Chapter 5
Managing the System

5

This chapter describes the EXEC commands used to monitor and troubleshoot a protocol translator. Most of the network management commands are executed at the privileged-level prompt, although there is a subset of monitoring (**show**) commands that may be entered at the user level prompt. User-level commands are noted in the descriptions.

The following information is provided in this chapter:

- Setting terminal parameters
- Managing connections
- Monitoring, debugging, and tracing system processes

Refer to "Using the EXEC Command Interpreter" in Chapter 3 for a description of system command levels and how to access them.

Changing Terminal Parameters

This section describes how to change the terminal parameters using the **terminal** commands. The capability applies only to two-step translations. The new settings temporarily override those made with the line subcommands, remaining in effect only until the user exits the system

To obtain information about the terminal configuration parameter settings for the current terminal line, use the **show terminal** command. The command syntax is:

show terminal

To display information about the active ports of the server, use the **show users** command. The command syntax is:

show users [all]

The information displayed includes the line number, connection name, idle time, and terminal location. The optional keyword, **all**, requests information for both active and inactive ports.

Some **terminal** commands use the decimal representation of an ASCII character as an argument. Refer to Appendix C, "ASCII Character Set," for ASCII-to-decimal conversion information.

To display a list of commands that you can enter to change the hardware and software parameters of the current terminal line, use the **terminal** command. The command syntax is:

terminal ?

Note: For the protocol translator, **terminal** EXEC commands apply only to two-step translations and from the system console.

Recording the Terminal Type

To record the current terminal type, use the **terminal terminal-type** EXEC command. The command syntax is:

terminal terminal-type *terminal-name*

The type in the argument *terminal-name* is passed as information to the remote host. The text *terminal-name* is used by TN3270 for display management.

Changing the Terminal Screen Length

To set the number of lines on the screen of the current terminal, use the **terminal length** EXEC command. The command syntax is:

terminal length *screen-length*

The argument *screen-length* is the desired number of lines.

The protocol translator uses this value to determine when to pause during multiple-screen outputs. The default length is 24 lines. A value of 0 (zero) disables pausing between screens of output. The screen length specified can be learned by hosts.

Changing the Terminal Screen Width

To set the number of characters (columns) on a single line of the current terminal screen, use the **terminal width** EXEC command. The command syntax is:

terminal width *columns*

The rlogin protocol uses the argument *columns* to set up terminal parameters on a remote UNIX host. Hosts can learn the specified width specified with this command.

Changing the Terminal Escape Character

To set the escape character for the current terminal line, use the **terminal escape-character** EXEC command. The command syntax is:

terminal escape-character *decimal-number*

The argument *decimal-number* is the ASCII decimal representation of the desired escape character or an escape character (Ctrl-P, for example).

Typing the escape character followed by the X key returns you to the EXEC when you are connected to another computer. The default escape characters are Ctrl-^.

To make the Break key function as the escape sequence, use the **terminal no escape-character** EXEC command. The command syntax is:

terminal no escape-character

The Break need not be followed by an "X". However, you cannot set the console escape character to Break, because the operating software interprets Break on the console as an attempt to halt the system. Depending upon the configuration register setting, console breaks will either be ignored or cause the server to shut down.

Refer to "Entering the Configuration Commands" in Chapter 4 for a complete list of escape character sequences supported by the protocol translator.

Changing the Character Padding

To set the character padding on the current terminal line, use the **terminal padding** command. The command syntax is:

terminal padding *decimal-number count*
terminal no padding *decimal-number*

The argument *decimal-number* is the ASCII decimal representation of the character. It can be any of the 127 ASCII characters, up to 255 padding characters in length.

The argument *count* is the number of NUL bytes sent after that character.

The **terminal no padding** command ends this padding after the character represented by *decimal-number.*

Changing the End-of-Line Characters

The **terminal telnet-transparent** command causes the current terminal line to send a Return (CR) as a CR followed by a NUL instead of a CR followed by a Line Feed (LF). This scheme permits interoperability with different interpretations of end-of-line handling in the Telnet protocol specification. It has the following syntax:

terminal telnet-transparent

Setting Terminal Flow Control

To set up the terminal flow control, use the **terminal flowcontrol** command. The command syntax is:

terminal flowcontrol { none | software [in | out] | hardware}

The **terminal flowcontrol** command sets the method of data flow control for the current terminal line.

The keyword **software** sets software flow control. An additional keyword specifies the direction: **in** causes the terminal server to listen to flow control from the attached device, and **out** causes the terminal server to send flow control information to the attached device. If you do not specify a direction, both directions are assumed.

Note: For software flow control, the default stop and start characters are Ctrl-S and Ctrl-Q (XOFF and XON). You can change them with the **terminal stop-character** and **terminal start-character** commands described later in this section.

By default, no flow control method is set for a line.

Setting the Packet Dispatch Character

To set up the packet dispatch character, use the **terminal dispatch-character** EXEC command. The command syntax is:

terminal dispatch-character *decimal-number1* **[***decimal-number2 … decimal-numberx***]**

The **terminal dispatch-character** command defines a character or string that causes a packet to be sent. The argument *decimal-number* is the ASCII decimal representation of the character or string.

This command causes the protocol translator to attempt to buffer characters into larger-sized packets for transmission to the remote host. The protocol translator normally dispatches each character as it is typed.

Establishing Input Notification

To establish input notification, use the **terminal notify** EXEC command. The command syntax is:

terminal notify
terminal no notify

When you have multiple concurrent connections, you may want to know when output is pending on a connection other than the current connection. For example, you may want to know when another connection receives mail or a message. The **terminal notify** command causes the protocol translator to notify you of pending output. The **terminal no notify** command ends such notifications.

Selecting File Download Mode

To set the line to maximum transparency for file transfers, use the **terminal download** EXEC command. The command syntax is:

terminal download
terminal no download

The **terminal download** command is used when running a program such as KERMIT, XMODEM, or CrossTalk that downloads a file across a protocol translator line. It sets up the terminal line as a transparent pipe that can be used to transmit data and is equivalent to entering all the following commands:

Example:

```
terminal telnet-transparent
terminal no escape-character
terminal no hold-character
terminal no pad 0
  .
  .
  .
terminal no pad 128
terminal parity none
terminal databits 8
```

The **terminal no download** command restores the line's original parameter settings.

Selecting the Preferred Terminal Transport Protocol

To select a preferred remote terminal protocol, use the **terminal transport** EXEC command. The command syntax is:

terminal transport [telnet|lat|pad|rlogin|none]

This EXEC command sets the preferred protocol for the duration of the current session.

For details on the parameters of this command, refer to "Defining the Transport Protocol for a Specific Line" in Chapter 4.

Sample Transport Commands:

This section includes sample screen outputs for the **terminal transport-preferred** commands.

The following example makes a connection using Telnet, after rlogin is refused as a preferred protocol by the remote host Eng2:

```
cpt>terminal transport rlogin
cpt>Eng2
Trying Eng2.Cisco.com (121.108.145.12)...
cpt>Eng2
Trying Eng2.Cisco.com (121.108.145.12)...Open
login:
```

The following example sets the preferred transport to **none** and then attempts to make a connection without specifying a protocol. A Telnet connection is made successfully once **telnet** is specified.

```
cpt>terminal transport none
cpt>Eng2
% unknown command "Eng2"
cpt>telnet Eng2
Trying Eng2.Cisco.com (121.108.145.12)...Open
login:
```

Displaying Debug Messages on the Console and Terminals

To enable logging of system debugging and event messages on the current terminal, use the **terminal monitor** EXEC command. The command syntax is:

terminal monitor

The **terminal monitor** command copies the system debugging and event messages to the current terminal. To use this command, you must first issue the **enable** command and enter the password to access the privileged command mode.

Maintaining the System

Use the privileged EXEC commands in this section to clear interface counters, reset hardware logic, and clear all activity on a line.

Clearing Interface Counters

To clear the interface counters shown with the EXEC **show interface** command, use the following command:

clear counters [*type unit***]**

The command clears all the current interface counters from the interface unless the optional arguments *type* and *unit* are specified to clear only a specific interface type (**serial** or **ethernet**) from a specific unit or card number.

Note: This command clears those displayed with the EXEC **show interface** command, but not counters retrieved using SNMP.

Resetting Interface Hardware Logic

Use the **clear interface** EXEC command to reset the hardware logic on a network interface. The syntax for this command is as follows:

clear interface *type unit*

The arguments *type* and *unit* specify the interface type and unit or card number (such as, Ethernet 0, Serial 0, etc.).

Clearing a Line

To reset a terminal line, use the **clear line** command. The command syntax is:

clear line *line-number*

This command aborts any connections, terminates the associated processes, and resets the data structures associated with a terminal line.

The argument *line-number* specifies the terminal line number.

Monitoring System Processes

Monitoring the system processes is accomplished by using the EXEC commands **show**.

The **show** commands display information about the network and the interfaces to aid in troubleshooting and monitoring the system.

To display the list of the **show** command options, use the **show ?** EXEC command. The command syntax is:

show ?

Sample:

The following is a sample output of the **show ?** command:

```
access-lists      Access control lists
arp <keyword>     ARP cache, type ? for list
buffers           Network buffer utilization
configuration     Display non-volatile configuration memory
controllers       Network interface controller statistics
debugging         State of debugging flags
entry             Incoming queue entries
hosts             Host/address cache
interfaces        Network interface statistics
ip <keyword>        Internet Protocol information, type ? for list
lat <keyword>     LAT Protocol information, type ? for list
line <line>       Line information, may specify a line
logging           Logging information
memory            Memory utilization statistics
printers          Parallel printer status
processes         Active system processes
rif               RIF cache
sessions          Telnet and rlogin connections
slip              SLIP statistics
stacks            Process and interrupt stack use
tcp <line>        TCP information, may specify a line
terminal          Terminal parameters
```

Displaying Buffer Statistics

To display the buffer statistics, use the **show buffers** EXEC command. The command syntax is:

show buffers *interface*

This command displays utilization statistics for the network packet buffer allocator. 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.

The optional argument *interface* will cause a search of all buffers that have been associated with that interface for longer than one minute. The contents of these buffers will be printed to the screen. This option is useful in diagnosing problems where the input queue count on an interface is consistently non-zero.

Following is sample output without the optional *interface* argument. Table 5-1 describes the fields seen.

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

0 failures (0 no memory)
```

Table 5-1 Show Buffers Field Descriptions

Field	Description
Buffer elements	Blocks of memory used in internal operating system queues
Small buffers Middle buffers Big buffers Large buffers Huge buffers	Blocks of memory used to hold network packets
hits	Count of successful attempts to allocate a buffer when needed
misses	Count of allocation attempts which failed for lack of a free buffer in the pool
created	Count of new buffers created
trims	Count of buffers destroyed
in free list	Number of buffers of a given type which are not currently allocated and are available for use
max allowed	The maximum number of buffers of a given type allowed in the system
failures	The total number of allocation requests that have failed for lack of a free buffer
no memory	Number of failures due to a lack of memory to create a new buffer

Note: The `misses` specified are not necessarily indicative of a system problem. They essentially reflect packets that are dropped.

Displaying the Contents of Non-Volatile Memory

To display the contents of the non-volatile memory, use the **show configuration** EXEC command. The command syntax is:

show configuration

This privileged command displays the contents of the non-volatile memory, only if it is present and valid. The non-volatile memory stores the configuration information in text form as configuration commands.

Example:

The following example shows screen output from the **show configuration** command:

```
Using 3203 out of 32768 bytes
!
enable-password Secret
service decimal-tty
service pad
!
boot system alpha.pt2 255.255.255.255
!
!
interface Ethernet 0
ip address 192.31.7.20 255.255.255.240
!
interface Serial 0
encapsulation X25
ip address 192.31.7.162 255.255.255.240
x25 win 7
x25 wout 7
x25 ips 512
x25 ops 512
x25 address 313700540651
clockrate 64000
!
!
domain-name CISCO.COM
ip name-server 255.255.255.255
snmp-server community
snmp-server trap-authentication
snmp-server host 131.108.1.27
tacacs-server host 192.31.7.24
tacacs-server host 131.108.1.27
!
hostname translator
!
!
line console 0
```

```
exec-timeout 0 0
login
password alsoSecret
line aux 0
no exec
login tacacs
line vty 0
login tacacs
line vty 1
login tacacs
line vty 2
login tacacs
line vty 3
login tacacs
line vty 4
login tacacs
line vty 5
login tacacs
 . . .
line vty 99
login tacacs
!
end
```

Displaying Controller Status

The **show controllers** command displays current internal status information for different interface cards. Enter this command at the EXEC prompt:

show controllers {serial | mci}

For the CPT, use the keyword **mci** or **serial** to display the information about that system's network interface card.

Sample output for the MCI controller card follows. Table 5-2 describes the fields displayed.

```
MCI 0, controller type 1.1, microcode version 1.5
  64 Kbytes of main memory, 4 Kbytes cache memory
5 system TX buffers, largest buffer size 1520
  Restarts: 0 line down, 0 hung output, 0 controller error
  Interface 0 is Ethernet0, station address 0000.0c00.d4a6
    15 total RX buffers, 11 buffer TX queue limit, buffer size 1520
    Transmitter delay is 0 microseconds
  Interface 1 is Serial0, electrical interface is V.35 DTE
    15 total RX buffers, 11 buffer TX queue limit, buffer size 1520
    Transmitter delay is 0 microseconds
    High speed synchronous serial interface
```

Table 5-2 Show Controllers Field Descriptions

Field	Description
MCI (number)	The unit number of the MCI card
controller type	The version number of the MCI card
microcode version	The version number of the MCI card's internal software (in read-only memory)
main memory cache memory	The amounts of main and cache memory on the card
system TX buffers	Number of buffers that hold packets to be transmitted
Restarts	Count of restarts due to the following conditions:
line down	Communication line down
hung output	Output unable to transmit
controller error	Internal error
Interface...is	Names of interfaces, by number
Station address	The hardware address of the interface
electrical interface	Line interface type for serial connections
RX buffers	Number of buffers for received packets
TX queue limit	Maximum number of buffers in transmit queue
Transmitter delay	Delay between outgoing frames

Note: For the IGS, information displayed relates to the LANCE network controller. Use the command **show controllers** with no card-type keyword.

Monitoring an Ethernet Interface

Use the command **show interfaces** to display information about the Ethernet interface. Enter this command at the EXEC prompt:

show interfaces [*type unit***]**

Where *type* is the **ethernet** keyword and *unit* is the interface unit number. If you do not provide values for the parameters *type* and *unit*, the command will display statistics for all the network interfaces.

Sample output of this command is provided below. Table 5-3 describes the fields seen.

```
Ethernet 0 is up, line protocol is up
  Hardware is MCI Ethernet, address is aa00.0400.0134 (bia 0000.0c00.4369)
  Internet address is 131.108.1.1, subnet mask is 255.255.255.0
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, PROBE, ARP Timeout 4:00:00
  Last input 0:00:00, output 0:00:00, output hang never
  Output queue 0/40, 0 drops; input queue 0/75, 2 drops
  Five minute input rate 61000 bits/sec, 4 packets/sec
  Five minute output rate 1000 bits/sec, 2 packets/sec
     2295197 packets input, 305539992 bytes, 0 no buffer
     Received 1925500 broadcasts, 0 runts, 0 giants
     3 input errors, 3 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     3594664 packets output, 436549843 bytes, 0 underruns
     8 output errors, 1790 collisions, 10 interface resets, 0 restarts
```

Table 5-3 Show Interface Ethernet Field Descriptions

Field	Descriptions
Ethernet ... is up ...is administratively down	Tells whether the interface hardware is currently active and if it's been taken down by an administrator.
line protocol is {up | down | administratively down}	Tells whether the software processes that handle the line protocol believe the interface is usable (are keepalives successful?).
Hardware	Specifies the hardware type (e.g. MCI Serial, MCI Ethernet) and address.
Internet address	Lists the Internet address followed by subnet mask.
Encapsulation	Encapsulation method assigned to interface.
ARP type:	Type of Address Resolution Protocol assigned.
Last input	The number of hours, minutes, and seconds since the last packet was successfully received by an interface. Useful for knowing when a dead interface failed.
output	Number of hours, minutes, and seconds since the last packet was successfully transmitted by the interface. Useful for knowing when a dead interface failed.
output hang	The number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the "last" fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed.
Output queue, Input queue, drops	Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue.
Five minute input rate, Five minute output rate	The average number of bytes and packets transmitted per second in the last five minutes.

packets input	The total number of error-free packets received by the system.
Received ..broadcasts	The total number of broadcast or multicast packets received by the interface.
runts	The number of packets which are discarded because they are smaller than the medium's minimum packet size. For instance, any Ethernet packet which is less than 64 bytes is considered a runt.
giants	The number of packets which are discarded because they exceed the medium's maximum packet size. For example, any Ethernet packet which is greater than 1518 bytes is considered a giant.
input error	Includes runts, giants, no buffer, CRC, frame, overrun, and ignored counts. Other input-related errors can also cause the input errors count to be increased, and some datagrams may have more than one error; therefore, this sum may not balance with the sum of enumerated input error counts.
CRC	The Cyclic Redundancy Checksum generated by the originating LAN station or far-end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRCs is usually the result of collisions or a station transmitting bad data.
frame	The number of packets received incorrectly having a CRC error and a non-integer number of octets. On a LAN, this is usually the result of collisions or a malfunctioning Ethernet device.
overrun	The number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data.
ignored	The number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be increased.
abort	An illegal sequence of one bits on a serial interface. This usually indicates a clocking problem between the serial interface and the data link equipment.
packets output	Total number of messages transmitted by the system.
bytes	Total number of bytes, including data and MAC encapsulation, transmitted by the system.
underruns	Number of times that the transmitter has been running faster than the router can handle. This may never happen (be reported) on some interfaces.

output errors	The sum of all errors which prevented the final transmission of datagrams out of the interface being examined. Note that this may not balance with the sum of the enumerated output errors, as some datagrams may have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories.
collisions	The number of messages retransmitted due to an Ethernet collision. This is usually the result of an overextended LAN (Ethernet or transceiver cable too long, more than two repeaters between stations, or too many cascaded multiport transceivers). A packet that collides is counted only once in output packets.
interface resets	The number of times an interface has been completely reset. This can happen if packets queued for transmission were not sent within several seconds time. On a serial line, this can be caused by a malfunctioning modem which is not supplying the transmit clock signal, or by a cable problem. If the system notices that the carrier detect line of a serial interface is up, but the line protocol is down, it periodically resets the interface in an effort to restart it. Interface resets can also occur when an interface is looped back or shut down.
restarts	The number of times a Type 2 Ethernet controller was restarted because of errors.

Monitoring a Serial Interface

Use the command **show interfaces** to display information about the serial interface and the state of source bridging. Enter this command at the EXEC prompt:

show interfaces [*type unit***]**

Where *type* is the keyword **serial** and *unit* is the interface unit number. If you do not provide values for the parameters *type* and *unit*, the command will display statistics for all the network interfaces.

Sample output of this command for Cisco's synchronous serial interfaces is provided below: Table 5-4 describes the fields seen.

```
Serial 0 is up, line protocol is up
  Hardware is MCI Serial
  Internet address is 192.31.7.50, subnet mask is 255.255.255.240
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input 0:00:00, output 0:00:00, output hang never
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 2000 bits/sec, 4 packets/sec
  Five minute output rate 1000 bits/sec, 2 packets/sec
     1131427 packets input, 87088176 bytes, 2 no buffer
     Received 312429 broadcasts, 0 runts, 0 giants
     8 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 8 abort
     753820 packets output, 41586176 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets, 0 restarts
     19 carrier transitions
```

Table 5-4 Show Interface Serial Field Descriptions

Field	Descriptions
Serial...is {up |down} ...is administratively down	Tells whether the interface hardware is currently active (whether carrier detect is present) and if it's been taken down by an administrator.
line protocol is {up | down | administratively down}	Tells whether the software processes that handle the line protocol thinks the line is usable (are keepalives successful?).
Hardware	Specifies the hardware type (e.g. MCI Serial, MCI Ethernet).
Encapsulation	Encapsulation method assigned to interface.
loopback	Tells whether loopback is set or not.
keepalive	Tells whether keepalives are set or not.
Last input	The number of hours, minutes, and seconds since the last packet was successfully received by an interface. Useful for knowing when a dead interface failed.
Last clearing	The time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared.
Output queue, Input queue, drops	Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue.
Five minute input rate, Five minute output rate	The average number of bytes and packets transmitted per second in the last five minutes.
packets input	The total number of error-free packets received by the system.
broadcasts	The total number of broadcast or multicast packets received by the interface.

runts	The number of packets which are discarded because they are smaller than the medium's minimum packet size.
giants	The number of packets which are discarded because they exceed the medium's maximum packet size.
CRC	The Cyclic Redundancy Checksum generated by the originating station or far-end device does not match the checksum calculated from the data received. On a serial link, CRCs usually indicate noise, gain hits, or other transmission problems on the data link.
frame	The number of packets received incorrectly having a CRC error and a non-integer number of octets. On a serial line, this is usually the result of noise or other transmission problems.
overrun	The number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data.
ignored	The number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be increased.
abort	An illegal sequence of one bits on a serial interface. This usually indicates a clocking problem between the serial interface and the data link equipment.
packets output	Total number of messages transmitted by the system.
bytes output	Total number of bytes, including data and MAC encapsulation, transmitted by the system.
underruns	Number of times that the transmitter has been running faster than the router can handle. This may never happen (be reported) on some interfaces.
congestion drop	The number of messages discarded because the output queue on an interface grew too long. This can happen on a slow, congested serial link.
output errors	The sum of all errors which prevented the final transmission of datagrams out of the interface being examined. Note that this may not balance with the sum of the enumerated output errors, as some datagrams may have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories.
interface resets	The number of times an interface has been completely reset. This can happen if packets queued for transmission were not sent within several seconds time. On a serial line, this can be caused by a malfunctioning modem which is not supplying the transmit clock signal, or by a cable problem. If the system notices that the carrier detect line of a serial interface is up, but the line protocol is down, it periodically resets the interface in an effort to restart it. Interface resets can also occur when an interface is looped back or shut down.

restarts	The number of times the controller was restarted because of errors.
carrier transitions	The number of times the carrier detect signal of a serial interface has changed state. Indicates modem or line problems if the carrier detect line is changing state often.

Displaying Line Status

To display a summary status of terminal lines on the protocol translator, use the **show line** EXEC command. The command syntax is:

show line [*line-number***]**

Include the optional argument *line-number* to display detailed information about a particular line.

When entered with no argument, this command displays a summary status of the virtual terminal lines on the protocol translator.

Example:

The following is sample command output:

```
  Tty Typ    Tx/Rx    A Modem  Roty AccO AccI  Uses    Noise
    0 CTY              -    -      -    -    -     0        0
    1 AUX  9600/9600   -    -      -    -    -     0        0
*   2 VTY  9600/9600   -    -      -    -    -     1        0
    3 VTY  9600/9600   -    -      -    -    -     0        0
    4 VTY  9600/9600   -    -      -    -    -     0        0
                            . . .
   97 VTY  9600/9600   -    -      -    -    -     0        0
   98 VTY  9600/9600   -    -      -    -    -     0        0
   99 VTY  9600/9600   -    -      -    -    -     0        0
  100 VTY  9600/9600   -    -      -    -    -     0        0
```

In the output, the `Tty` column lists the line number in or decimal (depending on the setting of the **service decimal-tty** global configuration command; an asterisk (*) indicates an active line. The `Typ` column identifies the line type:

- CTY is the console
- AUX is an auxiliary console line
- VTY is a virtual terminal line

The `Tx/Rx` column lists the current transmit and receive baud rates. The `A` column indicates the autobaud detect range. This column is not valid or detected on the protocol translator. The `Modem` column identifies the handling, if any, of RS-232 modem control signals. The `Roty` column lists the rotary group number, if any.

The `AccO` and `AccI` columns indicate the access classes for outgoing (Telnet and rlogin) and incoming (rotary and virtual terminal) connections, respectively. The `Uses` column shows the total number of connections made to or from the terminal line since the system was

booted. This count helps you evaluate terminal line use. The `Noise` column lists the total number of noise characters received. (A noise character is a nonactivating character received as a framing error or when the line is inactive. The default activating character is Return.)

The **show line** command entered with the argument *line-number* displays detailed parameter information about a particular line. The unprivileged **show terminal** command displays the same information for the current line.

Example:

The following is sample command output:

```
Line 42, Location: "", Type: ""
Length: 24 lines, Width: 80 columns
Baud rate (TX/RX) is 9600/9600
The escape character is "^^", followed by "x"
The local hold character is disabled
No flowcontrol in effect.
Status: Ready, Active, No Exit Banner
Capabilities: none
Idle EXEC timeout is not set.
Idle session timeout is not set.
Session limit is not set.
Modem answer timeout is 15 seconds
Dispatch timeout is not set.
Allowed transports are telnet rlogin. Preferred is telnet
Disconnect character is not set
Activation character is ^M (13)
No output characters are padded
Characters causing immediate data dispatching:
   Char    ASCII
```

Displaying the State of Error Loggings

To show the state of logging (syslog), use the following EXEC command:

show logging

This command displays the state of syslog error and event logging, including host addresses and whether console logging is enabled. This command also displays SNMP configuration parameters and protocol activity. See the section "Redirecting System Error Messages" in Chapter 4, for an explanation of how to configure message logging. Following is a sample output. Table 5-5 describes the fields seen.

```
Syslog logging: enabled
    Console logging: disabled
    Monitor logging: level debugging, 18 messages logged.
    Trap logging: level informational, 18 messages logged.
    Logging to 192.31.7.19

SNMP logging: enabled, retransmission after 30 seconds
    741 messages logged
    Logging to 131.108.1.27, 0/10
    Logging to 131.108.1.111, 0/10
    Logging to 131.108.2.63, 0/10
```

Table 5-5 Show Logging Field Descriptions

Field	Description
Syslog logging	When enabled, system logging messages are sent to a UNIX host which 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	The minimum level of severity required for a log message to be sent to a monitor terminal (not the console).
Trap logging	The minimum level of severity required for a log message to be sent to syslog server.
SNMP logging	Shows whether SNMP logging is enabled and the number of messages logged, and the retransmission interval.

Displaying System Memory Statistics

To display the activity statistics for the systems memory allocator, use the **show memory** EXEC command. The command syntax is:

show memory

This command displays memory free pool statistics. These statistics include summary information about the activities of the system memory allocator, and a block-by-block listing of memory use. Sample output follows.

```
 Head    Free Start   Total Bytes     Used Bytes      Free Bytes
Processor   AA0A8      E42D8       3497816        308700        3189116
Multibus  2000000    2000000         32768             0          32768


 --More--
Address   Bytes Prev.   Next    Free? PrevF   NextF   Alloc PC  What
AA0A8       916 0       AA43C                         7ACE      *Init*
AA43C      2024 AA0A8   AAC24                         AD2E      *Init*
AAC24       536 AA43C   AAE3C                         AD58      *Init*
AAE3C      2024 AAC24   AB624                         49BC      *Init*
AB624        72 AAE3C   AB66C                         248E0     *Init*
AB66C        44 AB624   AB698                         3614C     *Init*
AB698       152 AB66C   AB730                         1CFC      *Init*
AB730      2024 AB698   ABF18                         1D20      *Init*
ABF18       152 AB730   ABFB0                         1CFC      *Init*
ABFB0      2024 ABF18   AC798                         1D20      *Init*
AC798       100 ABFB0   AC7FC                         3F2FE     Logger
AC7FC      152 AC798   AC894    y   E2568   D74E8   74E12    TCP Protocols
AC894        44 AC7FC   AC8C0                          4BCC      *Sched*
AC8C0     1880 AC894   AD018    y   D74E8   D7134   74E3C    TCP Protocols
AD018       104 AC8C0   AD080                         5126      *Init*
AD080      2024 AD018   AD868                         67E6      *Init*
AD868       348 AD080   AD9C4                         54BA      *Init*
AD9C4       348 AD868   ADB20                         54BA      *Init*
ADB20       348 AD9C4   ADC7C                         54BA      *Init*
ADC7C       348 ADB20   ADDD8                         54BA      *Init*
ADDD8       348 ADC7C   ADF34                         54BA      *Init*
```

Table 5-6 describes the fields seen; Table 5-7 lists the characteristics of each block of memory in the system.

Table 5-6 Show Memory Field Descriptions

Field	Description
Head	The hexadecimal address of the head of the memory allocation chain
Free Start	The hexadecimal address of the base of the free list
Total Bytes	The total amount of system memory
Used Bytes	The amount of memory in use
Free Bytes	The amount of memory not in use

Table 5-7 Characteristics of Each Block of Memory

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)
Free?	Tells if the block is free
Alloc PC	Address of the system call that allocated the block
What	Name of process that owns the block

Displaying Active Processes

To see information about the active processes, use the following EXEC command:

show processes

Following is a partial display of the command output. Table 5-8 describes the fields seen.

```
CPU utilization for one minute: 38%; for five minutes: 37%

PID Q T      PC Runtime (ms)    Invoked   uSecs   Stacks  TTY Process
  1 M E   122DE        62812       4897   12826  780/1000   0 Net Background
  2 M E   22842            8         19     421  804/1000   0 Logger
809 M E   74AF0       272808     489888     556 1504/2000  36 Exec
  4 H E    67C0       373540     630248     592  628/900    0 IP Input
  5 M E   3E124        26044     630201      41  824/1000   0 IP Protocols
  6 M E   46BA2          592     255178       2  794/1000   0 TCP Timer
  7 L E   47CE6         1736       1635    1061  776/1000   0 TCP Protocols
  8 L E    67C0            0          1       0  958/1000   0 ARP Input
813 M *     768          384         93    4129 1456/2000  42 Virtual Exec
 10 M E   3F51E            0          1       0  894/1000   0 BOOTP Server
 11 H E    67C0        25096     194823     128  426/500    0 Net Input
 12 M T    36FA         5420     277303      19  850/1000   0 TTY Background
 13 L E   5444E        65996      24907    2649  686/1000   0 SNMP Server
 14 M E   6E842            0          1       0  966/1000   0 Serial Line IP
```

Table 5-8 Show Processes Field Descriptions

Field	Description
PID	Process ID
Q	Queue priority (high, medium, low)
T	Scheduler test (Event, Time, 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 invocation
Stacks	Low water mark/Total stack space available
TTY Process	Terminal that controls the process and name of process

Displaying User Sessions

To display information about your active terminal sessions, use the **show sessions** EXEC command. This command has the following syntax:

show sessions

Example:

```
Conn Host              Address            Byte  Idle Conn Name
*  1 GUN               131.222.3.11        0     0   GUN
   2 BIG               131.222.3.14        0     5   BIG
```

The information it displays includes the host name, address, number of characters waiting to be sent to the terminal, idle time, and connection name. An asterisk (*) indicates the user's current session.

Displaying Stack Utilization

To display and monitor the stack utilization of processes and interrupt routines, use the **show stacks** EXEC command. The command syntax is:

show stacks

This command monitors the stack utilization of processes and interrupt routines. The command output is of use only to Cisco Systems engineers analyzing software problems. The command is described here in case you need to issue it and read the displayed statistics to an engineer over the phone.

Displaying the Terminal Parameters Setup

To display the configuration parameter settings for the current terminal, use the EXEC **show terminal** command. The command syntax is:

show terminal

The privileged **show line** command entered with a line number displays the same information for other lines.

Example:

The following is sample command output:

```
Line 42, Location: "", Type: ""
Length: 24 lines, Width: 80 columns
Baud rate (TX/RX) is 9600/9600
The escape character is "^^", followed by "x"
The local hold character is disabled
No flowcontrol in effect.
Status: Ready, Active, No Exit Banner
Capabilities: none
Idle EXEC timeout is not set.
Idle session timeout is not set.
Session limit is not set.
Modem answer timeout is 15 seconds
Dispatch timeout is not set.
Allowed transports are telnet rlogin. Preferred is telnet
Disconnect character is not set
Activation character is ^M (13)
No output characters are padded
Characters causing immediate data dispatching:
   Char    ASCII
```

Displaying Line Information

To display line information, use either the **show users** or **systat** EXEC commands. The commands syntax are:

show users [all]
systat [all]

You can use either the **show users** or the **systat** command to display information about the active lines of the protocol translator, including the line number, connection names, idle time, and terminal location.

The optional keyword **all** displays information about both active and inactive lines.

Example:

The following is sample command output:

```
 TTY       Host(s)              Idle Location
 con0      idle
*vty2      idle                    1 CLASH.CISCO.COM
```

Displaying Hardware Configuration

To display the software version and the hardware configuration of the system, use the **show version** EXEC command. The command syntax is:

show version

Example:

The following is a sample of the command output:

```
Protocol Translator, Version 8.3
Copyright (c) 1986-1991 by cisco Systems, Inc.
Compiled Thu 01-Aug-91 13:00 by kph
System Bootstrap, Version 4.3(2)
char uptime is 2 weeks, 23 hours, 25 minutes
System restarted by reload
Software booted from 131.108.1.27
Host config file is ”translator-conf”, booted from 131.108.1.27
Network config file is ”network-confg”, booted from 131.108.1.27
CSC3 (68020) processor with 4096K bytes of memory.
Commercial X.25 software.
1 MCI controller.
1 Ethernet/IEEE 802.3 interface.
1 Serial network interface.
32K bytes of non-volatile configuration memory.
Configuration register is 0x200
```

In the output, the fifth line is the bootstrap version string. The second through fourth lines list information about the system software; the version number is on the first line. Always specify the complete version number when reporting a possible software problem. In the sample output, the version number is 8.3, initial release.

The fifth line shows the system name and uptime. The sixth line indicates the reason the system was restarted, including a response to an error.

If the software was booted over the network, the seventh line shows the Internet address of the boot host; if the software was loaded from onboard ROM, this line reads “running default software.” The eighth and ninth lines identify the names and sources of the host and network configuration files.

The remaining lines of output show the hardware configuration and any nonstandard software options. The configuration register contents are displayed in hexadecimal notation.

Troubleshooting Network Operations

The protocol translator includes software to aid in tracking down problems with the protocol translator or with other hosts on the network. The privileged EXEC command **debug** enables the display of several classes of network events on the console terminal. The privileged **undebug** command turns off the display of these classes. The EXEC command **show debugging,** displays the state of each debugging option. (Refer to the “Enabling Message Logging” section in Chapter 4 for an explanation of how to configure message logging.)

Normally, the **debug** command output goes only to the console terminal. To send a copy of this output to the current terminal, use the privileged EXEC command **terminal monitor**; use the **terminal no monitor** command to stop copying **debug** output to the line.

Note: The protocol-specific **debug** commands are outlined in the chapters focusing of each protocol in Part III, "Transmission Protocols," of this manual.

---

debug ?

The **debug ?** command displays a list of the **debug** command options.

debug all

The **debug all** command enables logging of all possible debugging options.

debug packet

The **debug packet** command enables logging of packets received with unknown packet types.

debug broadcast

The **debug broadcast** command enables logging of all broadcast traffic.

debug serial-interface

The **debug serial-interface** command enables logging of serial-interface hardware events for protocol translators equipped with serial network interfaces.

Testing Connectivity with the Ping Command

As an aid to diagnosing basic network connectivity, many network protocols support an echo mechanism. This involves sending a special datagram to the destination host, then waiting for a reply datagram from that host. Results from this echo protocol can help in evaluating the path-to-host reliability, delays over the path, and whether the host can be reached or is functioning.

To implement these features, use the privileged EXEC command:

ping

The **ping** command is the Cisco user interface to a number of echo protocols.

When the **ping** command is entered, the system issues a prompt for a protocol keyword. The default protocol is IP; the only alternative is **pad**.

After determining the protocol type, the **ping** command prompts for an address or host name, repeat count (default is 5), datagram size (default is 100 bytes), timeout interval (default is 2 seconds), and extended commands (default is none). The precise dialogue varies from protocol to protocol.

If a host name or address is typed on the same line as the EXEC **ping** command, the default actions will be taken as appropriate for the protocol type of that name or address.

The **ping** command uses the exclamation point (!) and period (.) in its display. Each exclamation point indicates receipt of a reply. A period (.) indicates the network server timed out while waiting for a reply. The output concludes with the success rate and minimum, average, and maximum round-trip times.

To abort a ping session, type the escape sequence (by default, Ctrl-^, X.)

A sample display and tips for using the ping protocol follows.

Sample:

For IP, the **ping** command sends ICMP *Echo Request messages* and waits for ICMP *Echo Reply messages*. The following example shows the **ping** command output for IP:

```
Protocol [ip]:
Target IP address:192.31.7.17
Repeat count [5]:50
Datagram size [100]:500
Timeout in seconds [2]:
Extended commands [n]:
Type Ctrl^X to abort.
Sending 50, 500-byte ICMP Echoes to 192.31.7.17, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent, round-trip min/avg/max = 16/22/36 ms
```

If the protocol translator receives an ICMP *Host Unreachable* message during the **ping** session, the letter "U" will be printed out rather than a "!" or ".". The letter "N" indicates receipt of an ICMP *Network Unreachable* message, and the letter "P" indicates receipt of a *Protocol Unreachable* message. The letter "Q" indicates *source quench received*, and the character "?" indicates an *unknown packet type.*

The IP **ping** command, in verbose mode, accepts a data pattern. The pattern is specified as a 16-bit hexadecimal number. The default pattern is 0xABCD. Patterns such as all ones or all zeros can be used to debug data sensitivity problems on CSU/DSUs.

Note: If the IP version of the **ping** command is used on a directly connected interface, the packet is sent out the interface and should be forwarded back to the router from the far end. The time travelled reflects this round trip route. This feature can be useful for diagnosing serial line problems. By placing the local or remote CSU/DSU into loopback mode and "pinging" your own interface, you can isolate the problem to the router or leased line.

Using the Trace Command

The **trace** command is a useful debugging command which allows the network administrator to discover the routes packets will actually take when travelling to their destination. The **trace** command supports IP route tracing. The command syntax is:

trace [*destination*]

To invoke a simple **trace** test, enter the destination address or host name on the command line. The default parameters for the appropriate protocol are assumed and the tracing action begins.

To use non-default parameters and invoke an extended **trace** test, enter the command without a destination argument. You will be stepped through a dialogue to select the desired parameters.

Typing the escape sequence (by default, Ctrl-^, X) terminates a **trace** command.

How the Trace Command Works

The **trace** command works by taking advantage of the error messages generated by routers when a datagram exceeds its time-to-live (TTL) value.

The **trace** command starts by sending probe datagrams with a TTL value of one. This causes the first router to discard the probe datagram and send back an error message. The **trace** command sends several probes at each TTL level and displays the round trip time for each.

The **trace** command sends out one probe at a time. Each outgoing packet may result in one of two error messages. A *time exceeded* error message indicates that an intermediate router has seen and discarded the probe. A *destination unreachable* error message indicates that the destination node has received the probe and discarded it because it could not deliver the packet. If the timer goes off before a response comes in, **trace** prints an asterisk (*).

The **trace** command terminates when the destination responds, when the maximum TTL was exceeded, or when the user interrupts the trace with the escape sequence.

Common Trace Command Problems

Due to bugs in the IP implementations of various hosts and routers, you may notice one or more of the following behaviors when using the **trace** command:

- Not all destinations will correctly respond to a *probe* message by sending back an *ICMP port unreachable* message. A long sequence of TTL levels with only asterisks, terminating only when the maximum TTL has been reached, may indicate this problem.
- There is a known problem with the way some hosts handle an *ICMP TTL exceeded* message. Some hosts generate an *ICMP* message but they re-use the TTL of the incoming packet. As this is zero, the ICMP packets do not make it back. When you trace the path to such a host, you may see a set of TTL values with asterisks (*). Eventually the TTL gets high enough that the *ICMP* message can get back. For example, if the host is six hops away, **trace** will time-out on responses 6 through 11. Response 12 and after should be fine.

Tracing IP Routes

When tracing IP routes, the following **trace** command parameters may be set:

- `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 protocol translator to use as a source address for the probes. The terminal 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 may suppress the symbolic display.
- `Timeout in seconds.` The number of seconds to wait for a response to a probe packet. The default is three seconds.
- `Probe count.` This is 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 may be set to a higher value to suppress the display of known hops.
- `Maximum Time to Live [30].` This is the largest TTL value which may be used. The default is 30. The **trace** command terminates when the destination is reached or when this value is reached.
- `Port Number.` This is the destination port used by the UDP probe messages. The default is 33,434.
- `Loose, Strict, Record, Timestamp, Verbose.` These are IP header options. You may specify any combination. The **trace** command issues prompts for the required fields. Note that **trace** will place the requested options in each probe; however, there is no guarantee that all routers (or end-nodes) will process the options.
- `Loose Source Routing.` You may specify a list of nodes which must be traversed when going to the destination.
- `Strict Source Routing.` You may specify a list of nodes which must be the *only* nodes traversed when going to the destination.
- `Record.` You may specify the number of hops to leave room for.
- `Timestamp.` You may specify the number of timestamps 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.

Example:

The following is an example of the simple use of the **trace** command with IP:

```
chaos#trace nic.ddn.mil
Type escape sequence to abort.
Tracing the route to NIC.DDN.MIL (26.0.0.73)
  1 DEBRIS.CISCO.COM (131.108.1.6) 1000 msec 8 msec 4 msec
  2 BARRNET-GW.CISCO.COM (131.108.16.2) 8 msec 8 msec 8 msec
  3 EXTERNAL-A-GATEWAY.STANFORD.EDU (192.42.110.225) 8 msec 4 msec 4 msec
  4 BB2.SU.BARRNET.NET (131.119.254.6) 8 msec 8 msec 8 msec
  5 SU.ARC.BARRNET.NET (131.119.3.8) 12 msec 12 msec 8 msec
  6 MOFFETT-FLD-MB.DDN.MIL (192.52.195.1) 216 msec 120 msec 132 msec
  7 NIC.DDN.MIL (26.0.0.73) 412 msec 628 msec 664 msec
```

Example:

The following is an example going through the extended dialogue of the **trace** command:

```
chaos#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 DEBRIS.CISCO.COM (131.108.1.6) 1000 msec 4 msec 4 msec
  2 BARRNET-GW.CISCO.COM (131.108.16.2) 16 msec 4 msec 4 msec
  3 EXTERNAL-A-GATEWAY.STANFORD.EDU (192.42.110.225) 16 msec 4 msec 4 msec
  4 NSS13.BARRNET.NET (131.119.254.240) 112 msec 8 msec 8 msec
  5 SALT_LAKE_CITY.UT.NSS.NSF.NET (129.140.79.13) 72 msec 64 msec 72 msec
  6 ANN_ARBOR.MI.NSS.NSF.NET (129.140.81.15) 124 msec 124 msec 140 msec
  7 PRINCETON.NJ.NSS.NSF.NET (129.140.72.17) 164 msec 164 msec 172 msec
  8 ZAPHOD-GATEWAY.JVNC.NET (128.121.54.72) 172 msec 172 msec 180 msec
  9 HOTBLACK-GATEWAY.JVNC.NET (130.94.0.78) 180 msec 192 msec 176 msec
 10 CAPITAL1-GATEWAY.JVNC.NET (130.94.1.9) 280 msec 192 msec 176 msec
 11 CHEESESTEAK2-GATEWAY.JVNC.NET (130.94.33.250) 284 msec 216 msec 200 msec
 12 CHEESESTEAK1-GATEWAY.JVNC.NET (130.94.32.1) 268 msec 180 msec 176 msec
 13 BEANTOWN2-GATEWAY.JVNC.NET (130.94.27.250) 300 msec 188 msec 188 msec
 14 NEAR-GATEWAY.JVNC.NET (130.94.27.10) 288 msec 188 msec 200 msec
 15 IHTFP.MIT.EDU (192.54.222.1) 200 msec 208 msec 196 msec
 16 E40-03GW.MIT.EDU (18.68.0.11) 196 msec 200 msec 204 msec
 17 MIT.EDU (18.72.2.1) 268 msec 500 msec 200 msec
```

In the output, "!N" represents ICMP *network unreachable*, "!H" represents ICMP *host unreachable*, "!P" represents ICMP *protocol unreachable*, and "!Q" represents *source quench received*. The asterisk "*" represents *time-out*, and "?" represents *unknown packet type.*

Using the DEC MOP Server

All Cisco internetworking products include a server which implements a subset of DEC's Maintenance Operation Protocol (MOP) for Ethernet interfaces. The MOP server supports the request ID message, periodic system ID messages, and the remote console carrier functions.

The MOP server periodically multicasts a system ID message, which is used by DEC's Ethernet configurator to determine what stations are present in an Ethernet network. The configurator is controlled by the Network Control Program (NCP) command **define module configurator**. For more information on this command, consult DECnet or VAX documentation.

The Cisco internetworking products use the MOP communication device code of 121. This code has been assigned to Cisco by DEC, although some versions of DECnet-VAX software may report the code numerically, rather than with a device name. The DEC Ethernet product also makes use of receipt of system ID messages when building network maps.

The MOP server supports the DEC remote console function. Using this capacity, a system manager on a DECnet system can create a virtual terminal connection to a Cisco protocol translator. Due to the nature of the MOP server, only a single inbound connection per Ethernet interface is supported. The MOP protocol does not contain the necessary mechanisms for supporting more than one connection at a time.

MOP is *not* a routable protocol. To bridge the MOP console carrier and system ID functions, you must enable bridging for protocol type 6002. The periodic system ID messages are sent to the multicast address AB00.0002.0000.

The EXEC command **debug mop** reports interesting events occurring within the MOP server, including reception of request ID messages, transmission of system ID messages, and reservation and release of the remote console.

Enabling MOP for an Interface

To control whether MOP is enabled for an interface, use the **mop enabled** command. The command syntax is:

mop enabled
no mop enabled

The default is enabled.

Use the **no mop enabled** to disable the MOP if you do not want to run it at all.

Controlling the MOP Sysid Messages

To control whether MOP periodic sysids messages are sent out to an interface, use the **mop sysid** command. The command syntax is:

mop sysid
no mop sysid

You can still run MOP, but not have the background sysid messages sent out. This lets you still use the MOP remote console, but does not generate messages used by the configurator.

Use the **no mop sysid** to disable the MOP from sending the sysid messages.

Loading Software Over the Network

As configured at the factory, the operating system software executes instructions in the onboard EPROM. If you have a CSC/3 CPU card, you need not change the system EPROMs with each software update. Instead, you can download the latest software over the network. This process is called *netbooting.*

Note: Refer to "Setting Configuration File Specifications" and "Automatic Configuration Using Remote Hosts," in Chapter 4 for more information concerning network booting.

Netbooting works as follows: when you power up your Cisco server product for the first time, it checks the processor configuration register or the non-volatile memory for special netbooting instructions. If the system finds no special instructions, it executes the default EPROM software.

If the system finds netbooting instructions, it determines its interface address and then runs a special process to load the new software into memory.

You can specify boot loading in two ways. The first way involves setting the low four bits of the processor configuration register; refer to Cisco's *Modular Products Hardware Installation and Reference* publication for details. If no bits are set, you must manually boot the system using the System Bootstrap program.

If only the low bit is set, the system runs the default software. The system interprets any other binary bit combination as an octal number for use in forming the boot file name. The system forms the boot filename by starting with the word *cisco* and then appending the octal number, a hyphen, and the processor type name. The System Bootstrap program displays the processor type name at system startup.

For example, if bit one in the four-bit field is set and the processor type is CSC/3, the boot file name formed is *cisco2-csc3*. Assuming no other information is available, the system would try to TFTP-load the file *cisco2-csc3* by first sending a broadcast TFTP read request to determine which server host had the file.

The second way to specify boot loading uses the non-volatile memory option, which enables you to provide more detailed instructions for software downloading. You can use the *boot* configuration command to specify both the boot file name and the IP address of the server host.

Booting Considerations

By default, the protocol translator continues sending TFTP Read Request messages until it receives a response. The protocol translator remains unusable as long as the network or the host with the specified file is unavailable. To limit the number of netbooting attempts, set bit 13 in the processor configuration register to 1. The protocol translator then gives up after five netbooting attempts and returns to the ROM operating software.

The protocol translator can use any network interface, regardless of the media type or encapsulation method, to load operating software. If the interface does not support broadcasts (for example, a protocol translator using the X.25 interface does not), you must use the non-volatile memory to specify the address of the server host with the desired file.

To display the Internet address of the server host that provided the current operating software, use the EXEC command **show version**.

Reloading the Operating System

To reload the operating system, use the **reload** EXEC command. The command syntax is:

reload

The command halts the protocol translator. If the system is set to restart on error, it reboots itself.

Writing System Configuration Information

This section describes the privileged **write** commands used to manage the system configuration information.

Erasing Configuration Information

To erase the configuration information, use the **write erase** EXEC command. The command syntax is:

write erase

This command erases the configuration information in the non-volatile memory. This command does not affect the configuration in use.

Writing Configuration Information to Memory

To save the configuration into non-volatile memory, use the **write memory** command. The command syntax is:

write memory

This command copies the current configuration information to the non-volatile memory.

Copying the Configuration to a Network

To save the configuration to the network via TFTP, use the **write network** command. The command syntax is:

write network

This command sends a copy of the current configuration information to a server host. You are prompted for a destination host and a file name.

Displaying Current Configuration Information

To write the configuration on the terminal, use the **write terminal** command. The command syntax is:

write terminal

This command displays the current configuration information on the terminal.

Sending Messages to Lines

To send messages to one or all lines, use the privileged **send** EXEC command. The syntax for the command is:

send {*line-number* | *}

To send a message to a particular line, use the argument *line-number* to specify the line. To send a message to all lines, use an asterisk (*). The system prompts for the message, which may be several text lines long. End the message by typing the Ctrl-Z key sequence. Type Ctrl-C to abort the command. This command can be used to inform users of an impending shutdown.

Example:

```
cisco-prompt#send all
Enter message, end with CTRL/Z; abort with CTRL/C:
System shutdown in 10 minutes.
<Ctrl-Z>
Send message? [confirm]yes


***
***
*** Message from tty56, cube CE10, to all terminals:
***
System shutdown in 10 minutes.
```

Testing the System

Included as part of the EXEC command set are commands that allow testing of system interfaces and memory.

Caution: Use of these commands is not recommended, as they are intended to aid Cisco manufacturing personnel in checking out system functionality.

Factory Test

To test the network interfaces, use the **test interfaces** EXEC command. The command syntax is:

test interfaces

This command is intended for the factory checkout of network interfaces. It is not intended for diagnosing problems with an operational protocol translator. The **test interfaces** output will not report correct results if the system is attached to a "live" network. For each network interface that has an IP address that can be tested in loopback (MCI Ethernet and all serial interfaces), the **test interface**s command sends a series of ICMP echoes. Error counters are examined to determine the operational status of the interface.

Asynchronous Card Tests

To test the asynchronous cards for ASM and MSM systems, use the **test lines** EXEC command. The command syntax is:

test lines

This test runs software diagnostics on asynchronous serial interface boards that can be useful when analyzing hardware failures and suspected hardware failures. The command is not designed to be used on a system while the system is in use. This test should be used only at the direction of your Cisco technical support representative.

Memory Test

To test system memory, use the **test memory** EXEC command. The command syntax is:

test memory

This command performs a test of Multibus memory, including the non-volatile memory.

Caution: This test will overwrite the contents of memory. You will need to rewrite non-volatile memory after running this command. If you test Multibus memory, you will need to reload the system to restore correct operation of the network interfaces.

System Management Interface Subcommand Summary

This section lists all system management interface subcommands in alphabetic order:

[no] mop enabled

Controls whether MOP is enabled for an interface or disabled.

The default is enabled.

Use the **no mop enabled** to disable the MOP if you do not want to run it at all.

[no] mop sysid

Controls whether MOP periodic sysids messages are sent out to an interface.

Use the **no mop sysid** to disable the MOP from sending the sysid messages.

System Management EXEC Command Summary

This section lists all of the EXEC system management and user commands in alphabetical order.

clear counters [*type unit***]**

Resets all interface counters listed in show interface statistics. The arguments *type* and *unit* specify the interface type and unit or card number (such as, Ethernet 0 or Serial 0).

clear interface *type unit*

Resets the hardware logic on an interface. The arguments *type* and *unit* specify the interface type and unit or card number (such as, Ethernet 0 or Serial 0).

clear line *line-number*

This command aborts any connections, terminates the associated processes, and resets the data structures associated with a terminal line. The argument *line-number* specifies the terminal line number.

ping

Issues a prompt for a protocol keyword. The default protocol is IP, the only alternative is **pad** for protocol translators. After determining the protocol type, the **ping** command will prompt for an address or host name, repeat count (default is 5), datagram size (default is 100 bytes), timeout interval (default is 2 seconds), and extended commands (default is none). The precise dialogue varies from protocol to protocol.

reload

Halts the protocol translator. If the system is set to restart on error, it reboots itself.

send {*line-number* **| *}**

Sends messages to one or all lines. To send a message to a particular line, use the argument *line-number* to specify the line. To send a message to all lines, use an asterisk (*). The system prompts for the message, which may be several text lines long. End the message by typing the Ctrl-Z key sequence. Type Ctrl-C to abort the command.

show ?

Lists all the **show** command options. Two lists may be displayed, one at the user-level prompt, and one at the enabled, privileged-level prompt.

show buffers *interface*

Displays statistics for the buffer pools on the protocol translator. The *interface* argument displays all the buffers for the specified interface.

show configuration

Displays the contents of the non-volatile memory, if present and valid.

show controllers {serial|mci}

Displays current internal status information for different interface cards. Use the keywords **mci** or **serial** to display the information about those cards.

show interfaces

Displays statistics for the network interfaces on the network server.

show line [*line-number***]**

Displays a summary status of terminal lines on the protocol translator. Include the optional argument *line-number* to display detailed information about a particular line.

show logging

Displays the state of syslog error and event logging, including host addresses and whether console logging is enabled. This command also displays SNMP (Simple Network Monitoring Protocol) configuration parameters and protocol activity.

show memory

Displays memory free pool statistics. These statistics include summary information about the activities of the system memory allocator, and a block-by-block listing of memory use.

show sessions

Displays information about your active terminal sessions. The information it displays includes the host name, address, number of characters waiting to be sent to the terminal, idle time, and connection name. An asterisk (*) indicates the users's current session.

show processes

Displays information about all active processes, including:

- Process ID
- Queue type (high, medium, low)
- Scheduler test (event, time, suspended)
- Total runtime (in milliseconds)
- Count of invocations
- Microseconds per invocation
- Stack utilization
- Controlling terminal
- Process name

show stacks

Monitors the stack utilization of processes and interrupt routines. Its display includes the reason for the last system reboot. If the system was reloaded because of a system failure, a saved system stack trace is displayed. This information can be useful to support personnel for analyzing crashes in the field.

show terminal

Displays the configuration parameter settings for the current terminal. This command may be issued at the user-level prompt.

show users [all]
systat [all]

Display information about the active lines of the network server, including the line number, connection names, and terminal location. Specify the keyword **all** to display information for both active and inactive lines. These commands enable monitoring of virtual terminal use. These commands may be issued at the user-level prompt.

show version

Displays the configuration of the system hardware, the software version strings, the names and sources of configuration files and/or boot images, and the Internet addresses of the interfaces.

terminal dispatch-character *decimal-number1* **[***decimal-number2 … decimal-numberx***]**

Sets up the packet dispatch character. Defines a character or string that causes a packet to be sent. The argument decimal-number is the ASCII decimal representation of the character or string.

terminal [no] download

Sets the line to maximum transparency file transfers. The **terminal download** command is used when running a program such as KERMIT, XMODEM, or CrossTalk that downloads a file across a protocol translator line. It sets up the terminal line as a transparent pipe that can be used to transmit data and is equivalent to entering all the following commands:

The **terminal no download** command restores the line's original parameter settings.

terminal escape-character *decimal-number*
terminal no escape-character

Sets the escape character for the current terminal line. The argument *decimal-number* is the ASCII decimal representation of the desired escape character or an escape character (Ctrl-P, for example).

terminal flowcontrol { none | software [in | out] | hardware}

Sets the method of data flow control for the current terminal line.

The keyword **software** sets software flow control. An additional keyword specifies the direction: **in** causes the terminal server to listen to flow control from the attached device, and **out** causes the terminal server to send flow control information to the attached device. If you do not specify a direction, both directions are assumed.

The keyword **hardware** sets hardware flow control. For information about setting up the RS-232 line, see your respective *Hardware Installation and Reference* publication.

By default, no flow control method is set for a line.

terminal length *screen-length*

Sets the number of lines on the screen of the current terminal. The argument *screen-length* is the desired number of lines. The default length is 24 lines. A value of 0 (zero) disables pausing between screens of output. The screen length specified can be learned by hosts.

terminal monitor

Enables logging of system debugging and event messages on the current terminal. The **terminal monitor** command copies the system debugging and event messages to the current terminal. To use this command, you must first issue the **enable** command and enter the password to access the privileged command mode.

terminal [no] notify

Establishes input notification. The **terminal notify** command causes the protocol translator to notify you of pending output. The **terminal no notify** command ends such notifications.

terminal padding *decimal-number count*
terminal no padding *decimal-number*

Sets the character padding on the current terminal line. The argument *decimal-number* is the ASCII decimal representation of the character. It can be any of the 127 ASCII characters, up to 255 padding characters in length.

The argument *count* is the number of NUL bytes sent after that character.

The **terminal no padding** command ends this padding after the character represented by *decimal-number.*

terminal telnet-transparent

Causes the current terminal line to send a Return (CR) as a CR followed by a NUL instead of a CR followed by a Line Feed (LF). This scheme permits interoperability with different interpretations of end-of-line handling in the Telnet protocol specification.

terminal terminal-type *terminal-name*

Records the current terminal type. The type in the argument *terminal-name* is passed as information to the remote host. The type specified in *terminal-name* is used by TN3270 for display management.

terminal transport [telnet|pad|rlogin|none]

Selects a preferred remote terminal protocol. This EXEC command sets the preferred protocol for the duration of the current session.

terminal width *columns*

Sets the number of characters (columns) on a single line of the current terminal screen. The rlogin protocol uses the argument *columns* to set up terminal parameters on a remote UNIX host.

test interfaces

Sends a series of ICMP echoes. Error counters are examined to determine the operational status of the interface.

test lines

Tests the asynchronous cards for ASM and MSM systems.

test memory

Performs a test of Multibus memory, including the non-volatile memory.

trace [*destination***]**

Allows the network administrator to discover the routes packets will actually take when travelling to their destination. The command supports IP route tracing.

To invoke a simple **trace** test, enter the destination address or host name on the command line. The default parameters for the appropriate protocol are assumed and the tracing action begins.

To use non-default parameters and invoke an extended **trace** test, enter the command without a destination argument. You will be stepped through a dialogue to select the desired parameters.

Typing the escape sequence (by default, Ctrl-^, X) terminates a **trace** command.

write erase

Erases the configuration information in the non-volatile memory. This command does not affect the configuration in use.

write memory

Copies the current configuration information to the non-volatile memory.

write network

Sends a copy of the current configuration information to a server host. You are prompted for a destination host and a file name.

write terminal

Displays the current configuration information.