

Chapter 5

System Management

5

This chapter describes how to manage the protocol translator system. You will find information about these system processes and tasks in this chapter:

- Asynchronous serial line use
- System buffer pool allocation
- System configuration and processes
- Flash ROM memory
- System error message log
- System memory allocation and stack utilization
- Monitoring, debugging, and tracing system processes
- Testing system controllers

See the chapters containing information about the interfaces and protocols supported by Cisco software for descriptions of the interface- and protocol-specific debugging and monitoring commands.

Most of the system 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. Refer to the section “The EXEC Command Interpreter” in the chapter “Startup and Basic Configuration” for a description of system command levels and how to access them.

A command summary is included at the end of the chapter.

Maintaining the Asynchronous Lines

Use the privileged EXEC commands in this section to monitor and maintain the asynchronous serial lines.

Displaying Line Status

To obtain the status of lines configured on the protocol translator, use the **show line** EXEC command. The command syntax is:

```
show line [line-number]
```

The **show line** command entered with no argument displays a summary status of the terminal lines on the protocol translator.

The following is sample command output of the **show line** command:

Tty	Typ	Tx/Rx	A	Modem	Roty	AccO	AccI	Uses	Noise
1	TTY	9600/9600	-	-	-	-	-	5	0
2	TTY	9600/9600	-	-	-	-	-	13	1
3	TTY	9600/9600	-	-	-	-	-	0	0
4	TTY	9600/9600	-	-	-	-	-	2	0
5	TTY	9600/9600	-	-	-	-	-	0	0
6	TTY	9600/9600	-	-	-	-	-	0	0
7	TTY	9600/9600	-	-	-	-	-	0	0
10	TTY	9600/9600	-	-	-	-	-	0	0
11	TTY	9600/9600	-	-	-	-	-	0	0
12	TTY	2400/2400	F	callin	-	-	-	26	5
*13	TTY	2400/2400	F	callin	-	-	-	14	10
14	TTY	9600/9600	-	-	-	-	-	0	0
15	TTY	38400/38400	-	-	-	-	-	0	0
16	TTY	4800/4800	-	-	-	-	-	0	0
17	TTY	9600/9600	-	-	-	-	-	0	0
20	TTY	9600/9600	-	-	-	-	-	0	0
*21	VTY		-	-	-	-	-	2	0
22	VTY		-	-	-	-	-	0	0
23	VTY		-	-	-	-	-	0	0

In the output, the `Tty` column lists the line number in octal 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
- TTY is a standard asynchronous (RS-232) 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; F specifies full range, and a hyphen (-) specifies no autobaud detection. 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 **show terminal** command displays the same information for the current line.

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
```

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

This section describes the EXEC **show** commands you use to monitor the protocol translator.

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 ?

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
processes        Active system processes
rif              RIF cache
sessions         Telnet and rlogin connections
stacks           Process and interrupt stack use
tcp <line>        TCP information, may specify a line
terminal         Terminal parameters
```

Displaying Buffer Pool Statistics

The protocol translator has one pool of queuing elements and five pools of packet buffers of different sizes. For each pool, the protocol translator keeps count 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. To display buffer pools, use the **show buffers** EXEC command. Enter this command at the EXEC prompt:

```
show buffers [interface]
```

The optional argument *interface* causes 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 nonzero.

Following is sample output without the optional *interface* argument. Table 1-1 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 1-1 Show Buffers Field Descriptions

Field	Description
Buffer elements	Blocks of memory used in internal operating system queues.
Small buffers	Blocks of memory used to hold network packets. Totals indicate the number of buffers of a given size that currently exist. Buffers are dynamically allocated and the numbers increase and decrease depending upon system demand.
Middle buffers	
Big buffers	
Large buffers	
Huge buffers	
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 in response to misses.
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 that the system will create.
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 Memory Utilization

Use the **show process memory** command to monitor the memory utilization of processes. Enter this command at the EXEC prompt:

show process memory

The following is a display of sample output. Table 1-2 describes the fields.

```

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
   13   0   2192         0   2192 TTY Background
   14   0         0         0         0 TCP Driver
   15   0         0         0         0 LAPB Timer
   16   0         0         0         0 X25 Timer
   17   0         0         0         0 HyBridge Input
   22   0    108      108         0 Exec
   19   0    104         0    104 IP SNMP
   20   0         0         0         0 Spanning Tree
                                     530936 Total

```

Table 1-2 Show Process Memory Field Descriptions

Field	Description
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.
Total	Total amount of memory held.

Displaying the System Configuration

To display the system configuration, use the **show configuration** EXEC command. The command syntax is:

show configuration

This privileged command displays the contents of nonvolatile memory, if present and valid. Nonvolatile memory stores the configuration information in text form as configuration commands.

Displaying the Event Logging Status

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 the chapter “System Configuration” for an explanation of how to configure message logging.

Following is a sample output. Table 1-3 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 1-3 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, 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 1-4 describes the fields and Table 1-5 lists the characteristics of each block of memory in the system.

Table 1-4 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 1-5 Characteristics of Each Block of Memory

Field	Description
Address	Hexadecimal address of block
Bytes	Size of block in bytes
Prev	Address of the previous block (should match the Address on the previous line)
Next	Address of the next block (should match the address on the next line)
Free?	Tells if the block is free
Alloc PC	Address of the system call that allocated the block
What	Name of the 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 1-6 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 1-6 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 the process.

Displaying Stack Utilization

To display stack utilizations, use the **show stacks** 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 System Version

The **show version** or the **show hardware** command displays the configuration of the system hardware, the software version, the names and sources of configuration files, and the boot images. Enter one of these commands at the EXEC prompt:

show version **show hardware**

The following shows sample output:

```
X.25/TCP/LAT Protocol Translator (PT3-LX), Version 9.0(1)
Copyright (c) 1986-1992 by cisco Systems, Inc.
Compiled Fri 27-Mar-92 08:56 by dchan

System Bootstrap, Version 4.5(0.5)

mintaka uptime is 22 minutes
System restarted by power-on
Running default software

CSC3 (68020) processor with 4096K bytes of memory.
X.25 software.
SuperLAT software (copyright 1990 by Meridian Technology Corp).
```

```
1 MCI controller (1 Ethernet, 1 Serial).
1 Ethernet/IEEE 802.3 interface.
1 Serial network interface.
32K bytes of non-volatile configuration memory.
Configuration register is 0x101
```

In the output, the first line is the bootstrap version string. The second through fourth lines list information about the system software; the version number is on the second line. The fifth line shows the system name and *uptime*, or the amount of time the system has been up and running. The sixth line provides a log of how the system was last booted, both as a result of normal system startup and of system error. For example, this line may be displayed to indicate a bus error that is generally the result of an attempt to access a nonexistent address:

```
System restarted by bus error at PC0XC4CA address 0X210C0C0
```

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 non-standard software options. The configuration register contents are displayed in hexadecimal notation.

Displaying Flash ROM Statistics

The Flash Memory Card allows system software images to be stored, booted, and rewritten as necessary. This card, known as the CSC-MC+, is supported on the A, M, and C chassis platforms. The CSC-MC+ provides a more fault-tolerant solution to users who net boot exclusively. The CSC-MC+ reduces the effects of network failure on system netbooting. The CSC-MC+ card replaces the nonvolatile memory card (CSC-MC) used in the M and C platform systems, and connects to the Multiport Communications Interface (MCI-3) card by a 50-pin flat cable.

The following list briefly describes the features of the CSC-MC+ card:

- Allows copying the TFTP image to Flash memories
- Allows copying the Flash memory image to a TFTP server
- Can be remotely loaded with multiple system software images through a TFTP transfer.
- Allows a communications server to be booted manually, or automatically from a system software image stored in Flash memory. Booting directly from ROM, and netbooting from a TFTP file server, are still available options.
- Provides write-protection against accidental erasure or reprogramming of the Flash memories.
- Can store up to 64 kilobytes of configuration memory information in nonvolatile memory.
- Provides four megabytes of Flash memory storage.

- The nonvolatile memory chips have built-in lithium batteries for nonvolatile memory backup in case of power failure. These batteries are designed to last ten years.

Refer to the *Hardware Installation and Reference* publication for your particular communications server for the hardware requirements and step-by-step installation instructions.

Note: Booting from ROM is faster than booting from Flash, however, if you are netbooting, Flash is faster and more reliable than booting over your network.

Use the **show flash** EXEC command to display the total amount of Flash memory present on the Flash card, the type of card connected to the Flash card, any files that may currently exist in Flash memory and their size, and the amounts of Flash memory used and remaining.

Use the **show flash all** EXEC command to display all information displayed by **show flash** and also all the information about each Flash memory device.

Once you configure Flash, the **show flash** or **show flash all** commands will display the names of the system software images. The command syntax follows:

show flash
show flash all

Following is sample output of the **show flash** command:

```
PT#show flash
4096K bytes of flash memory on MC+ card (via MCI).
  Contains:
    pt3-rx.90-0.17 (1648952)
    STS10X.45-0.4 (30444)
  [835192/4194304 bytes free]
```

Following is sample output of the **show flash all** command:

```
PT#show flash all
4096K bytes of flash memory on MC+ card (via MCI).
  ROM 0, U2 , code 0x89BD, size 0x40000, name INTEL 28F020
  ROM 1, U19, code 0x89BD, size 0x40000, name INTEL 28F020
  ROM 2, U3 , code 0x89BD, size 0x40000, name INTEL 28F020
  ROM 3, U20, code 0x89BD, size 0x40000, name INTEL 28F020
  ROM 4, U4 , code 0x89BD, size 0x40000, name INTEL 28F020
  ROM 5, U21, code 0x89BD, size 0x40000, name INTEL 28F020
  ROM 6, U5 , code 0x89BD, size 0x40000, name INTEL 28F020
  ROM 7, U22, code 0x89BD, size 0x40000, name INTEL 28F020
  ROM 8, U9 , code 0x89BD, size 0x40000, name INTEL 28F020
  ROM 9, U26, code 0x89BD, size 0x40000, name INTEL 28F020
  ROM 10, U10, code 0x89BD, size 0x40000, name INTEL 28F020
  ROM 11, U27, code 0x89BD, size 0x40000, name INTEL 28F020
  ROM 12, U11, code 0x89BD, size 0x40000, name INTEL 28F020
  ROM 13, U28, code 0x89BD, size 0x40000, name INTEL 28F020
  ROM 14, U12, code 0x89BD, size 0x40000, name INTEL 28F020
  ROM 15, U29, code 0x89BD, size 0x40000, name INTEL 28F020
  Contains:
    pt3-rx.90-0.17 (1648952) [invalidated]
    pt3-rx.90-0.17 (1648952)
```

```
STS10X.45-0.4 (30444) [invalidated]
STS10X.45-0.4 (30444)
[835192/4194304 bytes free]
```

Note the `[invalidated]` flag at the end of the second example. This flag will appear when a file is rewritten (recopied) into Flash memory. A prompt will tell you that the identical file already exists and that it will be invalidated. The first (now invalidated) copy of the file is still present within Flash memory, but it is rendered unusable in favor of the newest version.

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

Both examples illustrate that the Flash memory can store and display multiple, independent software images. This feature would be most useful for storing default system software as a back-up.

In the second example, `ROM 0` is at location `U2` on the Flash Memory Card. The `code` is a vendor code; `size` is in hex bytes; `INTEL` is the vendor name; and the last number is the chip part number.

Troubleshooting Network Operations

Cisco Systems protocol translators include hardware and software to aid in tracking down problems with the protocol translator or with other hosts on the network. The privileged debug EXEC commands start the console display of several classes of network events. The EXEC command **show debugging** displays the state of each debugging option.

Each **debug** command option has a corresponding **undebug** command that turns message logging off.

For a list and brief description of all the debug command options, enter the **debug ?** command at the privileged-level EXEC prompt. This section provides an overview of how to use the debugging commands and a table of the most commonly used system debugging commands. See the interface- and protocol-specific chapters for other **debug** command descriptions.

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 output to the line.

Note: Debugging output is given high priority by the system. For this reason, debugging commands should be turned on only for troubleshooting specific problems, or during troubleshooting sessions with Cisco engineers. Excessive debugging output can render the system inoperable.

Table 1-7 Commonly Used System Debug Commands

Command	Description
debug ?	Displays a list of the debug command options.
debug all	Enables logging of all possible debugging options.
debug broadcast	Enables logging of all broadcast traffic.
debug modem	Enables logging of modem control events such as signal transitions and autobaud progress.
debug packet	Enables logging of packets received with unknown packet types.

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 and it is the only one supported for the protocol translator.

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 that the network server timed out while waiting for a reply. Other characters may appear in the **ping** output display, depending on the protocol type. 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 Session:

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: 131.108.19.5
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 131.108.19.5, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent, round-trip min/avg/max = 16/22/36 ms
```

Table 1-8 Ping Response Characters

Char	Description
!	Reply received.
.	Server timed out waiting for reply.
U	Server received a <i>Host Unreachable</i> message.
N	Server received a <i>Network Unreachable</i> message.
P	Server received a <i>Protocol Unreachable</i> message.
Q	Server received a <i>Source Quench Received</i> message.
?	Server received an <i>Unknown Packet Type message</i> .

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.

Checking Routes with the Trace Command

The **trace** command is a useful debugging command that 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 nondefault 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 protocol translator 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.
 - `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.

Sample Session 1:

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)
 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.DDN.MIL (192.52.195.1) 216 msec 120 msec 132 msec
 6 NIC.DDN.MIL (26.0.0.73) 412 msec 628 msec 664 msec
```

Sample Session 2:

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

Table 1-9 Trace Response Characters

Char	Description
!N	Server received a <i>Network Unreachable</i> message.
!H	Server received a <i>Host Unreachable</i> message.
!P	Server received a <i>Port Unreachable</i> message.
!Q	Server received a <i>Source Quench</i> message.
?	Server received an <i>Unknown Packet Type</i> message.
*	Server timed out.

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 nonvolatile memory. This command does not affect the configuration in use.

Writing Configuration Information to Memory

To save the current configuration and copy it into nonvolatile memory, use the **write memory** command. The command syntax is:

write memory

This command copies the current configuration information to nonvolatile 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.

Testing the System

Included as part of the EXEC command set are commands that allow testing of the system interface, system memory, asynchronous cards, and 16 Mbps Token Ring interfaces.



Caution: Using these commands is not recommended, as they are intended to aid Cisco manufacturing personnel in checking 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 interfaces** 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 nonvolatile 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.

16 Mbps Token Ring Card Test

To test the CSC-R16 16 Mbps Token Ring cards, use the **test sbe** EXEC command. The command syntax is:

test sbe

This test runs software diagnostics on 16 Mbps Token Ring 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.

System Management EXEC Command Summary

This section provides an alphabetical list of the EXEC system management commands described in this chapter:

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.

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 broadcast

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

debug modem

The **debug modem** command enables logging of modem control events, such as signal transitions and autobaud progress.

debug packet

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

ping

Issues a prompt for a protocol keyword. The default protocol is IP and it is the only one needed for protocol translators.

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 protocol translator has one pool of queuing “elements” and four pools of packet buffers of different sizes. The *interface* argument displays all the buffers for the specified interface.

show configuration

Displays the contents of nonvolatile memory, if present and valid. Nonvolatile memory stores the configuration information in the protocol translator in text form as configuration commands.

show debugging

Displays the current settings of the **debug** and **undebug** command options.

show flash
show flash all

The **show flash** command displays the total amount of Flash memory present on the Flash card, the type of card connected to the Flash card, any files that may currently exist in Flash memory and their size, and the amounts of Flash memory used and remaining.

The **show flash all** command displays all information displayed by **show flash** and also all the information about each Flash memory device.

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 process memory

Displays memory utilization.

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.

test interfaces

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

test lines

Runs software diagnostics on asynchronous serial interface boards.

test memory

Performs a test of Multibus memory, including nonvolatile memory.

test sbe

Runs software diagnostics on 16 Mbps Token Ring interface boards.

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 nondefault 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 nonvolatile memory. This command does not affect the configuration in use.

write memory

Copies the current configuration information to nonvolatile 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.