

System Management Commands

This chapter describes the commands used to manage the communication server system and its performance on the network.

For system management configuration tasks and examples, refer to the chapter entitled “Managing the System” in the *Access and Communication Servers Configuration Guide*.

aaa accounting

To enable AAA accounting of requested services for billing or security purposes when using TACACS+, use the **aaa accounting** global configuration command. Use the **no** form of this command to disable accounting.

```
aaa accounting {system | network | connection | exec | command level} {start-stop |  
wait-start | stop-only} tacacs+  
no aaa accounting {system | network | connection | exec | command level}
```

Syntax Description

system	Accounting is performed for all system-level events not associated with users, such as reloads.
network	Accounting is run for all network-related service requests, including SLIP, PPP, PPP NCPs, and ARAP.
connection	Accounting is run for outbound Telnet and rlogin.
exec	Accounting is run for Execs (user shells). This may return user profile information such as autocommand information.
command	Accounting is run for all commands at the specified privilege level.
<i>level</i>	The command level that should be accounted. Valid entries are 0-15.
start-stop	A start record accounting notice is sent at the beginning of a process and a stop record is sent at the end of a process. The start accounting record is sent in the background. The requested user process begins regardless of whether or not the start accounting record was received by the accounting server.
wait-start	As in start-stop , both a start and a stop accounting record are sent to the accounting server. However, if you use the wait-start keyword, the requested user service will not begin until the start accounting record is acknowledged. A stop accounting record will also be sent.
stop-only	A stop record accounting notice is sent at the end of the requested user process.

Default

AAA accounting is not enabled.

Command Mode

Global configuration

Usage Guideline

The **aaa accounting** command allows you to set start/stop accounting for any or all of the listed functions in the Syntax Description for this command. For minimal accounting control, issue the **stop-only** command, which sends a stop record accounting notice at the end of the requested user

process. For additional accounting control, you can issue the **start-stop** command, where TACACS+ sends a start accounting notice at the beginning of the requested process and a stop accounting notice at the end of the process. You can further control access and accounting by issuing the **wait-start** command, which ensures that the start notice is received by the TACACS+ server before granting the user's process request. Accounting is only done to the TACACS+ server.

Note This command, along with **aaa authorization**, replaces the **tacacs-server authenticate** command in previous versions of TACACS, and can only be used with AAA/TACACS+.

Examples

In the following example, accounting is set for outbound Telnet and rlogin, and both a start and stop accounting notice is sent to the TACACS+ server:

```
aaa accounting connection start-stop tacacs+
```

In the following example, accounting is set for privilege level 15 commands, with a wait-start restriction:

```
aaa accounting command 15 wait-start tacacs+
```

Related Commands

aaa new-model
aaa authorization

aaa authentication arap

To enable an AAA authentication method for ARA users while using TACACS+, use the **aaa authentication arap** command. Use the **no** form of the command to disable this authentication.

```
aaa authentication arap {default | list-name} method1 [...[method4]]
no aaa authentication arap {default | list-name} method1 [...[method4]]
```

Syntax Description

default	Makes the listed methods that follow this argument the default list of methods used when a user logs in.
<i>list-name</i>	A character string used to name the following list of authentication methods tried when a user logs in.
<i>method</i>	One of the methods described in Table 5-1.

Default

If the **default** list is not set, only the local user database is checked. This has the same effect as issuing the following command:

```
aaa authentication arap default local
```

Command Mode

Global configuration

Usage Guideline

The list names and default that you set using the **aaa authentication arap** command are used with the **arap authentication** command. These lists can contain up to four authentication methods that will be used when a user tries to log in with ARA.

Create a list by entering the **aaa authentication arap list-name method** command, where *list-name* is any character string used to name this list, such as MIS-access. The *method* keyword refers to the list of methods the authentication algorithm will try, in the given sequence. You can enter up to four methods, which are described in Table 5-1.

To create a default list that will be used if no list is specified in the **arap authentication** command, use the **default** keyword followed by the methods you wish to be used in default situations.

The additional methods of authentication will only be used if the previous method returns an error, not if it fails.

Use the **write terminal** command to view lists of authentication methods.

Table 5-1 AAA Authentication ARAP Method Descriptions

Method	Description
if-needed	Will not authenticate if the user has already been authenticated on a TTY line.
line	Uses the line password for authentication.

Method	Description
local	Uses the local username database for authentication.
tacacs+	Uses TACACS+ authentication.

Note This command cannot be used with TACACS or XTACACS.

Examples

The following example creates a list called MIS-access, which first tries TACACS+ authentication and then none:

```
aaa authentication arap MIS-access tacacs+ none
```

The following example creates the same list, but sets it as the default list, which will be used for all arap authentications if no other list is specified:

```
aaa authentication arap default tacacs+ none
```

Related Commands

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

aaa authentication local-override

aaa new-model

arap authentication†

aaa authentication enable default

To enable AAA authentication to determine if a user can access the privileged command level with TACACS+, use the **aaa authentication enable default** command. Use the **no** form of the command to disable this authorization method.

```
aaa authentication enable default method1 [...method4]  
no aaa authentication enable default method1 [...method4]
```

Syntax Description

method At least one and up to four of the methods described in Table 5-2.

Default

If the **default** list is not set, the action will be to check only the enable password. This has the same effect as issuing the following command:

```
aaa authentication enable default enable
```

On the console, the enable password is used if it exists. If no password is set, the process will succeed anyway.

Command Mode

Global configuration

Usage Guideline

Use the **aaa authentication enable default** command to create a series of authentication methods that are used to determine if a user can access privileged command level. You can specify up to four authentication methods. Method keywords are described in Table 5-2. The additional methods of authentication will only be used if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

If a default authentication routine is not set for a function, the default is none—no authentication is performed. Use the **write terminal** command to view currently configured lists of authentication methods.

Table 5-2 AAA Authentication Enable Default Method Descriptions

Method	Description
enable	Uses the enable password for authentication.
line	Uses the line password for authentication.
none	Uses no authentication.
tacacs+	Uses TACACS+ authentication.

Note This command cannot be used with TACACS or XTACACS.

Example

The following example creates an authentication list that will first try to contact a TACACS+ server. If no server can be found, then AAA will try to use the enable password. If this also returns an error (because no enable password is configured on the server), the user will be allowed access with no authentication.

```
aaa authentication enable default tacacs+ enable none
```

Related Commands

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

aaa authentication local-override

aaa new-model

aaa authorization

enable password†

aaa authentication login

To set AAA authentication at login when using TACACS+, use the **aaa authentication login** global configuration command. Use the **no** form of the command to disable AAA authentication.

```
aaa authentication login { default | list-name } method1 [...method4]
no aaa authentication login { default | list-name } method1 [...method4]
```

Syntax Description

default	Makes the listed authentication methods that follow this argument the default list of methods used when a user logs in.
<i>list-name</i>	A character string used to name the following list of authentication methods tried when a user logs in.
<i>method</i>	At least one and up to four of the methods described in Table 5-3.

Default

If the **default** list is not set, only the local user database is checked. This has the same effect as issuing the following command:

```
aaa authentication login default local
```

Note On the console, login will succeed without any authentication checks if **default** is not set.

Command Mode

Global configuration

Usage Guideline

The default and optional list names that you create with the **aaa authentication login** command are used with the **login authentication** command.

Create a list by entering the **aaa authentication list-name method** command, where *list-name* is any character string used to name this list, such as MIS-access. The *method* keyword refers to the list of methods the authentication algorithm tries, in the given sequence. Method keywords are described in Table 5-3.

To create a default list that is used if no list is assigned to a line with the **login authentication** command, use the default argument followed by the methods you want in default situations.

The additional methods of authentication is only be used if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

If authentication is not specifically set for a line, the default is to deny access—no authentication is performed. Use the **write terminal** command to view currently configured lists of authentication methods.

Table 5-3 AAA Authentication Login Method Descriptions

Method	Description
enable	Uses the enable password for authentication.
line	Uses the line password for authentication.
local	Uses the local username database for authentication.
none	Uses no authentication.
tacacs+	Uses TACACS+ authentication.

Note This command cannot be used with TACACS or XTACACS.

Example

The following example creates an AAA authentication list called MIS-access. This authentication will first try to contact a TACACS+ server. If no server is found, TACACS+ will return an error and AAA will try to use the enable password. If this also returns an error (because no enable password is configured on the server), the user is allowed access with no authentication.

```
aaa authentication login MIS-access tacacs+ enable none
```

The following example creates the same list, but sets it as the default list that will be used for all login authentications if no other list is specified:

```
aaa authentication login default tacacs+ enable none
```

Related Commands

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

aaa authentication local-override

aaa new-model

login authentication†

aaa authentication local-override

To have the communication server check the local user database for authentication before attempting another form of authentication, use the **aaa authentication local-override** command. Use the **no** form of the command to disable the override.

aaa authentication local-override
no aaa authentication local-override

Syntax Description

This command has no arguments or keywords.

Default

Override is disabled.

Command Mode

Global configuration

Usage Guideline

This command is useful when you want to configure an override to the normal authentication process for certain personnel such as system administrators.

When this override is set, the user is always be prompted for the username. The system then checks to see if the entered username corresponds to a local account. If the username does not correspond to one in the local database, login proceeds with the methods configured using other **aaa** commands (such as **aaa authentication login**). Note that when using this command, the first prompt is fixed as

Username :

Example

The following example enables aaa authentication override:

```
aaa authentication local-override
```

Related Commands

aaa authentication arap
aaa authentication enable default
aaa authentication login
aaa new-model
aaa authentication ppp

aaa authentication ppp

To specify one or more AAA authentication methods for use on serial interfaces running PPP when using TACACS+, use the **aaa authentication ppp** command. Use the **no** form of the command to disable authentication.

```
aaa authentication ppp { default | list-name } method1 [...method4]]
no aaa authentication ppp { default | list-name } method1 [...method4]]
```

Syntax Description

default	Makes the listed authentication methods that follow this argument the default list of methods used when a user logs in.
<i>list-name</i>	A character string used to name the following list of authentication methods tried when a user logs in.
<i>method</i>	At least one and up to four of the methods described in Table 5-4.

Default

If the **default** list is not set, the action will be to check only the local user database. This has the same effect as issuing the following command:

```
aaa authentication ppp default local
```

Command Mode

Global configuration

Usage Guideline

The lists that you create using the **aaa authentication ppp** command are used with the **ppp authentication** command. These lists contain up to four authentication methods that will be used when a user tries to log in to the serial interface.

Create a list by entering the **aaa authentication ppp list-name method** command, where *list-name* is any character string used to name this list, such as MIS-access. The *method* keyword refers to the list of methods the authentication algorithm tries, in the given sequence. You can enter up to four methods. Method keywords are described in Table 5-4.

The additional methods of authentication are only used if the previous method returns an error, not if it fails. Specify **none** as the final method in the command line to have authentication succeed even if all methods return an error.

If authentication is not specifically set for a function, the default is none—no authentication is performed. Use the **write terminal** command to view lists of authentication methods.

Table 5-4 AAA Authentication PPP Method Descriptions

Method	Description
if-needed	Does not authenticate if user has already been authenticated on a TTY line.
local	Uses the local username database for authentication.
none	Uses no authentication.
tacacs+	Uses TACACS+ authentication.

Note This command cannot be used with TACACS or XTACACS.

Example

The following example creates an AAA authentication list called MIS-access for serial lines that use PPP. This authentication first tries to contact a TACACS+ server. If this returns an error, the user is allowed access with no authentication.

```
aaa authentication MIS-access ppp tacacs+ none
```

Related Commands

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

aaa authentication local-override

aaa new-model

ppp authentication†

aaa authorization

To set parameters that restrict a user's network access based on TACACS+ authorization, use the **aaa authorization** command. To disable authorization for a function, use the **no** form of the command.

```
aaa authorization {network | connection | exec | command level} methods
no aaa authorization {network | connection | exec | command level}
```

Syntax Description

network	Authorization is run for all network-related service requests, including SLIP, PPP, PPP NCPs, and ARAP.
connection	Authorization is run for outbound Telnet and rlogin.
exec	Authorization is run to determine if the user is allowed to run an Exec shell. This may return user profile information such as autocommand information.
command	Authorization is run for all commands at the specified privilege level.
<i>level</i>	Specific command level that should be authorized. Valid entries are 0-15.

Default

Authorization is disabled for all actions (equivalent to the keyword *none*).

Command Mode

Global configuration

Usage Guideline

Use the **aaa authorization** command to create a list of one and up to four authorization methods that can be used when a user accesses the specified function. Table 5-5 lists the different authorization methods.

The additional methods of authorization are only used if the previous method returns an error, not if it fails. Specify **none** as the final method in the command line to have authorization succeed even if all methods return an error.

Table 5-5 AAA Authorization Keyword Descriptions

Keyword	Description
<i>methods</i>	<ul style="list-style-type: none"> • tacacs+—request authorization information from the TACACS+ server. • if-authenticated—allow the user to access the requested function if the user is authenticated. • none—no authorization is performed. • local—use the local database for authorization.

If authorization is not specifically set for a function, the default is none—no authorization is performed.

Note This command, along with **aaa accounting**, replaces the **tacacs-server** suite of commands in previous versions of TACACS.

Examples

The following example specifies that TACACS+-style authorization is used for all network-related requests. If this authorization method returns an error (if the TACACS+ server cannot be contacted), no authorization is performed, and the request is successful.

```
aaa authorization network tacacs+ none
```

The following example specifies that TACACS+-style authorization is run for level 15 commands. If this authorization method returns an error (if the TACACS+ server cannot be contacted), no authorization is performed, and the request succeeds.

```
aaa authorization command 15 tacacs+ none
```

Related Commands

aaa accounting
aaa new-model

aaa new-model

To enable the new AAA access control model that includes TACACS+, issue the **aaa new-model** global configuration command. Use the **no** form of the command to disable this functionality.

```
aaa new-model  
no aaa new-model
```

Syntax Description

This command has no arguments or keywords.

Default

AAA/TACACS+ is not enabled.

Command Mode

Global configuration

Usage Guideline

This command enables the new AAA access control system and TACACS+. If you initialize this functionality and later decide to use TACACS or XTACACS, issue the **no** version of this command and then enable the version of TACACS you want to use.

Example

The following example initializes AAA and TACACS+:

```
aaa new-model
```

Related Commands

```
aaa accounting  
aaa authentication arap  
aaa authentication enable default  
aaa authentication local-override  
aaa authentication login  
aaa authentication ppp  
aaa authorization
```

alias

To create a command alias, use the **alias** global configuration command. Use the **no alias** command to delete all aliases in a command mode or to delete a specific alias, and to revert to the original command syntax.

```
alias mode alias-name alias-command-line
no alias mode [alias-name]
```

Syntax Description

<i>mode</i>	Command mode of the original command and alias commands. See Table 5-6 for a list of options for this argument.
<i>alias-name</i>	Command alias.
<i>alias-command-line</i>	Original command syntax.

Defaults

Default aliases are in EXEC mode as follows:

Command Alias	Original Command
h	help
lo	logout
p	ping
r	resume
s	show
w	where

Command Mode

Global configuration

Usage Guidelines

You can use simple words as aliases or abbreviations. The aliases in the Default section are predefined. They can be turned off using the **no alias** command.

Table 5-6 shows the acceptable options for the *mode* argument in the **alias** global configuration command.

Table 5-6 Mode Argument Options

Argument Options	Mode
configuration	Global configuration
controller	Controller configuration
exec	EXEC
hub	Hub configuration

Argument Options	Mode
interface	Interface configuration
ipx-router	IPX router configuration
line	Line configuration
map-class	Map class configuration
map-list	Map list configuration
route-map	Route map configuration
router	Router configuration

See the summary of command modes in the user interface chapter in the *Access and Communication Configuration Guide* for more information about command modes.

When you use online help, command aliases are indicated by an asterisk (*), as follows:

```
cs#lo?
*lo=logout lock login logout
```

When you use online help, aliases that contain spaces (for example, “telnet device.cisco.com 25”) are displayed as follows:

```
cs# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
cs(config)# alias exec device-mail telnet device.cisco.com 25
cs(config)# end
cs#device-mail?
*device-mail="telnet device.cisco.com 25"
```

When you use online help, the alias is expanded and replaced with the original command, as shown in the following example with the “td” alias:

```
cs(config)#alias exec td trace device
cs(config)#^Z
cs#t?
*td="trace device" telnet terminal test tn3270
trace
```

To list only commands and omit aliases, begin your input line with a space. In the following example, the alias *td* is not shown, because there is a space before the **t?** command line.

```
cs# t?
telnet terminal test tn3270 trace
```

As with commands, you can use online help to display the arguments and keywords that can follow a command alias. In the following example, the alias **td** is created to represent the command **telnet device**. The **/debug** and **/line** switches can be added to **telnet device** to modify the command:

```
cs(config)# alias exec td telnet device
cs(config)# ^Z
cs#td ?
  /debug      Enable telnet debugging mode
  /line       Enable telnet line mode
  ...
  whois       Whois port
  <cr>

cs# telnet device
```

You must enter the complete syntax for the **alias** command. Partial syntax for aliases are not accepted. In the following example, the parser does not recognize the command *t* as indicating the alias *td*.

```
bones# t
% Ambiguous command: "t"
```

Example

In the following example, the alias *fixmyrt* is created for the EXEC-mode command **clear ip route 198.92.116.16**.

```
alias exec fixmyrt clear ip route 198.92.116.16
```

Related Command

show aliases

arap authentication

To enable TACACS+ authentication for ARA on a line, use the **arap authentication** command. Use the **no** form of the command to disable authentication for an ARA line.

```
arap authentication { default | list-name }  
no arap authentication { default | list-name }
```

Syntax Description

default Use the default list created with the **aaa authentication arap** command.

list-name Use the indicated list created with the **aaa authentication arap** command.

Default

ARAP authentication uses the default set with **aaa authentication arap** command. If no default has been set, the local user database is checked.

Command Mode

Line configuration

Usage Guideline

This command is a per-line command, and specifies the name of a list of AAA authentication methods to try at login. If no list is specified, the default list will be used (whether or not it is specified in the command line). Defaults and lists are created by using the **aaa authentication arap** command. Entering the **no** version of **arap authentication** has the same effect as entering the command with the **default** argument.

Before issuing this command, create a list of authentication processes by using the **aaa authentication arap** global configuration command.



Caution If you use a *list-name* that was not configured with the **aaa authentication arap** command, ARAP will be disabled on this line.

Example

The following example specifies that the TACACS+ authentication list called MIS-access is used on ARA line 7:

```
line 7  
  arap authentication MIS-access
```

Related Command

aaa authentication arap

buffers

Use the **buffers** global configuration command to make adjustments to initial buffer pool settings and to the limits at which temporary buffers are created and destroyed. Use the **no buffers** command to return the buffers to their default size.

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

Syntax Description

small	Small buffer size.
middle	Medium buffer size.
big	Big buffer size.
large	Large buffer size.
huge	Huge buffer size.
<i>interface-type</i>	Interface type of the interface buffer pool.
<i>interface-number</i>	Interface number of the interface buffer pool.
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.
<i>number</i>	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.

Buffer pool allocation is a user tunable parameter. The buffer pool to tune depends on the type of encapsulation used by the interfaces. Correspondingly, the ring size changes with the size of the buffer required.

Examples

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

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

In the following example, the permanent buffer pool allocation for big buffers is increased to 200. On a Cisco 2509 1E2T box using HDLC encapsulation, there are four receive rings, each of 32 entries. The cache size is 32 buffers. The MTU for this sort of encapsulation is below 1524 bytes (the same as for Ethernet) which means that you must use buffers from the “big” pool. The basic number of “big” buffers required is $(2 + 1) * 32 = 96$. Adding a bit of “comfort” space, the following command can then be used:

```
buffers big permanent 100
```

The above example shows approximate figures.

In the following example, the initial and permanent interface buffer pools are set to 100:

```
buffers ethernet 0 initial 100  
buffers ethernet 0 permanent 100
```

Related Commands

buffers huge size

show buffers

buffers huge size

Use the **buffers huge size** global configuration command to dynamically resize all huge buffers to the value you specify. Use the **no buffers huge size** command to restore the default buffer values.

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

Syntax Description

number Number of buffers to be allocated

Default

18024 buffers

Command Mode

Global configuration

Usage Guidelines

Use this command only after consulting with technical support personnel. The buffer size cannot be lowered below the default.

Example

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

```
buffers huge size 20000
```

Related Commands

buffers
show buffers

cdp enable

To enable CDP on an interface, use the **cdp enable** interface configuration command. Use the **no cdp enable** command to disable CDP on an interface.

cdp enable
no cdp enable

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Interface configuration

Usage Guidelines

CDP is enabled by default at the global level, but it must be enabled on each interface in order to send or receive CDP information.

Example

In the following example, CDP is enabled on Ethernet interface 0:

```
interface ethernet 0
 cdp enable
```

Related Command

cdp run

cdp holdtime

To specify the amount of time the receiving device should hold a CDP packet from your communication server before discarding it, use the **cdp holdtime** global configuration command. Use the **no cdp holdtime** command to revert to the default setting.

cdp holdtime *seconds*
no cdp holdtime

Syntax Description

seconds Specifies the hold time to be sent in the CDP update packets.

Default

180 seconds

Command Mode

Global configuration

Usage Guidelines

CDP packets are sent with time-to-live, or hold time, that is nonzero after an interface is enabled and a hold time of 0 immediately before an interface is idled down.

The CDP hold time must be set to a higher number of seconds than the time between CDP transmissions, which is set using the **cdp timer** command.

Example

In the following example, the CDP packets being sent from your device should be held by the receiving device for 60 seconds before being discarded. You might want to set the hold time lower than the default setting of 180 seconds if information about your device changes often and you want the receiving devices to purge this information more quickly.

```
cdp holdtime 60
```

Related Commands

cdp timer
show cdp

cdp run

To enable CDP on your communication server, use the **cdp run** global configuration command. Use the **no cdp run** command to disable CDP.

```
cdp run  
no cdp run
```

Syntax Description

This command has no arguments or keywords.

Default

Enabled

Command Mode

Global configuration

Usage Guidelines

CDP is enabled on your communication server by default, which means the communication server will receive CDP information. However, to receive CDP packets it must be enabled on interfaces, using the **cdp enable** interface configuration command.

Example

In the following example, CDP is disabled for the communication server.

```
no cdp run
```

Related Command

cdp enable

cdp timer

To specify how often your communication server will send CDP updates, use the **cdp timer** global configuration command. Use the **no cdp timer** command to revert to the default setting.

cdp timer *seconds*
no cdp timer

Syntax Description

<i>seconds</i>	Specifies how often your communication server will send CDP updates.
----------------	--

Default

60 seconds

Command Mode

Global configuration

Usage Guidelines

The trade-off with sending more frequent transmissions is providing up-to-date information versus using bandwidth more often.

Example

In the following example, CDP updates will be sent from your communication server every 80 seconds, less frequently than the default setting of 60 seconds. You might want to make this change if you are concerned about preserving bandwidth.

```
cdp timer 80
```

Related Commands

cdp holdtime
show cdp

clear cdp counters

To reset CDP traffic counters to zero (0) on your communication server, use the **clear cdp counters** privileged EXEC command.

clear cdp counters

Syntax Description

This command has no arguments or keywords.

Command Mode

Privileged EXEC

Example

In the following example, the CDP counters have been cleared. The **show cdp traffic** output shows that all of the traffic counters have been reset to zero (0).

```
cs# clear cdp counters
cs# show cdp traffic

CDP counters :
  Packets output: 0, Input: 0
  Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
  No memory: 0, Invalid packet: 0, Fragmented: 0
```

Related Commands

show cdp traffic

clear cdp table

clear cdp table

To clear the table that contains CDP information about neighbors, use the **clear cdp table** privileged EXEC command.

clear cdp table

Syntax Description

This command has no arguments or keywords.

Command Mode

Privileged EXEC

Example

In the following example, the CDP table is cleared. The output of the **show cdp neighbors** command shows that all information has been deleted from the table.

```
cs# clear cdp table

CDP-AD: Deleted table entry for neon.cisco.com, interface Ethernet0
CDP-AD: Deleted table entry for neon.cisco.com, interface Serial0
cs# show cdp neighbors

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP

Device ID          Local Intrfce    Holdtme    Capability Platform Port ID
```

Related Commands

clear cdp counters

show cdp neighbors

clock set

To manually set the system clock, use the **clock set** EXEC command.

clock set *hh:mm:ss day month year*

clock set *hh:mm:ss month day year*

Syntax Description

hh:mm:ss Current time in hours (military format), minutes, and seconds

day Current day (by date) in the month

month Current month (by name)

year Current year (no abbreviation)

Command Mode

EXEC

Usage Guidelines

Generally, if the system is synchronized by a valid outside timing mechanism, such as an NTP clock source, you need not set the system clock. Use this command if no other time sources are available. The time specified in this command is relative to the configured time zone.

Example

In the following example, the system clock is manually set to 1:32 p.m. on July 23, 1993:

```
clock set 13:32:00 23 July 1993
```

Related Commands

calendar set

clock read-calendar

clock summer-time

clock timezone

clock summer-time

To configure the system to switch to summer time (daylight savings time) automatically, use one of the formats of the **clock summer-time** global configuration command. Use the **no** form of this command to configure the communication server not to automatically switch to summer time.

```
clock summer-time zone recurring [week day month hh:mm week day month hh:mm [offset]]  
clock summer-time zone date date month year hh:mm date month year hh:mm [offset]  
clock summer-time zone date month date year hh:mm month date year hh:mm [offset]  
no clock summer-time
```

Syntax Description

<i>zone</i>	Name of the time zone (PDT, ...) to be displayed when summer time is in effect
<i>week</i>	Week of the month (1 to 5 or last)
<i>day</i>	Day of the week (Sunday, Monday ...)
<i>date</i>	Date of the month (1 to 31)
<i>month</i>	Month (January, February, ...)
<i>year</i>	Year (1993 to 2035)
<i>hh:mm</i>	Time (military format) in hours and minutes
<i>offset</i>	(Optional) Number of minutes to add during summer time (default is 60)

Default

Summer time is disabled. If **clock summer-time zone recurring** is specified without parameters, the summer time rules default to United States rules. Default of *offset* is 60.

Command Mode

Global configuration

Usage Guidelines

Use this command if you want to automatically switch to summer time (for display purposes only). Use the **recurring** form of the command if the local summer time rules are of this form. Use the **date** form to specify a start and end date for summer time if you cannot use the first form.

In both forms of the command, the first part of the command specifies when summer time begins, and the second part specifies when it ends. All times are relative to the local time zone. The start time is relative to standard time. The end time is relative to summer time. If the starting month is after the ending month, the system assumes that you are in the Southern Hemisphere.

Examples

In the following example, summer time starts on the first Sunday in April at 02:00 and ends on the last Sunday in October at 02:00:

```
clock summer-time PDT recurring 1 Sunday April 2:00 last Sunday October 2:00
```

If you live in a place where summer time does not follow the pattern in the first example, you could set it to start on October 12, 1993 at 02:00, and end on April 28, 1994 at 02:00, with the following example:

```
clock summer-time date 12 October 1993 2:00 28 April 1994 2:00
```

Related Commands

calendar set

clock timezone

clock timezone

To set the time zone for display purposes, use the **clock timezone** global configuration command. To set the time to Coordinated Universal Time (UTC), use the **no clock timezone** command.

clock timezone *zone hours* [*minutes*]
no clock timezone

Syntax Description

zone Name of the time zone to be displayed when standard time is in effect

hours Hours offset from UTC

minutes (Optional) Minutes offset from UTC

Default
UTC

Command Mode
Global configuration

Usage Guidelines

The system internally keeps time in UTC, so this command is used only for display purposes and when the time is manually set.

Example

In the following example, the time zone is set to Pacific Standard Time and is offset 8 hours behind UTC:

```
clock timezone PST -8
```

Related Commands

calendar set
clock set
clock summer-time
show clock

custom-queue-list

To assign a custom queue list to an interface, use the **custom-queue-list** interface configuration command. To remove a specific list or all list assignments, use the **no** form of this command.

custom-queue-list *list*
no custom-queue-list [*list*]

Syntax Description

list Number of the custom queue list you want to assign to the interface. An integer from 1 to 10.

Default

No custom queue list is assigned.

Command Mode

Interface configuration

Usage Guidelines

You can assign only one queue list per interface. Use this command in place of the **priority-list** command (not in addition to it). Custom queuing allows a fairness that is not provided with priority queuing. With custom queuing, you can control the interfaces' available bandwidth when it is unable to accommodate the aggregate traffic enqueued. Associated with each output queue is a configurable byte count, which specifies how many bytes of data should be delivered from the current queue by the system before the system moves on to the next queue. When a particular queue is being processed, packets are sent until the number of bytes sent exceeds the queue byte count or until the queue is empty.

Example

In the following example, custom queue list number 3 is assigned to serial interface 0:

```
interface serial 0
  custom-queue-list 3
```

Related Commands

queue-list default
queue-list interface
queue-list protocol
queue-list queue byte-count
queue-list queue limit
queue-list stun

downward-compatible-config

To have the access server try to generate a configuration that is compatible with an earlier Cisco IOS release, use the **downward-compatible-config** global configuration command. To remove this feature, use the **no** form of this command.

downward-compatible-config *version*
no downward-compatible-config

Syntax Description

version Cisco IOS Release number, not earlier than 10.2.

Default

Disabled

Command Mode

Global configuration

Usage Guidelines

In Cisco IOS Release 10.3, IP access lists changed format. Use this command to regenerate a configuration in a format prior to Release 10.3 if you are going to downgrade from a Release 10.3 or later to an earlier release. The earliest release this command accepts is 10.2.

When this command is configured, the router attempts to generate a configuration that is compatible with the specified version. Currently, this command affects only IP access lists.

Under some circumstances, the software might not be able to generate a fully backward-compatible configuration. In such a case, the software issues a warning message whenever it tries to write a configuration that is not downward compatible.

Example

The following example, the router will attempt to generate a configuration file compatible with Cisco IOS Release 10.2:

```
downward-compatible-config 10.2
```

Related Commands

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

access-list (extended)[†]
access-list (standard)[†]

enable last-resort

To specify what happens if the TACACS servers used by the **enable** command do not respond, use the **enable last-resort** global configuration command. The **no** form of this command restores the default.

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

Syntax Description

password Allows users to enable by entering the privileged command level password.

succeed Allows users to enable without further question.

Default

Default action is to fail.

Command Mode

Global configuration

Usage Guideline

The secondary authentication is used only if the first attempt failed. The second authentication will not occur if the first authentication was only unsuccessful.

Note This command is not used in AAA/TACACS+ and has been replaced by the **aaa authentication** suite of commands.

Example

In the following example, if the TACACS servers do not respond to the **enable** command, the user can enable by entering the privileged level password:

```
enable last-resort password
```

Related Command

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

enable †

enable password

Use the **enable password** global configuration command to set a local password to control access to various privilege levels. Use the **no** form of this command to remove the password requirement.

```
enable password [level level] {password}  
enable password [level level] {encryption-type encrypted-password}  
no enable password [level]
```

Syntax Description

<i>level level</i>	(Optional) Level for which the password applies. You can specify up to sixteen privilege levels, using numbers 0 through 15. Level 1 is normal EXEC-mode user privileges. If this argument is not specified, the privilege level defaults to 15 (traditional enable privileges). The same holds true for the no form of the command.
<i>password</i>	The enable password password as you would type it when enabling. This password should be different from the password created with the enable secret command. If you have the service password-encryption flag set, when the router displays the password for you later it will be displayed encrypted (an encrypted form of what you typed).
<i>encryption-type</i>	(Optional) The Cisco-proprietary algorithm used to encrypt the password. Currently the only encryption type available is 7 . If you specify <i>encryption-type</i> , the next argument you supply must be an encrypted password (a password already encrypted by a Cisco router).
<i>encrypted-password</i>	An encrypted password you enter, copied from another router configuration.

Default

No password is defined.

Command Mode

Global configuration

Usage Guidelines

You will not ordinarily enter an encryption type. Typically you enter an encryption type only if you copy and paste back into this command a password that has already been encrypted by a Cisco router.

Caution If you specify an encryption type and then enter a clear-text password, you will not be able to re-enter enable mode. You cannot recover a lost password that has been encrypted by any method.

Use this command with the **level** option to define a password for a specific privilege level. Once the level and the password are specified, give the password to the users you want to have access at this level. Use the **privilege level (global)** configuration command to specify commands accessible at various level.

Enable or disable password encryption with the **service password-encryption** command.

An enable password can contain from 1 to 80 uppercase and lowercase alphanumeric characters. The first character cannot be a number. Some spaces are valid password characters; for example, “two words” is valid. Leading spaces are ignored, but trailing spaces are recognized. For example “woolly” is interpreted as "woolly" (without the space), while "woolly " is interpreted as "woolly " (with the space). To create an enable password containing a question mark (?), precede the question mark with the keystrokes **Ctrl-V**. For example, to create the password "abc?123", you enter the letters abc followed by **Ctrl-V** followed by ? followed by the numbers 123. When the system prompts you to enter the enable password, you need not precede the question mark with the **Ctrl-V**; you can simply enter abc?123 at the password prompt.

Example

In the following example, the password *pswd2* is enabled for privilege level 2:

```
enable password level 2 pswd2
```

In the following example the encrypted password Xyti5Rkls3LoyxzS8t9, which has been copied from a router configuration file, is set for privilege level 2 using encryption type 7:

```
enable password level 2 7 Xyti5Rkls3LoyxzS8t9
```

Related Commands

A dagger (†) indicates that the command is documented outside this chapter.

disable †

enable †

privilege level (global)

service password-encryption

show privilege

enable secret

Use the **enable secret** command to specify an additional layer of security over the **enable password** command. Use the **no** form of the command to turn off the enable secret function.

```
enable secret [level level] {password}  
enable secret [level level] {encryption-type encrypted-password}  
no enable secret [level level]
```

Syntax Description

level <i>level</i>	(Optional) Level for which the password applies. You can specify up to sixteen privilege levels, using numbers 0 through 15. Level 1 is normal EXEC-mode user privileges. If this argument is not specified, the privilege level defaults to 15 (traditional enable privileges). The same holds true for the no form of the command.
<i>password</i>	The enable secret password—the password you would type when enabling. This password should be different from the password created with the enable password command. When the router displays the password for you later, it will be displayed encrypted (an encrypted form of what you typed).
<i>encryption-type</i>	(Optional) The Cisco-proprietary algorithm used to encrypt the password. Current the only encryption type available for this command is 5 . If you specify <i>encryption-type</i> , the next argument you supply must be an encrypted password (a password encrypted by a Cisco router).
<i>encrypted password</i>	An encrypted password you enter, copied from another router configuration.

Default

Disabled

Command Mode

Global configuration

Usage Guidelines

You will not ordinarily enter an encryption type. Typically you enter an encryption type only if you paste back into this command an encrypted password that you copied from a router configuration file.

Caution If you specify encryption-type and then enter a clear-text password, you will not be able to re-enter enable mode. You cannot recover a lost password that has been encrypted by any method.

The **enable secret** command is used in conjunction with the **enable password** command to provide an additional layer of security over the enable password. This scheme provides better security by storing the enable secret using a non-reversible cryptographic function.

This added layer of security is useful in environments where the password crosses the network or is stored on a TFTP server.

If you use the same password for **enable password** and **enable secret**, you receive an error message warning that this practice is not recommended; but the password will be accepted. By using the same password, however, you undermine the additional security the **enable secret** command provides.

Note After you set a password using **enable secret**, a password set using the **enable password** command will no longer work unless **enable secret** is disabled or an older version of Cisco IOS software is being used, such as when running an older rxboot image. Additionally, you cannot recover a lost password that has been encrypted by any method.

Examples

The following example specifies an enable secret password of gobbledegook:

```
enable secret gobbledegook
```

After specifying an enable secret password, users must enter this password to gain access. Any passwords set through enable password will no longer work.

```
Password: gobbledegook
```

In the following example the encrypted password Xyti5Rkls3LoyxzS8t9, which has been copied from a router configuration file, is enabled for privilege level 2 using encryption type 5:

```
enable password level 2 5 $1$FaD0$Xyti5Rkls3LoyxzS8t98j2
```

Related Commands

A dagger (†) indicates that the command is documented outside this chapter.

enable †
enable password

enable use-tacacs

To enable use of TACACS to determine whether a user can access the privileged command level, use the **enable use-tacacs** global configuration command. Use the **no enable use-tacacs** command to disable TACACS verification.

enable use-tacacs
no enable use-tacacs

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Global configuration

Usage Guidelines

When you add this command to the configuration file, the EXEC **enable** command prompts for a new username and password pair. This pair is then passed to the TACACS server for authentication. If you are using extended TACACS, it also will pass any already-existing UNIX user identification code to the server.

Note This command initializes TACACS. Use the **tacacs server-extended** command to initialize XTACACS, or use the **aaa new-model** command to initialize AAA/TACACS+.



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

hostname

To specify or modify the host name for the network server, use the **hostname** global configuration command.

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 *cs*.

Command Mode

Global configuration

Usage Guidelines

The order of display at startup is the message-of-the-day (MOTD) banner, then login and password prompts, then the EXEC banner.

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

Example

The following example changes the host name to *sandbox*:

```
hostname sandbox
```

load-interval

To change the length of time for which data is used to compute load statistics, use the **load-interval** interface configuration command. Use the **no load-interval** command to revert to the default setting.

load-interval *seconds*
no load-interval *seconds*

Syntax Description

seconds Length of time for which data is used to compute load statistics. A value that is a multiple of thirty, between 30 and 600 (30, 60, 90, 120, and so forth).

Default

300 seconds (or 5 minutes)

Command Mode

Interface configuration

Usage Guidelines

If you want load computations to be more reactive to short bursts of traffic rather than averaged over five-minute periods, you can shorten the length of time over which load averages are computed.

If the load interval is set to thirty seconds, new data is used for load calculations over a thirty-second period. This data is used to compute load statistics, including input rate in bits and packets per second, output rate in bits and packets per second, load, and reliability.

Load data is gathered every five seconds on the communication server. This data is used for a weighted average calculation in which more-recent load data has more weight in the computation than older load data. If the load interval is set to thirty seconds, the average is computed for the last thirty seconds of load data.

The **load-interval** command allows you to change the default interval five minutes to a shorter or longer period of time. If you change it to a shorter period of time, the input and output statistics that are displayed when you use the **show interface** command will be more current and based on more instantaneous data rather than reflecting a more average load over a longer period of time.

This command is often used for dial backup purposes to increase or decrease the likelihood of a backup interface being implemented, but it can be used on any interface.

Example

In the following example, the default five-minute average is set it to a thirty-second average. A burst in traffic that would not trigger a dial backup for an interface configured with the default five-minute interval might trigger a dial backup for this interface that is set for a shorter, thirty-second interval.

```
interface serial 0
load-interval 30
```

Related Command

show interface

logging

To log messages to a syslog server host, use the **logging** global configuration command. The **no logging** command deletes the syslog server with the specified address from the list of syslogs.

logging *host*
no logging *host*

Syntax Description

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

Default

No messages are logged to a syslog server host.

Command Mode

Global configuration

Usage Guidelines

This command identifies a syslog server host to receive logging messages. By issuing this command more than once, you build a list of syslog servers that receive logging messages.

Example

The following example logs messages to a host named *johnson*:

```
logging johnson
```

Related Commands

logging trap
service timestamps

logging buffered

To log messages to an internal buffer, use the **logging buffered** global configuration command. The **no logging buffered** command cancels the use of the buffer and writes messages to the console terminal, which is the default.

logging buffered
no logging buffered

Syntax Description

This command has no arguments or keywords.

Default

The communication server displays all messages to the console terminal.

Command Mode

Global configuration

Usage Guidelines

This command copies logging messages to an internal buffer instead of writing them to the console terminal. The buffer is circular in nature, so newer messages overwrite older messages.

To display the messages that are logged in the buffer, use the EXEC command **show logging**. The first message displayed is the oldest message in the buffer.

Example

The following example illustrates how to enable logging to an internal buffer:

```
logging buffered
```

logging console

To limit messages logged to the console based on severity, use the **logging console** global configuration command. To disable logging to the console terminal, use the **no** form of the command.

logging console *level*
no logging console

Syntax Description

level Limits the logging of messages displayed on the console terminal to the specified level and levels below it. See Table 5-7 for a list of the *level* keywords.

Default

The **debugging** level

Command Mode

Global configuration

Usage Guidelines

Specifying one of the level names shown in Table 5-7 causes messages at that level and numerically lower levels to be displayed at the console terminal.

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

Table 5-7 Error Message Logging Priorities

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

Example

The following example changes the level of messages displayed to the console terminal to **alerts**, which means alerts and emergencies are displayed:

```
logging console alerts
```

Related Command

logging facility

logging facility

To configure the syslog facility in which error messages are sent, use the **logging facility** global configuration command. To revert to the default of local7, use the **no logging facility** command.

logging facility *facility-type*
no logging facility

Syntax Description

facility-type Logging facility type. See Table 5-8 for the *facility-type* keywords.

Default

local7

Command Mode

Global configuration

Usage Guidelines

Table 5-8 Logging Facility Facility-Type Keywords

Keyword	Description
auth	Authorization system
cron	Cron facility
daemon	System daemon
kern	Kernel
local0–7	Reserved for locally defined messages
lpr	Line printer system
mail	Mail system
news	USENET news
sys9	System use
sys10	System use
sys11	System use
sys12	System use
sys13	System use
sys14	System use
syslog	System log
user	User process
uucp	UNIX-to-UNIX copy system

Example

The following example configures the syslog facility to Kernel:

```
logging facility kern
```

Related Command

logging console

logging monitor

To limit messages logged to the terminal lines (monitors) based on severity, use the **logging monitor** global configuration command. Use the **no** form of this command to disable logging to terminal lines other than the console line.

logging monitor *level*
no logging monitor

Syntax Description

level One of the *level* keywords listed in Table 5-7

Default

debugging

Command Mode

Global configuration

Usage Guidelines

Specifying a level causes messages at that level and numerically lower levels to be displayed to the monitor.

This command limits the logging messages displayed on terminal lines other than the console line to messages with a level at or above the specified level.

Example

The following example specifies that only messages of the levels **errors**, **critical**, **alerts**, and **emergencies** be displayed on terminals:

```
logging monitor errors
```

Related Command

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

terminal monitor †

logging on

To control logging of error messages, use the **logging on** global configuration command. This command enables or disables message logging to all destinations except the console terminal. The **no logging on** command enables logging to the console terminal only.

logging on
no logging on

Syntax Description

This command has no arguments or keywords.

Default

The communication server logs messages to the console terminal.

Command Mode

Global configuration

Example

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

```
no logging on
```

logging synchronous

To synchronize unsolicited messages and **debug** output with solicited system output and prompts for a specific line, use the **logging synchronous** line configuration command. To disable this capability, use the **no** form of this command.

logging synchronous [**level** *severity-level* | **all**] [**limit** *number-of-buffers*]
no logging synchronous [**level** *severity-level* | **all**] [**limit** *number-of-buffers*]

Syntax Description

level <i>severity-level-number</i>	(Optional) Message severity level. Messages with a severity level equal to or higher than this value are printed asynchronously. When specifying a severity level number, consider that for the logging system, low numbers indicate greater severity and high numbers indicate lesser severity. The default value is 2.
all	(Optional) Specifies that all messages are printed asynchronously, regardless of the severity level.
limit <i>number-of-buffers</i>	(Optional) Number of buffers to be queued for the terminal after which new messages are dropped. The default value is 20.

Defaults

This feature is turned off by default.

If you do not specify a severity level, the default value of 2 is assumed.

If you do not specify the maximum number of buffers to be queued, the default value of 20 is assumed.

Command Mode

Line configuration

Usage Guidelines

When synchronous logging of unsolicited messages and debug output is turned on, unsolicited router output is displayed on the console or printed after solicited router output is displayed or printed. Unsolicited messages and debug output is displayed on the console after the prompt for user input is returned. This is to keep unsolicited messages and debug output from being interspersed with solicited router output and prompts. After the unsolicited messages are displayed, the console displays the user prompt again.

When specifying a severity level number, consider that for the logging system, low numbers indicate greater severity and high numbers indicate lesser severity.

When a terminal line's message-queue limit is reached, new messages are dropped from the line, although these messages might be displayed on other lines. If messages are dropped, the notice "%SYS-3-MSGLOST *number-of-messages* due to overflow" follows any messages that are displayed. This notice is displayed only on the terminal that lost the messages. It is not sent to any other lines, any logging servers, or the logging buffer.



Caution By configuring abnormally large message-queue limits and setting the terminal to “terminal monitor” on a terminal that is accessible to intruders, you expose yourself to “denial of service” attacks. An intruder could carry out the attack by putting the terminal in synchronous output mode, making a Telnet connection to a remote host, and leaving the connection idle. This could cause large numbers of messages to be generated and queued, and these messages would consume all available RAM. Although unlikely to occur, you should guard against this type of attack through proper configuration.

Example

The following example identifies a line and configures synchronous logging for that line, then it does this for another line:

```
line 0 4
logging synchronous level 6
line 2
logging synchronous level 7 limit 70000
```

Related Command

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

line †

logging trap

To limit messages logged to the syslog servers based on severity, use the **logging trap** global configuration command. Use the **no** form of this command to disable logging to syslog servers.

logging trap *level*
no logging trap

Syntax Description

level One of the *level* keywords listed in Table 5-7

Default

informational

Command Mode

Global configuration

Usage Guidelines

The EXEC command **show logging** displays the addresses and levels associated with the current logging setup. The command output also includes ancillary statistics. This command limits the logging of error messages sent to syslog servers to only those messages at the specified level.

Table 5-7 lists the syslog definitions that correspond to the debugging message levels. Additionally, there are four categories of messages generated by the software, as follows:

- Error messages about software or hardware malfunctions at the LOG_ERR level
- Output for the **debug** commands at the LOG_WARNING level
- Interface up/down transitions and system restarts at the LOG_NOTICE level
- Reload requests and low process stacks are at the LOG_INFO level

Use the **logging** and **logging trap** commands to send messages to a UNIX syslog server.

Example

The following example logs messages to a host named *johnson* and limits messages logged to the syslog server.

```
logging johnson
logging trap notifications
```

Related Command

logging

login authentication

To enable TACACS+ authentication for logins, use the **login authentication** command. Use the **no** form of the command to return to the default.

```
login authentication { default | list-name }  
no login authentication { default | list-name }
```

Syntax Description

default	Uses the default list created with the aaa authentication login command.
<i>list-name</i>	Uses the indicated list created with the aaa authentication login command.

Default

Uses the default set with **aaa authentication login**.

Command Mode

Line configuration

Usage Guideline

This command is a per-line command used with AAA, and specifies the name of a list of TACACS+ authentication methods to try at login. If no list is specified, the default list will be used (whether or not it is specified in the command line). Defaults and lists are created by using the **aaa authentication login** command. Entering the **no** version of **login authentication** has the same effect as entering the command with the **default** argument.

Before issuing this command, create a list of authentication processes by using the global configuration **aaa authentication login** command.



Caution If you use a *list-name* that was not configured with the **aaa authentication login** command, you will disable login on this line.

Examples

The following example specifies that the default AAA authentication is to be used on line 4:

```
line 4  
login authentication default
```

The following example specifies that the AAA authentication list called MIS-access is to be used on line 7:

```
line 7  
login authentication MIS-access
```

Related Command

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

aaa authentication login†

ntp access-group

To control access to the system's Network Time Protocol (NTP) services, use the **ntp access-group** global configuration command. To remove access control to the system's NTP services, use the **no** form of this command.

```
ntp access-group { query-only | serve-only | serve | peer } access-list-number
no ntp access-group { query-only | serve-only | serve | peer }
```

Syntax Description

query-only	Allows only NTP control queries. See RFC 1305 (NTP version 3).
serve-only	Allows only time requests.
serve	Allows time requests and NTP control queries, but does not allow the system to synchronize to the remote system.
peer	Allows time requests and NTP control queries and allows the system to synchronize to the remote system.
<i>access-list-number</i>	Number (1 to 99) of a standard IP access list.

Default

No access control (full access granted to all systems)

Command Mode

Global configuration

Usage Guidelines

The access group options are scanned in the following order from least restrictive to most restrictive:

- 1 **peer**
- 2 **serve**
- 3 **serve-only**
- 4 **query-only**

Access is granted for the first match that is found. If no access groups are specified, all access is granted to all sources. If any access groups are specified, only the specified access is granted. This facility provides minimal security for the time services of the system. However, it can be circumvented by a determined programmer. If tighter security is desired, use the NTP authentication facility.

Example

In the following example, the system is configured to allow itself to be synchronized by a peer from access list 99. However, the system restricts access to allow only time requests from access list 42.

```
ntp access-group peer 99
ntp access-group serve-only 42
```

Related Command

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

access-list †

ntp authenticate

To enable Network Time Protocol (NTP) authentication, use the **ntp authenticate** global configuration command. Use the **no** form of this command to disable the feature.

ntp authenticate
no ntp authenticate

Syntax Description

This command has no keywords or arguments.

Default

No authentication

Command Mode

Global configuration

Usage Guidelines

Use this command if you want authentication. If this command is specified, the system will not synchronize to a system unless it carries one of the authentication keys specified in the **ntp trusted-key** command.

Example

The following example enables NTP authentication:

```
ntp authenticate
```

Related Commands

ntp authentication-key
ntp trusted-key

ntp authentication-key

To define an authentication key for Network Time Protocol (NTP), use the **ntp authentication-key** global configuration command. Use the **no** form of this command to remove the authentication key for NTP.

```
ntp authentication-key number md5 value  
no ntp authentication-key number
```

Syntax Description

<i>number</i>	Key number (1 to 4294967295)
md5	Key type
<i>value</i>	Key value (an arbitrary string of up to eight characters)

Default

No authentication key is defined for NTP.

Command Mode

Global configuration

Usage Guidelines

Use this command to define authentication keys for use with other NTP commands in order to provide a higher degree of security. Currently, only the key type **md5** is supported.

Example

The following example sets authentication key 10 to *aNiceKey*:

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

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

Related Commands

```
ntp authenticate  
ntp peer  
ntp server  
ntp trusted-key
```

ntp 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 this command to disable this capability.

```
ntp broadcast [version number]  
no ntp broadcast
```

Syntax Description

version *number* (Optional) Number from 1 to 3 indicating the NTP version

Default

Disabled

Command Mode

Interface configuration

Example

In the following example, Ethernet interface 0 is configured to send NTP version 2 packets:

```
interface ethernet 0  
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** interface configuration command. Use the **no** form of this 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 Ethernet interface 1:

```
interface ethernet 1
 ntp broadcast client
```

Related Commands

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:

```
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 automatically saves this value into the system configuration using the **ntp clock-period** global configuration command. The system uses the **no** form of this command to revert to the default.

ntp clock-period *value*
no ntp clock-period

Syntax Description

value Amount to add to the system clock for each clock hardware tick (in units of 2-32 seconds).

Default

17179869 (4 milliseconds)

Command Mode

Global configuration

Usage Guidelines

If a **write memory** command is entered to save the configuration to nonvolatile memory, this command will automatically be added to the configuration. It is a good idea to use the **write memory** command after NTP has been running for a week or so; this will help NTP synchronize more quickly if the system is restarted.

Do not enter this command; it is documented for informational purposes only. The system automatically generates this command as Network Time Protocol (NTP) determines the clock error and compensates.

ntp disable

To prevent an interface from receiving Network Time Protocol (NTP) packets, use the **ntp disable** interface configuration command. To enable receipt of NTP packets on an interface, use the **no** form of this command.

ntp disable
no ntp disable

Syntax Description

This command has no arguments or keywords.

Default

Enabled

Command Mode

Interface configuration

Usage Guidelines

This command provides a simple method of access control.

Example

In the following example, Ethernet interface 0 is prevented from receiving NTP packets:

```
interface ethernet 0
 ntp disable
```

ntp master

To configure the communication server as a Network Time Protocol (NTP) master clock to which peers synchronize themselves when an external NTP source is not available, use the **ntp master** 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

Because 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 communication server has **ntp master** configured, and it cannot reach any clock with a lower stratum number, the communication server will claim to be synchronized at the configured stratum number, and other communication servers will be willing to synchronize to it via NTP.

Note The system clock must have been set from some source, either by taking the time from another source or by having the time set 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
```


Related Command
clock calendar-valid

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** form of this command.

```
ntp peer ip-address [version number] [key keyid] [source interface] [prefer]  
no ntp peer ip-address
```

Syntax Description

<i>ip-address</i>	IP address of the peer providing, or being provided, the clock synchronization.
version <i>number</i>	(Optional) Defines the Network Time Protocol (NTP) version number (1 to 3).
key <i>keyid</i>	(Optional) Defines the authentication key when sending packets to this peer.
source <i>interface</i>	(Optional) Names 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 communication server 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 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 interface 0.

```
ntp peer 131.108.22.33 version 2 source Ethernet 0
```

Related Commands

```
ntp authentication-key  
ntp server  
ntp source
```

ntp server

To allow the 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** form of this command.

```
ntp server ip-address [version number] [key keyid] [source interface] [prefer]  
no ntp server ip-address
```

Syntax Description

<i>ip-address</i>	IP address of the time server providing the clock synchronization.
version number	(Optional) Defines the Network Time Protocol (NTP) version number (1 to 3).
key keyid	(Optional) Defines the authentication key to use when sending packets to this peer.
source interface	(Optional) Identifies 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 communication server to synchronize with the specified server. The server will not synchronize to this communication server.

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 Network Time Protocol (NTP) packets, use the **ntp source** global configuration command. Use the **no** form of this command to remove the specified source address.

```
ntp source interface  
no ntp source
```

Syntax Description

interface Any valid system interface name

Default

Source address is determined by the outgoing interface.

Command Mode

Global configuration

Usage Guidelines

Use this command when you want to use a particular source IP address for all NTP packets. The address is taken from the named interface. This command is useful if the address on an interface cannot be used as the destination for reply packets. If the **source** keyword is present on an **ntp server** or **ntp peer** command, that value overrides the global value.

Example

In the following example, the communication server is configured to use the IP address of Ethernet interface 0 as the source address of all outgoing NTP packets:

```
ntp source ethernet 0
```

Related Commands

```
ntp peer  
ntp server
```

ntp trusted-key

If you want to authenticate the identity of a system to which Network Time Protocol (NTP) will synchronize, use the **ntp trusted-key** global configuration command. Use the **no** form of this command to disable authentication of the identity of the system.

```
ntp trusted-key key-number  
no ntp trusted-key key-number
```

Syntax Description

key-number Key number of authentication key to be trusted

Default

Disabled

Command Mode

Global configuration

Usage Guidelines

If authentication is enabled, use this command to define one or more key numbers (corresponding to the keys defined with the **ntp authentication-key** command) that a peer NTP system must provide in its NTP packets, in order for this system to synchronize to it. This provides protection against accidentally synchronizing the system to a system that is not trusted, since the other system must know the correct authentication key.

Example

In the following example, the system is configured to synchronize only to systems providing authentication key 42 in its NTP packets:

```
ntp authenticate  
ntp authentication-key 42 md5 aNiceKey  
ntp trusted-key 42
```

Related Commands

ntp authenticate
ntp authentication-key

ping (user)

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

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

Syntax Description

<i>protocol</i>	(Optional) Protocol keyword, either ip or novell
<i>host</i>	Host name of system to ping
<i>address</i>	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. Unlike the privileged-level **ping** command, the values for the number of ping packets sent, the datagram size, and the timeout cannot be adjusted.

If the system cannot map an address for a host name, it will return an “%Unrecognized host or address” error message.

To abort a ping session, type the escape sequence (by default, Ctrl-^ X, which is done by simultaneously pressing the Ctrl, Shift, and 6 keys, letting go, then pressing the X key).

Table 5-9 describes the test characters that the ping facility sends.

Table 5-9 Ping Test Characters

Char	Meaning
!	Each exclamation point indicates receipt of a reply.
.	Each period indicates the network server timed out while waiting for a reply.
U	A destination unreachable error PDU was received.
C	A congestion experienced packet was received.
I	User interrupted test.
?	Unknown packet type.
&	Packet lifetime exceeded.

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
```

Table 5-10 describes the default **ping** fields shown in the display.

Table 5-10 Ping Field Descriptions

Field	Description
Sending 5, 100-byte ICMP echos to ...	Indicates the number of ping packets sent to the specified host name, the datagram size, and the timeout value.
!!!!	Each exclamation point (!) indicates receipt of a reply. A period (.) indicates the network server timed out while waiting for a reply. Other characters might 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 = 1/3/4 ms	Round-trip travel time intervals for the protocol echo packets, including minimum/average/maximum (in milliseconds).

Related Command

ping (privileged)

ping (privileged)

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

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

Syntax Description

protocol (Optional) Protocol keyword, either **ip** or **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.

Depending upon the protocol type, You can adjust values for the number of ping packets to be sent, the datagram size, the timeout interval, additional command to include, and the sizes of the echo packets being sent.

After you enter the **ping** command in privileged mode, the system prompts for one of the following keywords: **ip** or **ipx**. 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.

To abort a ping session, type the escape sequence (by default, Ctrl-^ X, which is done by simultaneously pressing the Ctrl, Shift, and 6 keys, letting go, then pressing the X key).

Table 5-11 describes the test characters that the ping facility sends.

Table 5-11 Ping Test Characters

Char	Meaning
!	Each exclamation point indicates receipt of a reply.
.	Each period indicates the network server timed out while waiting for a reply.
U	A destination unreachable error PDU was received.
C	A congestion experienced packet was received.
I	User interrupted test.
?	Unknown packet type.
&	Packet lifetime exceeded.

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

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 5-12 describes the default **ping** fields shown in the display.

Table 5-12 Ping Field Descriptions

Field	Description
Protocol [ip]:	Prompts for a supported protocol. Enter ip or 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 might 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)

ppp authentication

To enable Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP), and to enable an AAA authentication method on an interface, use the **ppp authentication** interface configuration command. Use the **no** form of the command to disable this authentication.

```
ppp authentication { chap | pap } [if-needed] [list-name]  
no ppp authentication
```

Syntax Description

chap	Enable CHAP on a serial interface.
pap	Enable PAP on a serial interface.
if-needed	(Optional) Used with TACACS and XTACACS. Do not perform CHAP or PAP authentication if user has already provided authentication. This option is available only on asynchronous interfaces.
<i>list-name</i>	(Optional) Used with AAA/TACACS+. Specify the name of a list of AAA methods of authentication to use. If no listname is specified, the system will use the default. Lists and default are created using the aaa authentication ppp command.

Default

PPP authentication is not enabled.

Command Mode

Interface configuration

Usage Guidelines

Once you have enabled CHAP or PAP, the local communication server requires a password from remote devices. If the remote device does not support CHAP or PAP, no traffic will be passed to that device.

If you are using **autoselect** on a tty line, you will probably want to use the **ppp authentication** command to turn on PPP authentication for the corresponding interface.

When you specify the **if-needed** option, PPP authentication is not required when the user has already provided authentication. This option is useful in conjunction to the **autoselect** command, but cannot be used with AAA/TACACS+.

The *list-name* keyword can only be used when AAA/TACACS+ is initialized and cannot be used with the **if-needed** argument.



Caution If you use a *list-name* that was not configured with the **aaa authentication ppp** command, you will disable PPP on this interface.

Example

The following example enables CHAP on asynchronous interface 4:

```
interface async 4
 encapsulation ppp
 ppp authentication chap
```

Related Commands

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

aaa authentication ppp†

aaa new-model

autoselect†

encapsulation ppp†

ppp use-tacacs†

username†

ppp use-tacacs

To enable TACACS for PPP authentication, use the **ppp use-tacacs** interface configuration command. Use the **no** form of this command to disable TACACS for PPP authentication.

```
ppp use-tacacs [single-line]
no ppp use-tacacs
```

Syntax Description

single-line (Optional) Accept the username and password in the username field. This option applies only when using CHAP authentication.

Default

TACACS is not used for PPP authentication.

Command Mode

Interface configuration

Usage Guidelines

This is a per-interface command. Use this command only when you have set up an extended TACACS server. This command requires the new extended TACACS server.

When CHAP authentication is being used, the **ppp use-tacacs** command with the **single-line** option specifies that if a username and password are specified in the username, separated by an asterisk (*), then a standard tacacs login query is performed using that username and password. If the username does not contain an asterisk, then normal CHAP authentication is performed using TACACS.

This feature is useful when integrating TACACS with other authentication systems that require a clear-text version of the user's password. Such systems include one-time password systems, token card systems, and others.



Caution Normal CHAP authentications prevent the clear-text password from being transmitted over the link. When you use the single-line option, passwords will cross the link in the clear.

If the username and password are contained in the CHAP password, then the CHAP secret is not used by the Cisco system. Because most PPP clients will require that a secret be specified, you can use any arbitrary string; the Cisco system will ignore it.

Note This command is not used in AAA/TACACS+ and has been replaced with the **aaa authentication ppp** command.

Examples

In the following example, asynchronous serial interface 1 is configured to use TACACS for CHAP authentication:

```
interface async 1
  ppp authentication chap
  ppp use-tacacs
```

In the following example, asynchronous serial interface 1 is configured to use TACACS for PAP authentication:

```
interface async 1
  ppp authentication pap
  ppp use-tacacs
```

Related Commands

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

ppp authentication chap[†]
ppp authentication pap[†]
tacacs-server extended[†]
tacacs-server host[†]

priority-group

To assign the specified priority list to an interface, use the **priority-group** interface configuration command. Use the **no** form of this command to remove the specified **priority-group** assignment.

priority-group *list*
no priority-group

Syntax Description

list Priority list number assigned to the interface

Default

None

Command Mode

Interface configuration

Usage Guidelines

Only one list can be assigned per interface. Priority output queueing provides a mechanism to prioritize packets transmitted on an interface.

Example

The following example causes packets on serial interface 0 to be classified by priority list 1:

```
interface serial 0
priority-group 1
```

Related Commands

priority-list
priority-list interface
priority-list queue-limit
priority-list stun

priority-list default

To assign a priority queue for those packets that do not match any other rule in the priority list, use the **priority-list default** global configuration command. Use the **no** form of this command to return to the default or assign **normal** as the default.

```
priority-list list-number default {high | medium | normal | low}  
no priority-list list-number default {high | medium | normal | low}
```

Syntax Description

list-number Arbitrary integer between 1 and 10 that identifies the priority list selected by the user

high | **medium** | **normal** | **low** Priority queue level

Default

The **normal** queue is assumed if you use the **no** form of the command.

Command Mode

Global configuration

Example

The following example sets the priority queue for those packets that do not match any other rule in the priority list to a low priority:

```
priority-list 1 default low
```

Related Commands

priority-group
show queueing

priority-list interface

To establish queuing priorities on packets entering from a given interface, use the **priority-list interface** global configuration command. Use the **no priority-list** command with the appropriate arguments to remove an entry from the list.

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

Syntax Description

<i>list-number</i>	Arbitrary integer between 1 and 10 that identifies the priority list selected by the user
<i>interface-type</i>	Name of the interface
<i>interface-number</i>	Number of the specified interface
high medium normal low	Priority queue level

Default

No queuing priorities are established.

Command Mode

Global configuration

Example

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

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

Related Commands

priority-group
show queueing

priority-list protocol

To establish queuing priorities based upon the protocol type, use the **priority-list protocol** global configuration command. Use the **no** form of this command with the appropriate list number to remove an entry from the list.

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

Syntax Description

<i>list-number</i>	Arbitrary integer between 1 and 10 that identifies the priority list selected by the user.
<i>protocol-name</i>	Specifies the protocol type: arp , compressedtcp , ip , ipx , pad , and x25 .
high medium normal low	Priority queue level.
<i>queue-keyword</i> <i>keyword-value</i>	Possible keywords are gt , lt , list , tcp , and udp . See Table 5-13.

Default

No queuing priorities are established.

Command Mode

Global configuration

Usage Guidelines

When using multiple rules for a single protocol, remember that the system reads the priority settings in order of appearance. When classifying a packet, the system searches the list of rules specified by **priority-list** commands for a matching protocol type. When a match is found, the packet is assigned to the appropriate queue. The list is searched in the order it is specified, and the first matching rule terminates the search.

Use Table 5-13, Table 5-14, and Table 5-15 to configure the queuing priorities for your system.

Table 5-13 Protocol Priority Queue Keywords and Values

Option	Description
gt <i>byte-count</i>	Specifies a greater-than count. The priority level assigned goes into effect when a packet exceeds the value entered for the argument <i>byte-count</i> . The size of the packet must also include additional bytes due to MAC encapsulation on the outgoing interface.
lt <i>byte-count</i>	Specifies a less-than count. The priority level assigned goes into effect when a packet size is less than the value entered for <i>byte-count</i> . The size of the packet must also include additional bytes due to MAC encapsulation on the outgoing interface.

Option	Description
list <i>list-number</i>	Assigns traffic priorities according to a specified list when used with IP or IPX. The <i>list-number</i> argument is the access list number as specified by the access-list global configuration command for the specified <i>protocol-name</i> .
tcp <i>port</i>	Assigns the priority level defined to TCP segments originating from or destined to a specified port (for use with the IP protocol only). Table 5-14 lists common TCP services and their port numbers.
udp <i>port</i>	Assigns the priority level defined to UDP packets originating from or destined to the specified port (for use with the IP protocol only). Table 5-15 lists common UDP services and their port numbers.

Table 5-14 Common TCP Services and Port Numbers

Service	Port
Telnet	23
SMTP	25

Table 5-15 Common UDP Services and Port Numbers

Service	Port
TFTP	69
NFS	2049
SNMP	161
RPC	111
DNS	53

Note The TCP and UDP ports listed in Table 5-14 and Table 5-15 include some of the more common port numbers. However, you can specify any port number to be prioritized; you are not limited to those listed.

Use the **no priority-list** global configuration command followed by the appropriate *list-number* argument and the **protocol** keyword to remove a priority list entry assigned by protocol type.

Examples

The following example assigns a high-priority level to traffic that matches IP access list 10:

```
priority-list 1 protocol ip high list 10
```

The following example assigns a medium-priority level to Telnet packets:

```
priority-list 4 protocol ip medium tcp 23
```

The following example assigns a medium-priority level to UDP Domain Name Service packets:

```
priority-list 4 protocol ip medium udp 53
```

The following example assigns a high-priority level to traffic that matches Ethernet type code access list 201:

```
priority-list 1 protocol bridge high list 201
```

Related Commands

priority-group
show queueing

priority-list queue-limit

To specify the maximum number of packets that can be waiting in each of the priority queues, use the **priority-list queue-limit** global configuration command. Use the **no** form of this command to select the normal queue.

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

Syntax Description

<i>list-number</i>	Arbitrary integer between 1 and 10 that identifies the priority list selected by the user.
<i>high-limit medium-limit normal-limit low-limit</i>	Priority queue maximum length. A value of 0 for any of the four arguments means that the queue can be of unlimited size for that particular queue.

Default

The default queue limit arguments are listed in Table 5-16.

Table 5-16 Default Priority Queue Packet Limits

Priority Queue Argument	Packet Limits
<i>high-limit</i>	20
<i>medium-limit</i>	40
<i>normal-limit</i>	60
<i>low-limit</i>	80

Command Mode

Global configuration

Usage Guidelines

If a priority queue overflows, excess packets are discarded and quench messages can be sent, if appropriate, for the protocol.

Example

The following example sets the maximum packets in the priority queue to 10:

```
priority-list 2 queue-limit 10 40 60 80
```

Related Commands

priority-group
show queueing

privilege level (global)

To set the privilege level for a command, use the **privilege level** global configuration command. Use the **no privilege level** command to revert to default privileges for a given command.

```
privilege mode level level command  
no privilege mode level level command
```

Syntax Description

<i>mode</i>	Configuration mode. See Table 5-6 in the description of the alias command for a list of acceptable options.
<i>level</i>	Privilege level to be associated with the specified command. You can specify up to sixteen privilege levels, using numbers 0 through 15.
<i>command</i>	Command to which privilege level is associated.

Default

Level 15 is the level of access permitted by the **enable** password.

Level 1 is normal EXEC-mode user privileges.

Command Mode

Global configuration

Usage Guidelines

Table 5-6 in the description of the **alias** command shows the acceptable options for the *mode* argument in the **privilege level** global configuration command.

The password for the privilege level defined using the **privilege level** global configuration mode is configured using the **enable password** command.

Level 0 can be used to specify a more-limited subset of commands for specific users or lines. For example, you can allow user “guest” to only use the **show users** and **exit** commands.

Example

In the following example, the **configure** command in global configuration mode is assigned a privilege level of 14. Only users who know the level 14 password will be able to use the **configure** command.

```
privilege exec level 14 configure  
enable password level 14 pswd14
```

Related Commands

enable password
privilege level (line)

privilege level (line)

To set the default privilege level for a line, use the **privilege level** line configuration command. Use the **no privilege level** command to restore the default user privilege level to the line.

privilege level *level*
no privilege level

Syntax Description

level Privilege level to be associated with the specified line.

Default

Level 15 is the level of access permitted by the enable password.

Level 1 is normal EXEC-mode user privileges.

Command Mode

Line configuration

Usage Guidelines

The privilege level that is set using this command can be overridden by a user logging into the line and enabling a different privilege level. The user can lower the privilege level by using the **disable** command. If they know the password to a higher privilege level, they can use that password to enable the higher privilege level.

Level 0 can be used to specify a more limited subset of commands for specific users or lines. For example, you can allow user “guest” to only use the **show users** and **exit** commands.

You might specify a high level of privilege for your console line if you are able to restrict who uses that line.

Example

In the following example, the auxiliary line is configured for privilege level 5. Anyone who is using the auxiliary line will have privilege level 5 by default.

```
line aux 0
 privilege level 5
```

Related Commands

enable password
privilege level (global)

prompt

To customize the communication server prompt, use the **prompt** global configuration command. To revert to the default communication server prompt, use the **no prompt** form of this command.

```
prompt string
no prompt [string]
```

Syntax Description

string Communication server prompt. It can consist of all printing characters and the escape sequences listed in Table 5-17 in the Usage Guidelines section below.

Default

The default communication server prompt is either *cs* or the communication server name defined with the **hostname** global configuration command, followed by an angle bracket (>) for EXEC mode or a pound sign (#) for privileged EXEC mode.

Command Mode

Global configuration

Usage Guidelines

You can include escape sequences when specifying the communication server prompt. All escape sequences are preceded by a percent sign (%). Table 5-17 lists the valid escape sequences.

Table 5-17 Custom Communication Server Prompt Escape Sequences

Escape Sequence	Interpretation
%h	Communication server's host name. This is either <i>cs</i> or the name defined with the hostname global configuration command.
%n	Physical terminal line (TTY) number of the EXEC user.
%p	Prompt character itself. It is either an angle bracket (>) for EXEC mode or a pound sign (#) for privileged EXEC mode.
%s	Space.
%t	Tab.
%%	Percent sign (%)

Specifying the command **prompt %h** has the same effect as issuing the **no prompt** command.

Examples

The following example changes the EXEC prompt to include the TTY number, followed by the communication server name and a space:

```
prompt TTY%n@%h%s%p
```

The following are examples of user and privileged EXEC prompts that result from the previous command:

```
TTY17@cs >  
TTY17Scs #
```

Related Command

hostname

queue-list default

To assign a priority queue for those packets that do not match any other rule in the queue list, use the **queue-list default** global configuration command. To restore the default value, use the **no** form of this command.

```
queue-list list-number default queue-number  
no queue-list list-number default queue-number
```

Syntax Description

<i>list-number</i>	Number of the queue list. An integer from 1 to 10.
<i>queue-number</i>	Number of the queue. An integer from 1 to 10.

Default

Queue number 1

Command Mode

Global configuration

Usage Guidelines

Queue number 0 is a system queue. It is emptied before any of the other queues are processed. The system enqueues high-priority packets, such as keepalives, to this queue.

Example

In the following example, the default queue for list 10 is set to queue number 2:

```
queue-list 10 default 2
```

Related Commands

custom-queue-list
show queueing

queue-list interface

To establish queuing priorities on packets entering on an interface, use the **queue-list interface** global configuration command. To remove an entry from the list, use the **no** form of this command.

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

Syntax Description

<i>list-number</i>	Number of the queue list. An integer from 1 to 10.
<i>interface-type</i>	Required argument that specifies the name of the interface.
<i>interface-number</i>	Number of the specified interface.
<i>queue-number</i>	Number of the queue. An integer from 1 to 10.

Default

No queuing priorities are established.

Command Mode

Global configuration

Example

In the following example, queue list 4 established queuing priorities for packets entering on interface tunnel 3. The queue number assigned is 10.

```
queue-list 4 interface tunnel 3 10
```

Related Commands

custom-queue-list
show queueing

queue-list protocol

To establish queuing priority based upon the protocol type, use the **queue-list protocol** global configuration command. Use the **no** form of this command with the appropriate list number to remove an entry from the list.

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

Syntax Description

<i>list-number</i>	Number of the queue list. An integer from 1 to 10.
<i>protocol-name</i>	Required argument that specifies the protocol type: arp , compressedtcp , ip , ipx , pad , and x25 .
<i>queue-number</i>	Number of the queue. An integer from 1 to 10.
<i>queue-keyword</i> <i>keyword-value</i>	Possible keywords are gt , lt , list , tcp , and udp . See Table 5-13.

Default

No queuing priorities are established.

Command Mode

Global configuration

Usage Guidelines

When classifying a packet, the system searches the list of rules specified by **queue-list** commands for a matching protocol type. When a match is found, the packet is assigned to the appropriate queue. The list is searched in the order it is specified, and the first matching rule terminates the search.

Use Table 5-13, Table 5-14, and Table 5-15 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 System packets to queue number 2:

```
queue-list 4 protocol ip 2 udp 53
```

The following example assigns traffic that matches Ethernet type code access list 201 to queue number 1:

```
queue-list 1 protocol bridge 1 list 201
```

Related Commands

custom-queue-list

show queueing

queue-list queue byte-count

To designate the byte size allowed per queue, use the **queue-list queue byte-count** global configuration command. To return the byte size to the default value, use the **no** form of this command.

```
queue-list list-number queue queue-number byte-count byte-count-number  
no queue-list list-number queue queue-number byte-count byte-count-number
```

Syntax Description

<i>list-number</i>	Number of the queue list. An integer from 1 to 10.
<i>queue-number</i>	Number of the queue. An integer from 1 to 10.
<i>byte-count-number</i>	Specifies the lower boundary on how many bytes the system allows to be delivered from a given queue during a particular cycle.

Default

1500 bytes

Command Mode

Global configuration

Example

In the following example, queue list 9 establishes the byte-count as 1400 for queue number 10:

```
queue-list 9 queue 10 byte-count 1400
```

Related Commands

custom-queue-list

show queueing

queue-list queue limit

To designate the queue length limit for a queue, use the **queue-list queue limit** global configuration command. To return the queue length to the default value, use the **no** form of this command.

```
queue-list list-number queue queue-number limit limit-number  
no queue-list
```

Syntax Description

<i>list-number</i>	Number of the queue list. An integer from 1 to 10.
<i>queue-number</i>	Number of the queue. An integer from 1 to 10.
<i>limit-number</i>	Maximum number of packets which can be enqueued at any time. Range is 0 to 32767 queue entries.

Default

20 entries

Command Mode

Global configuration

Example

In the following example, the queue length of queue 10 is increased to 40:

```
queue-list 5 queue 10 limit 40
```

Related Commands

custom-queue-list
show queueing

scheduler-interval

To control the maximum amount of time that can elapse without running the lowest-priority system processes, use the **scheduler-interval** global configuration command. Use the **no** form of this command to restore the default.

scheduler-interval *milliseconds*
no scheduler-interval

Syntax Description

milliseconds Integer that specifies the interval, in milliseconds. The minimum interval that you can specify is 500 milliseconds; there is no maximum value.

Default

500 milliseconds

Command Mode

Global configuration

Usage Guidelines

The normal operation of the network server allows the switching operations to use as much of the central processor as is required. If the network is running unusually heavy loads that do not allow the processor the time to handle the routing protocols, give priority to the system process scheduler. High-priority operations are allowed to use as much of the central processor as needed.

Example

The following example changes the low-priority process schedule to an interval of 750 milliseconds:

```
scheduler-interval 750
```

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. Use the **no** form of this command to restore the default.

```
service decimal-tty  
no service decimal-tty
```

Syntax Description

This command has no arguments or keywords.

Default

Octal line numbers on the ASM-CS; decimal numbers on the 500-CS and Cisco 2500 Series.

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** form of this command to disable this feature.

service exec-wait
no service exec-wait

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Global configuration

Usage Guidelines

This command delays startup of the EXEC until the line has been idle (no traffic seen) for 3 seconds. The default is to enable the line immediately on modem activation.

This command is useful on noisy modem lines or when a modem attached to the line is configured to ignore MNP or V.42 negotiations, and MNP or V.42 modems may be dialing in. In these cases, noise or MNP/V.42 packets might be interpreted as usernames and passwords, causing authentication failure before the user gets a chance to type a username/password. The command is not useful on nonmodem lines or lines without some kind of login configured.

Example

The following example delays the startup of the EXEC:

```
service exec-wait
```

service finger

To allow Finger protocol requests (defined in RFC 742) to be made of the network server, use the **service finger** global configuration command. This service is equivalent to issuing a remote **show users** command. The **no service finger** command removes this service.

```
service finger
no service finger
```

Syntax Description

This command has no arguments or keywords.

Default

Enabled

Command Mode

Global configuration

Example

The following is an example of how to disable the Finger protocol:

```
no service finger
```

service nagle

To enable the Nagle congestion control algorithm, use the **service nagle** global configuration command. Use the **no** form of this command to disable this feature.

service nagle
no service nagle

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Global configuration

Usage Guidelines

When using a standard TCP implementation to send keystrokes between machines, TCP tends to send one packet for each keystroke typed. On larger networks, many small packets use up bandwidth and contribute to congestion.

John Nagle's algorithm (RFC 896) helps alleviate the small-packet problem in TCP. In general, it works this way: The first character typed after connection establishment is sent in a single packet, but TCP holds any additional characters typed until the receiver acknowledges the previous packet. Then the second, larger packet is sent, and additional typed characters are saved until the acknowledgment comes back. The effect is to accumulate characters into larger chunks, and pace them out to the network at a rate matching the round-trip time of the given connection. This method is usually a good for all TCP-based traffic. However, do not use the **service nagle** command if you have XRemote users on X Window sessions.

Example

The following example enables the Nagle algorithm on the communication server:

```
service nagle
```

service password-encryption

To encrypt passwords, use the **service password-encryption** global configuration command. Use the **no** form of this 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. Use the **no** form of this command with the appropriate keyword to disable the keepalives.

```
service tcp-keepalives {in | out}  
no service tcp-keepalives {in | out}
```

Syntax Description

in Generates keepalives on incoming connections (initiated by remote host).
out Generates keepalives on outgoing connections (initiated by a user).

Default
Disabled

Command Mode
Global configuration

Example

The following example generates keepalives on incoming TCP connections:

```
service tcp-keepalives in
```

service telnet-zero-idle

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

```
service telnet-zero-idle  
no service telnet-zero-idle
```

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Global configuration

Usage Guidelines

Normally, data sent to noncurrent Telnet connections is accepted and discarded. When **service telnet-zero-idle** is enabled, if a session is suspended (that is, some other connection is made active or the EXEC is sitting in command mode), the TCP window is set to zero. This action prevents the remote host from sending any more data until the connection is resumed. Use this command when it is important that all messages sent by the host be seen by the users and the users are likely to use multiple sessions.

Do not use this command if your host will eventually time out and log out a TCP user whose window is zero.

Example

The following example sets the TCP window to zero when the Telnet connection is idle:

```
service telnet-zero-idle
```

Related Command

resume

service timestamps

To configure the system to timestamp debugging or logging messages, use one of the **service timestamps** global configuration commands. Use the **no** form of this command to disable this service.

```

service timestamps [type uptime]
service timestamps type datetime [msec] [localtime] [show-timezone]
no service timestamps [type]

```

Syntax Description

<i>type</i>	(Optional) Type of message to timestamp: debug or log .
uptime	(Optional) Timestamp with time since the system was rebooted.
datetime	Timestamp with the date and time.
msec	(Optional) Timestamp includes milliseconds with the date and time.
localtime	(Optional) Timestamp relative to the local time zone.
show-timezone	(Optional) Timestamp includes the time-zone name.

Default

No timestamping.

If **service timestamps** is specified with no arguments or keywords, the 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** with no arguments or keywords disables timestamps for both debugging and logging messages.

Command Mode

Global configuration

Usage Guidelines

Timestamps can be added to either debugging or logging messages independently. The **uptime** form of the command adds timestamps in the format HHHH:MM:SS, indicating the time since the system was rebooted. The **datetime** form of the command adds timestamps in the format MMM DD HH:MM:SS, indicating the date and time according to the system clock. If the system clock has not been set, the date and time are preceded by an asterisk (*) to indicate that the date and time are probably not correct.

Examples

The following example enables timestamps on debugging messages, showing the time since reboot:

```
service timestamps debug uptime
```

The following example enables timestamps on logging messages, showing the current time and date relative to the local time zone, with the time zone name included:

```
service timestamps log datetime localtime show-timezone
```

Related Commands

clock set

debug (Refer to the *Debug Command Reference* publication.)

ntp

show aliases

To display all alias commands, or the alias commands in a specified mode, use the **show aliases** privileged EXEC command.

```
show aliases [mode]
```

Syntax Description

mode (Optional) Command mode. See Table 5-6 in the description of the **alias** command for acceptable options for the *mode* argument.

Table 5-6 shows the options for the optional *mode* argument in the **show aliases** privileged EXEC command.

Command Mode

Privileged EXEC

Usage Guidelines

All of the modes listed in Table 5-6 have their own prompts, except for the null interface mode. For example, the prompt for interface configuration mode is *cs(config-if)*.

Sample Display

The following is sample output from the **show aliases exec** commands. The aliases configured for commands in EXEC mode are displayed.

```
cs# show aliases exec

Exec mode aliases:
  h          help
  lo        logout
  p          ping
  r          resume
  s          show
  w          where
```

Related Command

alias

show buffers

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

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

```
show buffers [type number] | all
show buffers [interface [type number]] [alloc [dump]]
```

Syntax Description

<i>type number</i>	(Optional) Causes a search of all buffers that have been associated with that interface type and number 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.
all	(Optional) Displays all public and interface pool information.
interface	(Optional) Displays all interface pool information. If the specified interface <i>type</i> and <i>number</i> has its own buffer pool, displays information for that pool.
<i>type number</i>	(Optional) Displays interface pool information for the specified interface <i>type</i> and <i>number</i> if the specified interface <i>type</i> and <i>number</i> has its own buffer pool. Value of <i>type</i> can be ethernet , serial , or tokenring .
alloc	(Optional) Displays a brief listing of all allocated buffers. When alloc is specified after interface <i>type number</i> , displays a brief listing of buffers allocated for that interface.
dump	(Optional) Dumps all allocated buffers. This keyword must be used with the alloc keyword, not by itself. When alloc dump is specified after interface <i>type number</i> , dumps buffers allocated for that interface.

Command Mode

EXEC

Sample Displays

The following is sample output from the **show buffers** command with no optional arguments; showing all buffer pool and interface buffer pool information:

```
CS# show buffers

Buffer elements:
    421 in free list (500 max allowed)
    409 hits, 0 misses, 0 created

Public buffer pools:
Small buffers, 104 bytes (total 50, permanent 50):
    50 in free list (20 min, 150 max allowed)
    277 hits, 0 misses, 0 trims, 0 created
Middle buffers, 600 bytes (total 25, permanent 25):
```

```

    24 in free list (10 min, 75 max allowed)
    19 hits, 0 misses, 0 trims, 0 created
Big buffers, 1524 bytes (total 50, permanent 50):
    50 in free list (5 min, 40 max allowed)
    4 hits, 0 misses, 0 trims, 0 created
Large buffers, 5024 bytes (total 0, permanent 0):
    0 in free list (0 min, 10 max allowed)
    0 hits, 0 misses, 0 trims, 0 created
Huge buffers, 18024 bytes (total 0, permanent 0):
    0 in free list (0 min, 4 max allowed)
    0 hits, 0 misses, 0 trims, 0 created

Interface buffer pools:
Fddi buffers, 5024 bytes (total 256, permanent 256):
    0 in free list (0 min, 256 max allowed)
    256 hits, 0 misses
    256 max cache size, 110 in cache
    14 buffer threshold, 0 threshold transitions
Ethernet0 buffers, 1524 bytes (total 64, permanent 64):
    16 in free list (0 min, 64 max allowed)
    48 hits, 0 misses
    16 max cache size, 16 in cache
Ethernet1 buffers, 1524 bytes (total 64, permanent 64):
    16 in free list (0 min, 64 max allowed)
    48 hits, 0 misses
    16 max cache size, 16 in cache
Serial0 buffers, 1524 bytes (total 64, permanent 64):
    16 in free list (0 min, 64 max allowed)
    48 hits, 0 misses
    16 max cache size, 16 in cache
Serial1 buffers, 1524 bytes (total 64, permanent 64):
    16 in free list (0 min, 64 max allowed)
    48 hits, 0 misses
    16 max cache size, 16 in cache

0 failures (0 no memory)

```

Table 5-18 describes significant fields shown in the display.

Table 5-18 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.
— in free list (— max allowed)	Maximum number of buffers that are available for allocation.
hits	Count of successful attempts to allocate a buffer when needed.
misses	Count of buffer allocation attempts that resulted in growing the buffer pool in order to allocate a buffer.
created	Count of new buffers created to satisfy buffer allocation attempts when the available buffers in the pool have already been allocated.
Public buffer pools	
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.
bytes	Size of this type of buffer.

show buffers

Field	Description
(total , permanent)	Total number of this type of buffer, and the number of these buffers that are permanent.
trims	Count of buffers released to the system because they were not being used. This field is displayed only for dynamic buffer pools, not interface buffer pools, which are static.
created	Count of new buffers created in response to misses. This field is displayed only for dynamic buffer pools, not interface buffer pools, which are static.
failures	Total number of allocation requests that have failed because no buffer was available for allocation; the datagram was lost. Such failures normally occur at interrupt level.
(no memory)	Number of failures because no memory was available to create a new buffer.
Interface buffer pools	Static.

The following is sample output from the **show buffers** command when an interface *type* and *number* are specified:

```

CS# show buffers ethernet 0

Ethernet0 buffers, 1524 bytes (total 64, permanent 64):
  16 in free list (0 min, 64 max allowed)
  48 hits, 0 misses
  16 max cache size, 16 in cache

```

The following is sample output from the **show buffers** command when **all** is specified:

```

CS# show buffers all

Buffer elements:

              Free      Max      Hit  Miss      Creat
              Free      Free
-----
              427      500     17553    0         0

Public buffer pools:

Pool      Buffer Total Perm Free Min  Max      Hit  Miss  Trim Creat Need Ex-
Name      Size      Total Perm Free Min  Max      Hit  Miss  Trim Creat Need Ex-
-----
Small     104      50  50  50  20  150     14141    0    0    0
Middle    600      25  25  25  10  75       36    0    0    0
Big       1524     50  50  50  5  40       18    0    0    0
Large     5024     0  0  0  0  10        0    0    0    0
Huge     18024    0  0  0  0  4         0    0    0    0

Interface buffer pools:

Pool      Buffer Total Perm Free Min  Max      Hit  Fall Cache Cache Need Ex-
Name      Size      Total Perm Free Min  Max      Hit  back Max  Free  tra
-----
Serial0   1524     96  96  31  0  96       65    0   32   32
Serial1   1524     96  96  31  0  96       65    0   32   32

```

Ethernet0 1524 96 96 32 0 96 64 0 32 32

0 failures (0 no memory)

The following is sample output from the **show buffers** command when **alloc** is specified:

CS# **show buffers alloc**

Buffer elements:

	Free	Max Free	Hit	Miss	Created
-----	427	500	18589	0	0

Public buffer pools:

Pool Name	Buffer Size	Total	Perm	Free	Min Free	Max Free	Hit	Miss	Trim	Created	Need	Extra
Small	104	50	50	50	20	150	14982	0	0	0	0	
Middle	600	25	25	25	10	75	36	0	0	0	0	
Big	1524	50	50	50	5	40	18	0	0	0	0	
Large	5024	0	0	0	0	10	0	0	0	0	0	
Huge	18024	0	0	0	0	4	0	0	0	0	0	

0 failures (0 no memory)

Address	PakAddr	Data Area	Offset	Data Size	Pool	Ref Cnt	Link Type	Enc Type	Flags (Hex)	Output Idb	Input Idb
604A8C20	604A8C40	40007550	84	0	Serial	1	0	0	0		
604B6A80	604B6AA0	40007BD4	84	0	Serial	1	0	0	0		

show cdp

To display global CDP information, including timer and hold-time information, use the **show cdp** privileged EXEC command.

show cdp

Syntax Description

This command has no arguments or keywords.

Command Mode

Privileged EXEC

Sample Display

The following is sample output from the **show cdp** command. Global CDP timer and hold-time parameters are set to the defaults of 60 and 180 seconds, respectively.

```
cs# show cdp

Global CDP information:
  Sending CDP packets every 60 seconds
  Sending a holdtime value of 180 seconds
```

Related Commands

cdp holdtime

cdp timer

show cdp entry

show cdp neighbors

show cdp entry

To display information about a neighbor device listed in the CDP table, use the **show cdp entry** privileged EXEC command.

```
show cdp entry entry-name [protocol | version]
```

Syntax Description

<i>entry-name</i>	Name of neighbor about which you want information.
protocol	(Optional) Limits the display to information about the protocols enabled on a device.
version	(Optional) Limits the display to information about the version of software running on the device.

Command Mode

Privileged EXEC

Sample Displays

The following is sample output from the **show cdp entry** command with no limits. Information about the neighbor *device.cisco.com* is displayed, including device ID, address and protocol, platform, interface, hold-time, and version.

```
cs# show cdp entry device.cisco.com

Device ID: device.cisco.com
Entry address(es):
  IP address: 198.92.68.18
  CLNS address: 490001.1111.1111.1111.00
  DECnet address: 10.1
Platform: AGS, Capabilities: Router Trans-Bridge
Interface: Ethernet0, Port ID (outgoing port): Ethernet0
Holdtime : 155 sec

Version :
GS Software (GS3), Experimental Version 10.2(10302) [asmith 161]
Copyright (c) 1986-1994 by cisco Systems, Inc.
Compiled Mon 07-Nov-94 14:34
```

The following is sample output from the **show cdp entry privilege** command. Only information about the protocols enabled on *neon-cisco.com* is displayed.

```
cs# show cdp entry device.cisco.com protocol

Protocol information for device.cisco.com :
  IP address: 198.92.68.18
  CLNS address: 490001.1111.1111.1111.00
  DECnet address: 10.1
```

show cdp entry

The following is sample output from the **show cdp entry version** command. Only information about the version of software running on *device.cisco.com* is displayed.

```
cs# show cdp entry device.cisco.com version

Version information for device.cisco.com :
  GS Software (GS3), Experimental Version 10.2(10302) [asmith 161]
  Copyright (c) 1986-1994 by cisco Systems, Inc.
  Compiled Mon 07-Nov-94 14:34
```

Related Command

show cdp neighbors

show cdp interface

To display information about the interfaces on which CDP is enabled, use the **show cdp interface** privileged EXEC command.

```
show cdp interface [type number]
```

Syntax Description

<i>type</i>	(Optional) Type of interface about which you want information.
<i>number</i>	(Optional) Number of the interface about which you want information.

Command Mode

Privileged EXEC

Sample Displays

The following is sample output from the **show cdp interface** command. Status information and information about CDP timer and hold-time settings is displayed for all interfaces on which CDP is enabled.

```
cs# show cdp interface

Serial0 is up, line protocol is up, encapsulation is SMDS
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Ethernet0 is up, line protocol is up, encapsulation is ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
```

The following is sample output from the **show cdp interface** command with an interface specified. Status information and information about CDP timer and hold-time settings is displayed for Ethernet interface 0 only.

```
cs# show cdp interface ethernet 0

Ethernet0 is up, line protocol is up, encapsulation is ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
```

show cdp neighbors

To display information about neighbors, use the **show cdp neighbors** privileged EXEC command.

```
show cdp neighbors [interface-type interface-number] [detail]
```

Syntax Description

<i>interface-type</i>	(Optional) Type of the interface connected to the neighbors about which you want information.
<i>interface-number</i>	(Optional) Number of the interface connected to the neighbors about which you want information.
detail	(Optional) Displays detailed information about a neighbor (or neighbors) including network address, enabled protocols, hold time, and software version.

Command Mode

Privileged EXEC

Sample Displays

The following is sample output from the **show cdp neighbors** command. Device ID, interface type and number, hold-time settings, capabilities, platform, and port ID information about the communication server's neighbors is displayed.

```
cs# show cdp neighbors

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP

Device ID        Local Intrfce   Holdtme    Capability Platform Port ID
device.cisco.com Eth 0          151        R T        AGS       Eth 0
device.cisco.com Ser 0          165        R T        AGS       Ser 3
```

The following is sample output from the **show cdp neighbors detail** command. Additional detail is shown about the communication server's neighbors, including network address, enabled protocols, and software version:

```
cs# show cdp neighbors detail

Device ID: device.cisco.com
Entry address(es):
  IP address: 198.92.68.18
  CLNS address: 490001.1111.1111.1111.00
  DECnet address: 10.1
Platform: AGS, Capabilities: Router Trans-Bridge
Interface: Ethernet0, Port ID (outgoing port): Ethernet0
Holdtime : 143 sec

Version :
GS Software (GS3), Experimental Version 10.2(10302) [asmith 161]
Copyright (c) 1986-1994 by cisco Systems, Inc.
Compiled Mon 07-Nov-94 14:34
```

Related Command
show cdp entry

show cdp traffic

To display traffic information from the CDP table, use the **show cdp traffic** privileged EXEC command.

show cdp traffic

Syntax Description

This command has no arguments or keywords.

Command Mode

Privileged EXEC

Sample Display

The following is sample output from the **show cdp traffic** command.

```
cs# show cdp traffic

CDP counters :
  Packets output: 94, Input: 75
  Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
  No memory: 0, Invalid packet: 0, Fragmented: 0
```

In this example, traffic information is displayed, which includes the numbers of packets sent, the number of packets received, header syntax, checksum errors, failed encapsulations, memory problems, and invalid and fragmented packet. Header syntax indicates the number of packets CDP receives that have an invalid header format.

show clock

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

```
show clock [detail]
```

Syntax Description

detail (Optional) Indicates the clock source (NTP) 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 server time is invalid.

The symbol that precedes the **show clock** display indicates the following:

Symbol	Description
*	Time is not authoritative.
(blank)	Time is authoritative.
.	Time is 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  
cs#
```

Related Commands

clock set

show calendar

show debugging

To display information about the types of CDP debugging that are enabled for your communication server, use the **show debugging** privileged EXEC command.

show debugging

Syntax Description

This command has no arguments or keywords.

Command Mode

Privileged EXEC

Sample Display

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

```
cs# show debugging

CDP:
  CDP packet info debugging is on
  CDP events debugging is on
  CDP neighbor info debugging is on

CDP-PA: Packet received from neon.cisco.com on interface Ethernet0
CDP-EV: Encapsulation on interface Serial0 failed
CDP-AD: Aging entry for neon.cisco.com, on interface Ethernet0
```

In this example, all three types of CDP debugging are enabled.

Related Commands

See the *Debug Command Reference* for more information about the CDP debugging commands.

debug cdp packets
debug cdp adjacency
debug cdp events

show ip accounting

To display the active accounting or checkpointed database or to display access-list violations, use the **show ip accounting** privileged EXEC command.

```
show ip accounting [checkpoint] [output-packets / access-violations]
```

Syntax Description

checkpoint	(Optional) Indicates that the checkpointed database should be displayed.
output-packets	(Optional) Indicates that information pertaining to packets that passed access control and were successfully routed should be displayed. This is the default value if neither output-packets nor access-violations is specified.
access-violations	(Optional) Indicates that information pertaining to packets that failed access lists and were not routed should be displayed.

Defaults

If neither the **output-packets** nor **access-violations** keyword is specified, show ip accounting displays information pertaining to packets that passed access control and were successfully routed.

Command Mode

EXEC

Usage Guidelines

To use this command, you must first enable IP accounting on a per-interface basis.

Example

The following example displays information pertaining to packets that failed access lists and were not router (see sample display for command).

```
show ip accounting access-violations
```

Sample Display

Following is sample output from the **show ip accounting** command:

```
cs# show ip accounting

      Source           Destination           Packets           Bytes
131.108.19.40        192.67.67.20           7                 306
131.108.13.55        192.67.67.20          67                2749
131.108.2.50         192.12.33.51          17                1111
131.108.2.50         130.93.2.1             5                 319
131.108.2.50         130.93.1.2            463               30991
131.108.19.40        130.93.2.1             4                 262
131.108.19.40        130.93.1.2            28                2552
```

131.108.20.2	128.18.6.100	39	2184
131.108.13.55	130.93.1.2	35	3020
131.108.19.40	192.12.33.51	1986	95091
131.108.2.50	192.67.67.20	233	14908
131.108.13.28	192.67.67.53	390	24817
131.108.13.55	192.12.33.51	214669	9806659
131.108.13.111	128.18.6.23	27739	1126607
131.108.13.44	192.12.33.51	35412	1523980
192.31.7.21	130.93.1.2	11	824
131.108.13.28	192.12.33.2	21	1762
131.108.2.166	192.31.7.130	797	141054
131.108.3.11	192.67.67.53	4	246
192.31.7.21	192.12.33.51	15696	695635
192.31.7.24	192.67.67.20	21	916
131.108.13.111	128.18.10.1	16	1137

Table 5-19 describes significant fields shown in the display.

Table 5-19 Show IP Accounting Field Descriptions

Field	Description
Source	Source address of the packet
Destination	Destination address of the packet
Packets	Number of packets transmitted from the source address to the destination address
Bytes	Number of bytes transmitted from the source address to the destination address

Following is sample output from the **show ip accounting access-violations** command:

```

cs# show ip accounting access-violations

Source          Destination      Packets    Bytes    ACL
131.108.19.40   192.67.67.20    7          306     77
131.108.13.55   192.67.67.20    67         2749    185
131.108.2.50    192.12.33.51    17         1111    140
131.108.2.50    130.93.2.1      5          319     140
131.108.19.40   130.93.2.1      4          262     77
Accounting data age is 41
    
```

Table 5-20 describes significant fields shown in the display.

Table 5-20 Show IP Accounting Access-Violation Field Descriptions

Field	Description
Source	Source address of the packet
Destination	Destination address of the packet
Packets	For accounting keyword, number of packets transmitted from the source address to the destination address For access-violations keyword, number of packets transmitted from the source address to the destination address that violated the access control list
Bytes	For accounting keyword, number of bytes transmitted from the source address to the destination address For access-violations keyword, number of bytes transmitted from the source address to the destination address that violated the access-control list

Field	Description
ACL	Number of the access list of the last packet transmitted from the source to the destination that failed an access list

Related Commands

clear ip accounting

ip accounting

ip accounting-list

ip accounting-threshold

ip accounting-transits

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:

```
cs# show logging

Syslog logging: enabled
  Console logging: disabled
  Monitor logging: level debugging, 266 messages logged.
  Trap logging: level informational, 266 messages logged.
  Logging to 131.108.2.238

SNMP logging: disabled, retransmission after 30 seconds
  0 messages logged
```

Table 5-21 describes significant fields shown in the display.

Table 5-21 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 [*type*] [*free*]

Syntax Description

type (Optional) Memory type to display (**processor, multibus, io, sram**). If *type* is not specified, statistics for all memory types present in the communication server will be displayed.

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    0      0      84352    *Init*
2E1848   2052  2E0FF8 2E204C  1    0      0      86184    *Init*
2E204C    564  2E1848 2E2280  1    0      0      861B0    *Init*
2E2280   2052  2E204C 2E2A84  1    0      0      1266     *Init*
2E2A84    308  2E2280 2E2BB8  1    0      0      44974    *Init*
2E2BB8    220  2E2A84 2E2C94  1    0      0      3F788    *Init*
2E2C94   2052  2E2BB8 2E3498  1    0      0      3F7A8    *Init*
2E3498   4052  2E2C94 2E446C  1    0      0      46770    *Init*
2E446C    516  2E3498 2E4670  1    0      0      44E4C    *Packet Buffer*
2E4670    516  2E446C 2E4874  1    0      0      44E4C    *Packet Buffer*
2E4874    516  2E4670 2E4A78  1    0      0      44E4C    *Packet Buffer*
2E4A78    516  2E4874 2E4C7C  1    0      0      44E4C    *Packet Buffer*
2E4C7C    516  2E4A78 2E4E80  1    0      0      44E4C    *Packet Buffer*
2E4E80    516  2E4C7C 2E5084  1    0      0      44E4C    *Packet Buffer*
2E5084    516  2E4E80 2E5288  1    0      0      44E4C    *Packet Buffer*
2E5288    516  2E5084 2E548C  1    0      0      44E4C    *Packet Buffer*
2E548C    516  2E5288 2E5690  1    0      0      44E4C    *Packet Buffer*
2E5690    516  2E548C 2E5894  1    0      0      44E4C    *Packet Buffer*
```

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 5-22 describes significant fields shown in the first section of the display.

Table 5-22 Show Memory Field Descriptions—Summary Statistics

Field	Description
Head	Hexadecimal address of the head of the memory allocation chain
FreeList	Hexadecimal address of the base of the free list
Total (b)	Sum of used bytes plus free bytes
Used (b)	Amount of memory in use
Free (b)	Amount of memory not in use
Largest (b)	Size of largest available free block

The second section of the display is a block-by-block listing of memory use. Table 5-23 describes significant fields shown in the second section of the display.

Table 5-23 Show Memory Field Descriptions—Block Characteristics

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)

Field	Description
Ref	Reference count for that memory block, indicating how many different processes are using that block of memory
PrevF	Address of previous free block (if free)
NextF	Address of next free block (if free)
Alloc PC	Address of the system call that allocated the block
What	Name of process that owns the block, or "(fragment)" if the block is a fragment, or "(coalesced)" if the block was coalesced from adjacent free blocks

The **show memory io** command displays the free IO memory blocks. On the Cisco 2500, this command quickly shows how much unused IO memory is available.

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

```

csl# show memory io

I/O memory

  Address  Bytes Prev.   Next   Ref  PrevF  NextF  Alloc PC  What
100000     212  0       1000F8    1           3000F2C *Packet Data*
1000F8     212 100000   1001F0    1           3000F2C *Packet Data*
1001F0     212 1000F8   1002E8    1           3000F2C *Packet Data*
.          .      .       .         .           .         .
.          .      .       .         .           .         .
.          .      .       .         .           .         .
14AB94     4528 14A510   14BD68    0  146134  0       0          (fragment)
14BD68     1632 14AB94   14C3EC    1           3001C74 *Packet Data*
14C3EC    736240 14BD68    0         0  0       0          (fragment)

```

The **show memory** command on the Cisco 2500 includes information about processor and IO memory, and appears as follows:

```

csl# show memory

  Head  FreeList  Total(b)  Used(b)  Free(b)  Largest(b)
Processor  66ABC     2DD1C    628036   579460   48576     36096
  I/O    100000   32A14    1048576  179192   869384   736240

Processor memory

  Address  Bytes Prev.   Next   Ref  PrevF  NextF  Alloc PC  What
66ABC     2408  0       67448    1           30196A0 TTY data
67448     2000 66ABC   67C3C    1           301B640 TTY Input Buf
.          .      .       .         .           .         .
.          .      .       .         .           .         .
.          .      .       .         .           .         .
14AB94     4528 14A510   14BD68    0  146134  14542C  0          (fragment)
14BD68     1632 14AB94   14C3EC    1           3001C74 *Packet Data*
14C3EC    736240 14BD68    0         0  0       0          (fragment)
csl#

```

show ntp associations

To show the status of Network Time Protocol (NTP) associations, use the **show ntp associations EXEC** command.

show ntp associations [detail]

Syntax Description

detail (Optional) Shows detailed information about each NTP association.

Command Mode

EXEC

Sample Displays

Detailed descriptions of the information displayed by this command can be found in the NTP specification (RFC 1305).

The following is sample output from the **show ntp associations** command:

```
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
```

Table 5-24 describes significant fields shown in the display.

Table 5-24 Show NTP Associations Field Descriptions

Field	Description
(leading characters in display lines)	The first characters in a display line can be one or more of the following characters: * Synchronized to this peer # Almost synchronized to this peer + Peer selected for possible synchronization - Peer is a candidate for selection ~ Peer is statically configured
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 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 5-25 describes significant fields shown in the display.

Table 5-25 Show NTP Associations Detail Field Descriptions

Field	Descriptions
configured	Peer was statically configured.
dynamic	Peer was dynamically discovered.
our_master	Local machine is synchronized to this peer.
selected	Peer is selected for possible synchronization.
candidate	Peer is a candidate for selection.
sane	Peer passes basic sanity checks.
insane	Peer fails basic sanity checks.
valid	Peer time is believed to be valid.
invalid	Peer time is believed to be invalid.

show ntp associations

Field	Descriptions
leap_add	Peer is signaling that a leap second will be added.
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.
filtererror	Approximate error of each sample.

show ntp status

To show the status of Network Time Protocol (NTP), use the **show ntp status** EXEC command.

show ntp status

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Sample Display

The following is sample output from the **show ntp status** command:

```
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
```

Table 5-26 shows the significant fields in the display.

Table 5-26 Show NTP Status Field Descriptions

Field	Description
synchronized	System is synchronized to an NTP peer.
unsynchronized	System is not synchronized to any NTP peer.
stratum	NTP stratum of this system.
reference	Address of peer we are synchronized to.
nominal freq	Nominal frequency of system hardware clock.
actual freq	Measured frequency of system hardware clock.
precision	Precision of this system's clock (in Hz).
reference time	Reference timestamp.
clock offset	Offset of our clock to synchronized peer.
root delay	Total delay along path to root clock.
root dispersion	Dispersion of root path.
peer dispersion	Dispersion of synchronized peer.

show privilege

To display your current level of privilege, use the **show privilege EXEC** command.

show privilege

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Sample Display

The following is sample output from the **show privilege** command. The current privilege level is 15.

```
cs# show privilege

Current privilege level is 15
```

Related Command

enable password level

show processes

Use the **show processes EXEC** command to display information about the active processes.

show processes [cpu]

Syntax Description

cpu (Optional) Displays detailed CPU utilization statistics.

Command Mode

EXEC

Sample Displays

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

```
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
```

show processes

14	540	3466	155	0%	0%	0%	TTY Background
15	0	1	0	0%	0%	0%	BGP I/O
16	5100	1367	3730	0%	0%	0%	IGRP Router
17	88	4232	20	2%	1%	0%	BGP Router
18	152	14650	10	0%	0%	0%	BGP Scanner
19	224	99	2262	0%	0%	1%	Exec

Table 5-27 describes significant fields shown in the two displays. In the first line of the 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.

Table 5-27 Show Processes Field Descriptions

Field	Description
five seconds	CPU utilization by task in last 5 seconds.
one minute	CPU utilization by task in last minute.
five minutes	CPU utilization by task in last 5 minutes.
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.

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 5-28 describes significant fields shown in the display.

Table 5-28 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 global and interface-specific status of any configured Level 3 protocol such as IP or IPX.

show protocols

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Sample Display

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

```
cs# show protocols

Global values:
  Internet Protocol routing is enabled
  X.25 routing is enabled
Ethernet 0 is up, line protocol is up
  Internet address is 131.108.1.1, subnet mask is 255.255.255.0
Serial 0 is up, line protocol is up
  Internet address is 192.31.7.49, subnet mask is 255.255.255.240
Ethernet 1 is up, line protocol is up
  Internet address is 131.108.2.1, subnet mask is 255.255.255.0
Serial 1 is down, line protocol is down
  Internet address is 192.31.7.177, subnet mask is 255.255.255.240
```

For more information on the parameters or protocols shown in this sample output, refer to the *Access and Communication Servers Configuration Guide*.

show queueing

To list the current state of the queue lists, use the **show queueing** privileged EXEC command.

```
show queueing [custom | priority]
```

Syntax Description

custom (Optional) Shows status of custom queue lists.

priority (Optional) Shows status of priority lists.

Command Mode

Privileged EXEC

Usage Guidelines

If no keyword is entered, this command show the status of both custom and priority queue lists.

Sample Display

The following is sample output from the **show queueing custom** EXEC command:

```
cs# show queueing custom

Current custom queue configuration:

List  Queue  Args
3     10     default
3     3      interface Tunnel3
3     3      protocol ip
3     3      byte-count 444 limit 3
```

Related Commands

custom-queue-list

priority-group

priority-list interface

priority-list queue-limit

queue-list default

queue-list interface

queue-list protocol

queue-list queue byte-count

queue-list queue limit

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
```

Related Command

snmp-server chassis-id

show stacks

Use the **show stacks** EXEC command to monitor the stack utilization of processes and interrupt routines.

show stacks

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Usage Guidelines

The display from this command 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 your technical support representative in analyzing crashes in the field. It is included here in case you need to read the displayed statistics to an engineer over the phone.

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 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
Wakeup:          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
RTO: 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 5-29 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 5-29 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	<p>A connection progresses through a series of states during its lifetime. These states follow in the order in which a connection progresses through them.</p> <ul style="list-style-type: none"> • 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. <p>For more information, see RFC 793, <i>Transmission Control Protocol Functional Specification</i>.</p>
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.
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.

Field	Description
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 acknowledgment 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 5-30 describes the fields in the preceding lines of output.

Table 5-30 Show TCP Field Descriptions—Second Section of Output

Field	Description
Timer:	The names of the timers in the display.
Starts:	The number of times the timer has been started during this connection.
Wakeups:	Number of keepalives 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 that 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 5-31 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 5-31 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 5-32 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 5-32 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 5-33 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 5-33 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-policy

To create or update an access policy, use the **snmp-server access-policy** global configuration command. Use the **no** form of this command to remove the specified access policy.

```
snmp-server access-policy destination-party source-party context
privileges [volatile]
no snmp-server access-policy destination-party source-party context
```

Syntax Description

<i>destination-party</i>	<p>Name of a previously defined party identified as the destination party or target for this access policy. This name serves as a label used to reference a record defined for this party through the snmp-server party command.</p> <p>A destination party performs management operations that are requested by a source party.</p>
<i>source-party</i>	<p>Name of a previously defined party identified as the source party or subject for this access policy. This name serves as a label used to reference a record defined for this party through the snmp-server party command. A source party sends communications to a destination party requesting the destination party to perform management operations.</p>
<i>context</i>	<p>Name of a previously defined context that defines the resources for the access policy. This name serves as a label used to reference a record defined for this context through the snmp-server context command. A context identifies object resources accessible to a party.</p>
<i>privileges</i>	<p>Bit mask representing the access privileges that govern the management operations that the source party can ask the destination party to perform. Use decimal or hexadecimal format to specify privileges as a sum of values in which each value specifies an SNMP PDU type that the source party can use to request an operation. The decimal values are defined as follows:</p> <ul style="list-style-type: none"> • Get = 1 • GetNext = 2 • Response = 4 • Set = 8 • SNMPv1-Trap = 16 • GetBulk = 32 • SNMPv2-Trap = 128
volatile	<p>(Optional) Indicates that the access policy will not be written to nonvolatile memory when the write memory command is given or to the terminal when the write terminal command is given.</p>

Command Mode

Global configuration

Usage Guidelines

An access policy defines the management operations the destination party can perform in relation to resources defined by the specified context when requested by the source party. Access policies are defined on the router for communications from the manager to the agent; in this case, the agent is the destination party and the manager is the source party. Access policies can also be defined on the router for Response message and trap message communication from the agent to the manager; in this case, the manager is the destination party and the agent is the source party.

The *privileges* argument specifies the types of SNMP operations that are allowed between the two parties. There are seven types of SNMP operations. The bitmask identifies the commands that the source party can send to the destination party. These commands are sent in the form of messages from the source to the destination.

To remove an access-policy entry, all three arguments specified as command arguments must match exactly the values of the entry to be deleted. A difference of one value constitutes a different access policy.

The first snmp-server command that you enter enables both versions of SNMP.

Examples

The following example configures an access policy providing the manager with read-only access to the agent:

```
snmp-server access-policy agt1 mgr1 ctx1 0x23
```

The following example configures an access policy providing the manager with read-write access to the agent:

```
snmp-server access-policy agt2 mgr2 ctx2 43
```

The following example configures an access policy that allows responses and SNMP v.2 traps to be sent from the agent to a management station:

```
snmp-server access-policy mgr1 agt1 ctx1 132
```

The following example removes the access policy configured for the destination party named *agt1*, the source party named *mgr1*, and with a context named *ctx1*.

```
no snmp-server access-policy agt1 mgr1 ctx1
```

Related Commands

snmp-server party

snmp-server context

snmp-server chassis-id

To provide a message line identifying the SNMP server serial number, use the **snmp-server chassis-id** global configuration command. Use the **no** form of this command to restore the default value, if any.

```
snmp-server chassis-id text  
no snmp-server chassis-id
```

Syntax Description

text Message you want to enter to identify the chassis serial number

Default

On hardware platforms where the serial number can be machine-read, the default is the serial number.

Command Mode

Global configuration

Usage Guidelines

The Cisco MIB provides a chassis MIB variable that enables the SNMP manager to gather data on system card descriptions, chassis type, chassis hardware version, chassis ID string, software version of ROM monitor, software version of system image in ROM, bytes of processor RAM installed, bytes of nonvolatile memory installed, bytes of nonvolatile memory 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.

Use the **show snmp** command to see the chassis ID message.

Example

In the following example, the chassis serial number specified is 1234456:

```
snmp-server chassis-id 1234456
```

Related Command

show snmp

snmp-server community

To set up the community access string to permit access to the SNMPv1 protocol, use the **snmp-server community** global configuration command. To remove the specified community string, use the **no** form of this command.

```
snmp-server community string [RO | RW] [number]  
no snmp-server community string
```

Syntax Description

<i>string</i>	Community string that acts like a password and permits access to the SNMP protocol.
RO	(Optional) Specifies read-only access. Authorized management stations are only able to retrieve MIB objects.
RW	(Optional) Specifies read-write access. Authorized management stations are able to both retrieve and modify MIB objects.
<i>number</i>	(Optional) Integer from 1 to 99 that specifies an access list of IP addresses that may use the community string to gain access to the SNMP v.1 agent.

Default

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

Command Mode

Global configuration

Usage Guidelines

For the previous version of this command, the *string* argument was optional. The *string* argument is now required. However, to prevent errors and provide backward-compatibility, if the string option is omitted, a default value of public is assumed.

The **no snmp-server** command disables both versions of SNMP (SNMP v.1 and SNMP v.2).

The first snmp-server command that you enter enables both versions of SNMP.

Example

The following example assigns the string *comaccess* to SNMP v.1 allowing read-only access and specifies that IP access list 4 can use the community string:

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

The following example disables both versions of SNMP:

```
no snmp-server
```

Related Command

snmp-server party

snmp-server contact

To set the system contact (syscontact) string, use the **snmp-server contact** global configuration command. Use the **no** form of this command to remove the system contact information.

```
snmp-server contact text  
no snmp-server contact
```

Syntax Description

text String that describes the system contact information

Default

No syscontact string is set.

Command Mode

Global configuration

Example

The following is an example of a syscontact string:

```
snmp-server contact Dial System Operator at beeper # 27345
```

snmp-server context

To create or update a context record, use the **snmp-server context** global configuration command. To remove the specified context entry, use the **no** form of this command.

```
snmp-server context context-name context-oid viewname [volatile]  
no snmp-server context context-name
```

Syntax Description

<i>context-name</i>	Name of the context to be created or updated. This name serves as a label used to reference a record for this context.
<i>context-oid</i>	Object identifier to assign to the context. Specify this value in dotted decimal notation, with an optional text identifier; for example, 1.3.6.1.6.3.3.1.4.131.108.45.11.1(==initialContextId.131.108.45.11.1).
<i>viewname</i>	Name of a previously defined view. The view defines the objects available to the context.
volatile	(Optional) Indicates that the entry identified by <i>context-name</i> will not be written to nonvolatile memory when the write memory command is given, or to the terminal when the write terminal command is given.

Command Mode

General configuration

Usage Guidelines

A context record identifies object resources accessible to a party. A context record is one of the components that make up an access policy. Therefore, you must configure a context record before you can create an access policy that includes the context. Context records and party records further codify MIB views.

To remove a context entry, specify only the name of the context. The name identifies the context to be deleted.

The first `snmp-server` command that you enter enables both versions of SNMP.

Example

The following example shows how to create a context that includes all objects in the MIB-II subtree using a previously defined view named *mib2*:

```
snmp-server context mycontext initialContextid.131.108.24.56.3 mib2
```

Related Commands

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

snmp-server view

write memory[†]

write terminal[†]

snmp-server host

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

```
snmp-server host host community-string [envmon] [framerelay] [sdlc] [snmp] [tty] [x25]  
no snmp-server host host community-string [envmon] [framerelay] [sdlc] [snmp] [tty] [x25]
```

Syntax Description

<i>host</i>	Name or Internet address of the host.
<i>community-string</i>	Password-like community string to send with the trap operation.
envmon	(Optional) Enables Cisco enterprise-specific environmental monitor traps to be sent to the trap receiver <i>host</i> when an environmental threshold has been exceeded.
framerelay	(Optional) Enables Frame Relay traps to be sent to the trap receiver <i>host</i> .
sdlc	(Optional) Enables SDLC traps to be sent to the trap receiver <i>host</i> .
snmp	(Optional) Enables the SNMP traps defined in RFC 1157.
tty	(Optional) Enables Cisco enterprise-specific traps when a TCP connection closes.
x25	(Optional) Enables X.25 event traps to be sent to <i>host</i> .

Default

No traps are sent.

If you enter the command with no keywords, the default is to enable all trap types.

Command Mode

Global configuration

Usage Guidelines

The **snmp-server host** command specifies which host or hosts should receive SNMP traps. You need to issue the **snmp-server host** command once for each host acting as a trap recipient. When multiple **snmp-server host** commands are given, the community string in the last command is used, and in general, the trap types set in the last command will be used for all SNMP trap operations.

Examples

The following example sends the SNMP traps defined in RFC 1157 to the host specified by the name `cisco.com`. The community string is defined as the string `comaccess`.

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

The following example sends the SNMP and Cisco enterprise-specific traps to address 131.108.2.160:

```
snmp-server host 131.108.2.160
```

Related Command

snmp-server trap-timeout

snmp-server location

To set the system location string, use the **snmp-server location** global configuration command. Use the **no** form of this command to remove the location string.

snmp-server location *text*
no snmp-server location

Syntax Description

text String that describes the system location information

Default

No system location string is set.

Command Mode

Global configuration

Example

The following example illustrates a system location string:

```
snmp-server location Building 3/Room 214
```


snmp-server packetsize

To establish control over the largest SNMP packet size permitted when the SNMP server is receiving a request or generating a reply, use the **snmp-server packetsize** global configuration command. Use the **no** form of this command to restore the default value.

```
snmp-server packetsize byte-count  
no snmp-server packetsize
```

Syntax Description

byte-count Integer byte count from 484 to 8192

Default

484 bytes

Command Mode

Global configuration

Example

The following example establishes a packet filtering of a maximum size of 1024 bytes:

```
snmp-server packetsize 1024
```

snmp-server party

To create or update a party record, use the **snmp-server party** global configuration command. To remove a specific party entry, use the **no** form of the command.

```
snmp-server party partyname party-oid [protocol-address] [packetsize size]
    [local | remote] [authentication md5 key [clock clock]
    [lifetime lifetime] | snmpv1 string] [volatile]

no snmp-server party partyname
```

Syntax Description

<i>party-name</i>	Name of the party characterized by the contents of the record. This name serves as a label used to reference the party record that you are creating or modifying.
<i>party-oid</i>	Object identifier to assign to the party. Specify this value in dotted decimal notation, with an optional text identifier; for example, 1.3.6.1.6.3.3.1.3.131.108.34.54.1 (= initialPartyId.131.108.34.54.1)
<i>protocol-address</i>	(Optional) Address of the protocol that the party record pertains to. Currently the only supported protocol is UDP, so this value specifies a UDP address in the format <i>a.b.c.d port</i> . In future releases, additional protocols will be supported. This value is used to specify the destination of trap messages.
packetsize <i>size</i>	(Optional) Maximum size in bytes of a message that this party is able to receive. By default, the packet size set through the snmp-server packetsize command is used.
local remote	(Optional) Indicates that the party is local or remote. If neither local nor remote is specified, a default value of local is assumed.
authentication	(Optional) Indicates that the party uses an authentication protocol. If specified, either md5 or snmpv1 is required.
md5 <i>key</i>	Indicates that the party uses the Message Digest algorithm MD5 for message authentication. If md5 is specified, you must also specify a 16-byte hexadecimal ASCII string representing the MD5 authentication key for the party. If specified with the keyword md5 , all messages sent to this party will be authenticated using the SNMP v.2 MD5 authentication method with the key specified by <i>key</i> .
clock <i>clock</i>	(Optional) Initial value of the authentication clock.
lifetime <i>lifetime</i>	Lifetime, in seconds, that represents the upper bound on acceptable delivery delay for messages generated by the party.

snmpv1 <i>string</i>	Community string. The keyword snmpv1 indicates that the party uses community-based authentication. All messages sent to this party will be authenticated using the SNMP v.1 community string specified by <i>string</i> instead of MD5.
volatile	(Optional) Indicates that the entry identified by <i>party-name</i> will not be written to nonvolatile memory when the write memory command is given, or to the terminal when the write terminal command is given.

Defaults

If neither **local** nor **remote** is specified to indicate the location of the party, the party is assumed to be local.

If you do not specify a packet size value the packet size set through the **snmp-server packetsize** command is used.

Command Mode

General configuration

Usage Guidelines

You define parties to identify managers and agents. An SNMP v.2 party identity is unique; it includes the logical network location of the party, characterized by the transport protocol domain and transport addressing information, and, optionally, an authentication method and its arguments. The authentication protocol reliably identifies the origin of all messages sent by the party. The authentication protocol also ensures the integrity of the messages; in other words, it ensures that the message received is the message that was sent.

Specifying **md5** as the authentication method implies that this party record pertains to an SNMP v.2 party.

Specifying **snmpv1** as the authentication method implies that this party record pertains to an SNMP v.1 party. This allows a management station that supports only SNMP v.1 to use SNMP v.2 MIB views. Instead of using the **snmp-server community** command, you can use the **snmp-server party** command with the **snmpv1** keyword to define an SNMP v.1 party to be used to communicate with an SNMP v.1 management station. The **snmp-server community** command does not allow you to create MIB views for an SNMP v.1 management station.

If authentication is not specified, the party record pertains to an SNMP v.2 party, and no authentication will be performed for messages sent to this party.

To remove a party record, specify only the name of the party. The name identifies the party to be deleted.

The first snmp-server command that you enter enables both versions of SNMP.

Examples

The following example configures a remote unauthenticated party:

```
snmp-server party mgr1 initialPartyId.131.108.45.32.3 udp 131.108.45.76 162
```

The following example configures a local MD5-authenticated party with a large maximum packet size (You enter this command as a single line.):

```
snmp-server party agt1 initialPartyId.131.108.45.32.4 packetsize 1500 local
authentication md5 23de457623900ac3ef568fcb236589 lifetime 400
```

The following example configures an SNMP v.1 proxy party for the community *public*:

```
snmp-server party proxyv1 initialPartyId.131.108.45.32.100 authentication snmpv1 public
```

The following example removes the party named *mgr1*.

```
no snmp-server party mgr1
```

Related Commands

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

snmp-server community

write memory[†]

write terminal[†]

snmp-server queue-length

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

snmp-server queue-length *length*

Syntax Description

length Integer that specifies the number of trap events that can be held before the queue must be emptied

Default

10 events

Command Mode

Global configuration

Usage Guidelines

This command defines the length of the message queue for each trap host. Once a trap message is successfully transmitted, software will continue to empty the queue, but never faster than at a rate of four trap messages per second.

Example

The following example establishes a message queue that traps four events before it must be emptied:

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

snmp-server system-shutdown

To use the SNMP message reload feature, the device configuration must include the **snmp-server system-shutdown** global configuration command. Use the **no** form of this command to prevent an SNMP system-shutdown request (from an SNMP manager) from resetting the Cisco agent.

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

Syntax Description

This command has no arguments or keywords.

Default

This command is not included in the configuration file.

Command Mode

Global configuration

Example

The following example illustrates how to include the SNMP message reload feature in the device configuration:

```
snmp-server system-shutdown
```

snmp-server trap-authentication

To establish trap message authentication, use the **snmp-server trap-authentication** global configuration command. To remove message authentication, use the **no** form of this command.

```
snmp-server trap-authentication [snmp1 | snmp2]  
no snmp-server trap-authentication [snmp1 | snmp2]
```

Syntax Description

snmpv1	(Optional) Indicates that SNMP authentication traps will be sent to SNMP v.1 management stations only.
snmpv2	(Optional) Indicates that SNMP authentication traps will be sent to SNMP v.2 management stations only.

Defaults

Specifying the **snmp-server trap-authentication** command without a keyword turns on trap message authentication. In this case, messages are sent to the host that is specified through the **snmp-server host** command and to any SNMP stations configured through access policies to receive trap messages.

Command Mode

Global configuration

Usage Guidelines

Specify the **snmpv1** or **snmpv2** keyword to indicate the type of management stations to send the trap messages to.

This command enables the router as an agent to send a trap message when it receives an SNMP v.1 packet with an incorrect community string or an SNMP v.2 packet with an incorrect MD5 authentication key.

The SNMP specification requires that a trap message be generated for each packet with an incorrect community string or authentication key; however, because this action can result in a security breach, the router (as an agent) by default does not send a trap message when it receives an incorrect community string or authentication key.

The community string or key is checked before any access list that may be set, so it is possible to get spurious trap messages. In other words, if you have issued an **snmp-server community** command with a specified access list, you may receive messages that come from someone that is not on the access list; in this case, an authentication trap is issued. The only workarounds are to disable trap authentication or to configure an access list on a router between the SNMP agent and the SNMP manager to prevent packets from getting to the SNMP agent.

To turn off all message authentication traps, use the **no snmp-server trap-authentication** without a keyword. To turn off message authentication traps only for SNMP v.1 stations or only for SNMP v.2 stations, give the negative form of the command with the appropriate keyword.

The first snmp-server command that you enter enables both versions of SNMP.

Example

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

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

Related Command

snmp-server host

snmp-server trap-source

To specify the interface (and hence the corresponding IP address) that an SNMP trap should originate from, use the **snmp-server trap-source** global configuration command. Use the **no** form of this command to remove the source designation.

```
snmp-server trap-source interface  
no snmp-server trap-source
```

Syntax Description

interface Interface from which the SNMP trap originates. The argument includes the interface type and number in platform-specific syntax.

Default

No interface is specified.

Command Mode

Global configuration

Usage Guidelines

When an SNMP trap is sent from a Cisco SNMP server, it has a trap address of whatever interface it happened to go out of at that time. Use this command if you want to use the trap address to trace particular needs.

Example

The following example specifies that the IP address for Ethernet interface 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

snmp-server userid

To create or update an SNMP v.2 security context using the simplified security conventions method, use the **snmp-server userid** global configuration command. To remove a specified security context, use the **no** form of the command.

```
snmp-server userid user-id [view view-name] [RO | RW] [password password]  
no snmp-server userid user-id
```

Syntax Description

<i>user-id</i>	User ID name that identifies an approved SNMP v.2 user. The user ID represents a set of security information for this user. This value can identify a particular user of the system or a background process.
<i>view-name</i>	(Optional) View to be used for this security context. The argument <i>view-name</i> must be the name of a predefined view. For authenticated users, defaults to the predefined view <i>everything</i> . For users who are not authenticated, defaults to the predefined view <i>restricted</i> .
RO	(Optional) Specifies read-only access. This is the default for unauthenticated users.
RW	(Optional) Specifies read-write access. This is the default for authenticated users.
password <i>password</i>	(Optional) If specified, indicates that this is an authenticated user, and defines the password used to authenticate the user. The password must be at least eight characters long.

Defaults

For the **snmp-server userid** command, the default value for the *view-name* argument depends on whether the security context is password protected. Depending on whether the security context is password protected, one of the following default values applies:

- If the security context is password protected (meaning the user is authenticated), the default value for *view-name* is *everything*. *Everything* is a predefined value indicating that the user can see all objects.
- If the security context is not password protected (meaning that the user is not authenticated), the default value for *view-name* is *restricted*. *Restricted* is a predefined value indicating that the user can see three groups: system, snmpStats, and snmpParties.

These predefined views are described in RFC 1447.

Read-only access is the default for unauthenticated users.

Read-write access is the default for authenticated users.

Command Mode

Global configuration

Usage Guidelines

The **snmp-server userid** command implements the *simplified security conventions* method of configuring the relationship between an agent and a manager. It provides a single-step method that offers an alternative to the access policy configuration method of defining this relationship. The simplified method offers ease-of-use at the cost of forfeiting control over certain values that can be configured if you create an access policy. The simplified security conventions method applies to a configuration in which the agent is the destination or recipient of messages and the manager is the source or sender of messages. You cannot use this command to define a relationship in which the agent is the source and the manager is the destination. The security context created does not apply to trap messages.



Caution Use the simplified security conventions configuration method only if the management station participating in the manager-agent relationship also supports this method.

If you provide a password, the password is encrypted on write operations for which encryption is enabled.

If you use the **snmp-server userid** command, the SNMP v.2 implementation assumes default values that it determines internally for required information that you cannot provide through the command interface. SNMP v.2 uses the following methods to determine these values:

- To create the context, it constructs the *context-oid* from the agent's IP address and the *user-id* supplied as an argument to the **snmp-server userid** command.
- To create a party record for the agent, it constructs the *party-oid* from the agent's IP address and the *user-id* supplied as an argument to the **snmp-server userid** command. It assumes that the agent is **local**. If the user is authenticated—indicated by a password argument supplied on the **snmp-server userid** command—it constructs an MD5 key from the password.
- To create a party record for the manager, it constructs the *party-oid* from the agent's address and the *user-id* supplied as an argument to the **snmp-server userid** command. It assumes that the agent is **remote**. If the user is authenticated—indicated by a password argument supplied on the **snmp-server userid** command—it constructs an MD5 key from the password.
- To define the privileges, it sets a bit-mask value based on whether the user has read-only (**RO**) or read-write (**RW**) access, as specified on the **snmp-server userid** command. The SNMP v.2 implementation assumes the following default values:
 - For read-only access, it sets the bit-mask to 0x23; this means that the source party can send the Get, GetNext, and GetBulk commands to the destination party.
 - For read-write access, it sets the bit-mask to 0x2B; this means that the source party can send the Get, GetNext, GetBulk, and Set commands to the destination party.

The first **snmp-server** command that you enter enables both versions of SNMP.

Example

The following example configures a security context for the user *harold*, who is unauthenticated, uses the view *default*, and has read-only access:

```
snmp-server userid harold
```

Related Commands

snmp-server access-policy

snmp-server context

snmp-server party

snmp-server view

snmp-server view

To create or update a view entry, use the **snmp-server view** global configuration command. To remove the specified SNMP server view entry, use the **no** form of this command.

```
snmp-server view view-name oid-tree {included | excluded} [volatile]
no snmp-server view view-name
```

Syntax Description

<i>view-name</i>	Label for the view record that you are updating or creating. The name is used to reference the record.
<i>oid-tree</i>	Object identifier of the ASN.1 subtree to be included or excluded from the view. To identify the subtree, specify a text string consisting of numbers, such as <i>1.3.6.2.4</i> , or a word, such as <i>system</i> . Replace a single subidentifier with the asterisk (*) wildcard to specify a subtree family; for example <i>1.3.*.4</i> .
included excluded	Type of view. Either included or excluded is required.
volatile	(Optional) Indicates that the entry identified by <i>view-name</i> will not be written to nonvolatile memory when the write memory command is given or to the terminal when the write terminal command is given.

Command Mode

Global configuration

Usage Guidelines

Two standard predefined views can be used when a view is required, instead of defining a view. One is *everything*, which indicates that the user can see all objects. The other is *default*, which indicates that the user can see three groups: system, snmpStats, and snmpParties. The predefined views are described in RFC 1447.

Other SNMP v.2 commands require a predefined view as an argument. You use this command to define the view.

The first snmp-server command that you enter enables both versions of SNMP.

Examples

The following example creates a view that includes all objects in the MIB-II subtree:

```
snmp-server view mib2 mib-2 included
```

The following example creates a view that includes all objects in the MIB-II system group and all objects in the Cisco enterprise MIB:

```
snmp-server phred system included
snmp-server view phred cisco included
```

The following example creates a view that includes all objects in the MIB-II system group except for sysServices (System 7) and all objects for interface 1 in the MIB-II interfaces group:

```
snmp-server view agon system included
snmp-server view agon system.7 excluded
snmp-server view agon ifEntry.*.1 included
```

Related Commands

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

snmp-server userid

snmp-server context

write memory[†]

write terminal[†]

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** form of this command to remove this feature and restore the default.

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

Syntax Description

count Integer that sets the number of attempts

Default

Three attempts

Command Mode

Global configuration

Example

The following example changes the login attempt to just one try:

```
tacacs-server attempts 1
```


tacacs-server authenticate

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

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

Syntax Description

connection	Configures a required response when a user makes a TCP connection.
enable	Configures a required response when a user enters the enable command.
slip	Configures a required response when a user starts a SLIP or PPP session.
always	(Optional) Performs authentication even when a user is not logged in. This option 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

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

Note This command is not used in AAA/TACACS+ and has been replaced by the **aaa authorization** command.

Example

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

```
tacacs-server authenticate connection 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** form of this command to disable the mode.

tacacs-server extended
no tacacs-server extended

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Global configuration

Usage Guidelines

Note This command initializes XTACACS. To initialize AAA/TACACS+, use the **aaa new-model** command.

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. Use the **no** form of this command to delete the specified name or address.

```
tacacs-server host name  
no tacacs-server host name
```

Syntax Description

name Name or IP address of the host

Default

No TACACS host is specified.

Command Mode

Global configuration

Usage Guidelines

You can use multiple **tacacs-server host** commands to specify multiple hosts. The software searches for the hosts in the order you specify them.

Example

The following example illustrates how to specify a TACACS host named *SCACAT*:

```
tacacs-server host SCACAT
```

Related Commands

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

```
login tacacs †  
ppp †  
slip †
```

tacacs-server key

Use the **tacacs-server key** command to set the authentication/encryption key used for all TACACS+ communications between the access server and the TACACS+ daemon. To disable the key, use the **no** form of the command.

```
tacacs-server key key  
no tacacs-server key [key]
```

Syntax Description

key The key used to set authentication and encryption. This key must match the key used on the TACACS+ daemon.

Command Mode

Global Configuration

Usage Guidelines

After enabling AAA with the **aaa new-model** command, you must set the authentication and encryption key using the **tacacs-server key** command.

The key entered must match the key used on the TACACS+ daemon. All leading spaces are ignored, spaces within and at the end of the key are not. If you use spaces in your key, do not enclose the key in double quotes unless the quotes themselves are part of the key.

Example

The following example illustrates how to set the authentication and encryption key to 'dare to go':

```
tacacs-server key dare to go
```

Related Commands

aaa new-model

tacacs-server last-resort

To cause the network server to request the privileged password as verification, or to force successful login without further input from the user, use the **tacacs-server last-resort** global configuration command. Use the **no** form of this command to restore the system to the default behavior.

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

Syntax Description

password	Allows the user to access the EXEC command mode by entering the password set by the enable command.
succeed	Allows the user to access the EXEC command mode without further question.

Default

If, when running the TACACS server, the TACACS server does not respond, the default action is to deny the request.

Command Mode

Global configuration

Usage Guidelines

Use the **tacacs-server last-resort** command to be sure that login can occur; for example, when a systems administrator needs to log in to troubleshoot TACACS servers that might be down.

Note This command is not used in AAA/TACACS+.

Example

The following example illustrates how to force successful login:

```
tacacs-server last-resort succeed
```

Related Commands

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

enable password

login (exec) †

tacacs-server notify

Use the **tacacs-server notify** global configuration command to cause a message to be transmitted to the TACACS server, with retransmission being performed by a background process for up to 5 minutes.

```
tacacs-server notify { connection [always] | enable | logout [always] | slip [always]}
```

Syntax Description

connection	Specifies that a message be transmitted when a user makes a TCP connection.
enable	Specifies that a message be transmitted when a user enters the enable command.
logout	Specifies that a message be transmitted when a user logs out.
slip	Specifies that a message be transmitted when a user starts a SLIP or PPP session.
always	(Optional) Sends a message even when a user is not logged in. This option only applies to SLIP or PPP sessions, and can be used with the connection , logout , or slip keywords.

Default

No message is transmitted to the TACACS server.

Command Mode

Global configuration

Usage Guidelines

The terminal user receives an immediate response allowing access to the feature specified. Enter one of the keywords to specify notification of the TACACS server upon the corresponding action (when user logs out, for example).

Note This command is not used in AAA/TACACS+ and has been replaced by the **aaa accounting** suite of commands.

Example

The following example sets up notification of the TACACS server when a user logs out:

```
tacacs-server notify logout
```

tacacs-server optional-passwords

To specify that the first TACACS request to a TACACS server be made *without* password verification, use the **tacacs-server optional-passwords** global configuration command. Use the **no** form of this command to restore the default.

```
tacacs-server optional-passwords  
no tacacs-server optional-passwords
```

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Global configuration

Usage Guidelines

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

Note This command is not used by AAA/TACACS+.

Example

The following example 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. Use the **no** form of this command to restore 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

Usage Guidelines

The communication server software will try all servers, allowing each one to timeout before increasing the retransmit count.

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. Use the **no** form of this command to restore 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 (user)

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

```
trace [protocol] [destination]
```

Syntax Description

<i>protocol</i>	(Optional) The only protocol currently supported is ip .
<i>destination</i>	(Optional) Destination address or host name on the command line. The default parameters for the appropriate protocol are assumed and the tracing action begins.

Default

The *protocol* argument is based on the communication server's examination of the format of the *destination* argument. For example, if the communication server finds a *destination* in IP format, the *protocol* defaults to **ip**.

Command Mode

user 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, might 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. Because this is zero, the *ICMP* packets do not make it back. When you trace the path to such a host, you might 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)
 0 DEBRIS.CISCO.COM (131.108.1.6) 1000 msec 8 msec 4 msec
 1 BARRNET-GW.CISCO.COM (131.108.16.2) 8 msec 8 msec 8 msec
 2 EXTERNAL-A-GATEWAY.STANFORD.EDU (192.42.110.225) 8 msec 4 msec 4 msec
 3 BB2.SU.BARRNET.NET (131.119.254.6) 8 msec 8 msec 8 msec
 4 SU.ARC.BARRNET.NET (131.119.3.8) 12 msec 12 msec 8 msec
 5 MOFFETT-FLD-MB.in.MIL (192.52.195.1) 216 msec 120 msec 132 msec
 6 ABA.NYC.mil (26.0.0.73) 412 msec 628 msec 664 msec
```

Table 5-34 describes the fields shown in the display.

Table 5-34 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 the communication server.
131.108.1.61	IP address of the communication server.
1000 msec 8 msec 4 msec	Round-trip time for each of the three probes that are sent.

Table 5-35 describes the characters that can appear in **trace** output.

Table 5-35 IP Trace Text Characters

Char	Description
<i>nm</i> 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)

trace (privileged)

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

```
trace [protocol] [destination]
```

Syntax Description

<i>protocol</i>	(Optional) The only protocol currently supported is ip .
<i>destination</i>	(Optional) Destination address or host name on the command line. The default parameters for the appropriate protocol are assumed and the tracing action begins.

Default

The *protocol* argument is based on the communication server's examination of the format of *destination*. For example, if the communication server finds a *destination* in IP format, the *protocol* defaults to **ip**.

Command Mode

Privileged EXEC

Usage Guidelines

The **trace** command works by taking advantage of the error messages generated by 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.

The privileged-level **trace** command differs from the user-level **trace** command in that you can use nondefault parameters and invoke an extended **trace** test by entering 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, might 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. Because this is zero, the *ICMP* packets do not make it back. When you trace the path to such a host, you might 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 ABA.NYC.mil

Type escape sequence to abort.
Tracing the route to ABA.NYC.mil (26.0.0.73)
 0 DEBRIS.CISCO.COM (131.108.1.6) 1000 msec 8 msec 4 msec
 1 BARRNET-GW.CISCO.COM (131.108.16.2) 8 msec 8 msec 8 msec
 2 EXTERNAL-A-GATEWAY.STANFORD.EDU (192.42.110.225) 8 msec 4 msec 4 msec
 3 BB2.SU.BARRNET.NET (131.119.254.6) 8 msec 8 msec 8 msec
 4 SU.ARC.BARRNET.NET (131.119.3.8) 12 msec 12 msec 8 msec
 5 MOFFETT-FLD-MB.in.MIL (192.52.195.1) 216 msec 120 msec 132 msec
 6 ABA.NYC.mil (26.0.0.73) 412 msec 628 msec 664 msec
    
```

Table 5-36 describes the fields shown in the display.

Table 5-36 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 the communication server.
131.108.1.6	IP address of the 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.

```

cs# trace

Protocol [ip]:
Target IP address: mit.edu
Source address:
Numeric display [n]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Type escape sequence to abort.
Tracing the route to MIT.EDU (18.72.2.1)
    
```

```

1 ICM-DC-2-V1.ICP.NET (192.108.209.17) 72 msec 72 msec 88 msec
2 ICM-FIX-E-H0-T3.ICP.NET (192.157.65.122) 80 msec 128 msec 80 msec
3 192.203.229.246 540 msec 88 msec 84 msec
4 T3-2.WASHINGTON-DC-CNSS58.T3.ANS.NET (140.222.58.3) 84 msec 116 msec 88 msec
5 T3-3.WASHINGTON-DC-CNSS56.T3.ANS.NET (140.222.56.4) 80 msec 132 msec 88 msec
6 T3-0.NEW-YORK-CNSS32.T3.ANS.NET (140.222.32.1) 92 msec 132 msec 88 msec
7 T3-0.HARTFORD-CNSS48.T3.ANS.NET (140.222.48.1) 88 msec 88 msec 88 msec
8 T3-0.HARTFORD-CNSS49.T3.ANS.NET (140.222.49.1) 96 msec 104 msec 96 msec
9 T3-0.ENSS134.T3.ANS.NET (140.222.134.1) 92 msec 128 msec 92 msec
10 W91-CISCO-EXTERNAL-FDDI.MIT.EDU (192.233.33.1) 92 msec 92 msec 112 msec
11 E40-RTR-FDDI.MIT.EDU (18.168.0.2) 92 msec 120 msec 96 msec
12 MIT.EDU (18.72.2.1) 96 msec 92 msec 96 msec

```

Table 5-37 describes the fields that are unique to the extended trace sequence, as shown in the display.

Table 5-37 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 can specify any combination. The trace command issues prompts for the required fields. Note that trace will place the requested options in each probe; however, there is no guarantee that all communication servers (or end nodes) will process the options.
Loose	Allows you to specify a list of nodes that must be traversed when going to the destination.
Strict	Allows you to specify a list of nodes that must be the only nodes traversed when going to the destination.
Record	Allows you to specify the number of hops to leave room for.
Timestamp	Allows you to specify the number of time stamps to leave room for.
Verbose	If you select any option, the verbose mode is automatically selected and trace prints the contents of the option field in any incoming packets. You can prevent verbose mode by selecting it again, toggling its current setting.

Table 5-38 describes the characters that can appear in **trace** output.

Table 5-38 IP Trace Text Characters

Char	Description
<i>nn</i> msec	For each node, the round-trip time in milliseconds for the specified number of probes.
*	The probe timed out.
?	Unknown packet type.
Q	Source quench.
P	Protocol unreachable.
N	Network unreachable.
U	Port unreachable.
H	Host unreachable.

Related Command

trace (user)

username

To establish a username-based authentication system at login, even though your network cannot support a TACACS service, use the **username** global configuration command.

```
username name [nopassword | password encryption-type password]
username name password secret
username name [access-class number]
username name [autocommand command]
username name [noescape] [nohangup]
```

Syntax Description

<i>name</i>	Host name, server name, user ID, or command name.
nopassword	(Optional) No password is required for this user to log in. This is usually most useful in combination with the autocommand keyword.
password	Specifies a possibly encrypted password for this username.
<i>encryption-type</i>	(Optional) A single-digit number that defines whether the text immediately following is encrypted, and, if so, what type of encryption is used. Currently defined encryption types are 0, which means that the text immediately following is not encrypted, and 7, which means that the text is encrypted using a Cisco-defined encryption algorithm.
<i>password</i>	(Optional) A password can contain embedded spaces and must be the last option specified in the username command.
<i>secret</i>	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	(Optional) Specifies an outgoing access list that overrides the access list specified in the access-class line configuration command. It is used for the duration of the user's session.
<i>number</i>	(Optional) The access list number.
autocommand	(Optional) Causes the specified command to be issued automatically after the user logs in. When the command is complete, the session is terminated. As the command can be any length and contain imbedded spaces, commands using the autocommand keyword must be the last option on the line.
<i>command</i>	(Optional) The command string.

- noescape** (Optional) Prevents a user from using an escape character on the host to which that user is connected.
- nohangup** (Optional) Prevents the communication server from disconnecting the user after an automatic command (set up with the **autocommand** keyword) has completed. Instead, the user gets another login prompt.

Default
None

Command Mode
Global configuration

Usage Guidelines

The **username** command provides username/password authentication for login purposes only. (Note that it does not provide username/password authentication for enable mode when the **enable use-tacacs** command is also used.)

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

Add a **username** entry for each remote system that the local 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.

The keyword **noescape** prevents users from using escape characters on the hosts to which they are connected.

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 serial interface 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

username
