number of seconds to wait for a response to a probe packet. The default is 3 seconds.
Probe count	The number of probes to be sent at each TTL level. The default count is 3.
Minimum Time to Live [1]	The TTL value for the first probes. The default is 1, but it can be set to a higher value to suppress the display of known hops.
Maximum Time to Live [30]	The largest TTL value that can be used. The default is 30. The **trace** command terminates when the destination is reached or when this value is reached.
Port Number	The destination port used by the UDP probe messages. The default is 33434.
Loose, Strict, Record, Timestamp, Verbose	IP header options. You may 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 communication servers (or end nodes) will process the options.
Loose Source Routing	Allows you to specify a list of nodes that must be traversed when going to the destination.
Strict Source Routing	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.

Char	Description
N	Network unreachable.
U	Port unreachable.
H	Host unreachable.

Related Command

trace (user level)

trace ip (user level)

Use the **trace** EXEC command to discover the IP routes the communication server's packets will actually take when traveling to their destination.

trace ip *destination*

Syntax Description

destination	Destination address or host name on the command line. The default parameters for the appropriate protocol are assumed and the tracing action begins. The protocol that can be used is IP.

Type

EXEC

Usage Guidelines

The **trace** command works by taking advantage of the error messages generated by communication servers 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 communication server 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 communication server 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 communication servers, the IP **trace** command might 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:

```
cs# 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 communication server in the path to the host.
DEBRIS.CISCO.COM	Host name of this communication server.
131.108.1.61	IP address of this communication server.
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 level)

username

Networks that cannot support a TACACS service still might want 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	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	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	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 communication server or the remote device. The secret is encrypted when it is stored on the local communication server. 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	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	The access list number.
autocommand	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	The command string.
noescape	Prevents a user from using an escape character on the host to which that user is connected.
nohangup	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 communication server communicates with and requires authentication from. The remote device must have a **username** entry for the local communication server. This entry must have the same password as the local communication server'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 communication server communicates with from which it requires authentication, add a **username** entry.

Note To enable the local communication server 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 communication server.

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 communication server, 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