WELLFLEET SYSTEMS OPERATOR'S GUIDE

Wellfleet Systems Operator's Guide

Please address questions and comments to:

Technical Publications
Wellfleet Communications, Inc.
12 DeAngelo Drive
Bedford, MA 01730-2204
Tel: (617) 275-2400
Fax: (617) 275-5001

Information presented in this document is subject to change without notice.

Ethernet is a trademark of Xerox Corporation.

DECnet is a trademark of Digital Equipment Corporation.

Printed in U.S.A.

July 1989

Table of Contents

List of Figures

Preface

Purpose of this Guide

The material contained in this guide provides the information needed to monitor and control a Wellfleet system. Monitoring is enabled by a series of statistical screens that present a dynamic representation of on-going system activities, and by an event log. Control is enabled by the Network Command Language Interpreter, a dual-function language that manages specific network entities and provides access to the system management information base.

Audience

This guide is intended for experienced network operators and administrators who understand communications bridging and routing. Users should be acquainted with the internet protocol suite (TCP/IP) and DECnet architecture and routing.

Organization

The *Wellfleet Systems Operator's Guide* contains three sections.

Section 1, ***Statistics Screens***, tells you how to access Statistics Screens from the Main Menu and how to interpret the data presented by each available screen.

Section 2, ***Network Command Langauge Interpreter***, describes the functions of the Network Command Language Interpreter. It provides a description of each NCL command.

Section 3, ***Event Log***, describes the format of the system event log and tells you how to interpret event log entries.

Conventions Used in This Guide

Two different type faces distinguish system-generated and user-generated data, as follows:

PRESS 'r' for reset	This type face identifies system output that appears on the console screen.
`dis echo`	This type face identifies user responses entered from the console keyboard.

Two sets of delimiters distinguish required and optional arguments in command syntaxes.

< and >	The left (<) and right (>) angle brackets delimit required arguments. The brackets are not entered as part of the command.
{ and }	The left ({) and right (}) braces delimit optional arguments. As with angle brackets, braces are not entered as part of the command.

For example:

type <filename>↵

where:

type
Is the TYPE command

filename
Is the required name of the ASCII file to be displayed

Or:

ping <remote_host> {count} {timeout}

where:

ping
Is the PING command

remote_host
Is the required remote host address in dotted decimal notation

count
Is the optional number of times to repeat the PING command

timeout
Is the optional timeout (in seconds) for each ping

The ↵ character designates the carriage return required for command completion.

For purposes of clarity, all command syntax examples appear in bold face.

Associated Documents

The *Wellfleet Systems Operator's Guide* references the following documents:

- *Wellfleet Systems Configuration Guide*

  This guide explains how to create the *config* file, the required system database that describes your network topology to the Wellfleet system.

- *Wellfleet Systems Installation Guide*

  This guide explains how to install the concentrator, link, and feeder nodes.

- *Wellfleet Systems NetManager User's Guide*

  This guide explains how to install SNMP application software in a network monitor.

- *Wellfleet Systems Overview Guide*

  This guide provides an introduction to Wellfleet's implementation of bridging and routing technology, and describes Wellfleet hardware and application software.

1. Statistics Screens

This section of the *Wellfleet Systems Operator's Guide* describes the statistics recorded by the system during network operation. It tells you how to access the Statistics Screen Menu, how to display specified statistics screens on the console screen, and how to interpret statistical displays.

1.1 Accessing the Statistics Screen Menu

You begin displaying statistical data from the Main Menu. Use the UPARROW (⇑) or DOWNARROW (⇓) key to position the cursor at **Statistics Screen Menu**, then press RETURN--or, you may simply press the <1> key. After you press RETURN or type <1>, the system displays the Statistics Screen Menu on the console screen.

Figure 1-1 shows a sample Statistics Screen Menu listing all available screens. The actual menu displayed on your console reflects your system's line configuration and the resident application software modules (Learning Bridge, IP Router, and DECnet Router). Two screens, **Circuits Statistics** and **Buffers Usage Statistics**, are always available for display. Availability of the remaining screens (**T1 Line Statistics**, **DoD IP Router Statistics**, **Learning Bridge Statistics**, and **DECnet Router Statistics**) is configuration-dependent.

- The **Circuits Statistics** Screen provides summary data for each individual circuit. It shows the number of bytes and frames received and transmitted, and the number of received and transmitted frames that contained errors.

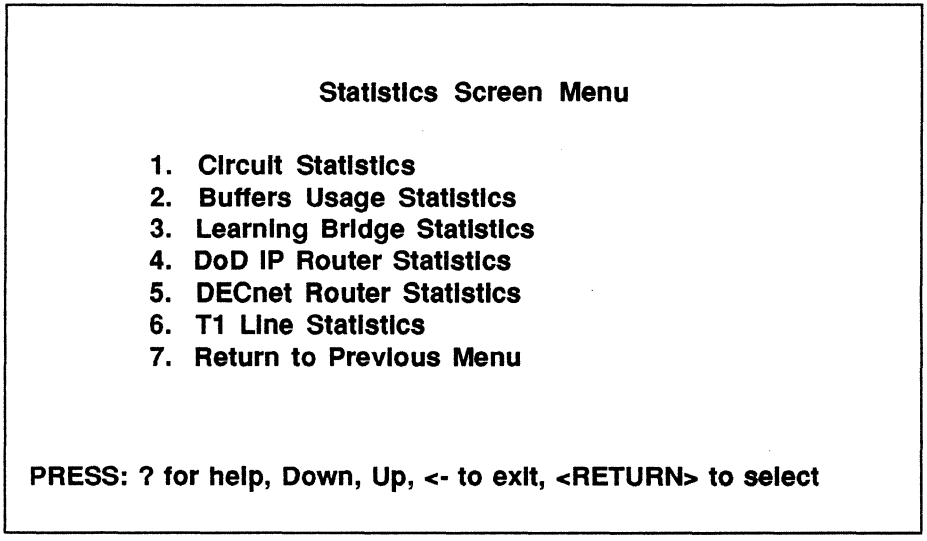

Figure 1-1: Statistics Screen Menu

- The **Buffers Usage Statistics** Screen provides information on buffer allocation and use.
- The **Learning Bridge Statistics** Screen provides summary data for each Learning Bridge circuit group. It shows the number of frames that were received, forwarded, flooded, and dropped.
- The **DoD IP Router Statistics** Screen provides summary data for each IP Router network interface. It shows the number of IP datagrams received, forwarded, handled within the router, and dropped.
- The **DECnet Router Statistics** Screen provides summary data fro each DECnet Router circuit group. It shows the number of frames received, forwarded, or dropped.
- The **T1 Line Statistics** Screen provides summary data for each T1 line. It shows the number of alarms received and generated.

All statistics screens display cumulative information gathered since the system last booted. If you wish, you can reset (set to zero) values from individual statistics screens (Section 1.3.2), or you can use the NCL RESET command (Section 2.17) to reset values.

1.2 Getting Help

You can obtain a summary description of the contents of any statistics screen from the Statistics Screen Menu. To obtain such a description, use the UPARROW (⇑) or DOWNARROW (⇓) to position the cursor to the immediate left of the desired screen, and type <?>. The system then displays a brief summary of the screen's contents. After examining this summary, you press any key to return to the Statistics Screen Menu.

1.3 Displaying a Statistics Screen

You display any statistics screen from the Statistics Screen Menu. To display a screen, press the DOWNARROW (⇓) or UPARROW (⇑) to position the cursor to the immediate left of the desired screen and then press RETURN -- or, you may simply type the menu item number that appears to the left of the screen name. After you press RETURN or type a number, the system displays the specified screen.

1.3.1 Refreshing the Statistics Screens

All statistics screens are dynamic and are updated periodically. The update cycle is configurable and may be as short as one second or as long as one minute; its default period is three seconds. The cycle duration is determined by the **Screen Refresh Rate** parameter which is set during system configuration.

1.3.2 Resetting Statistics Screen Values

Occasionally when examining a screen you may see a number prefixed with an asterisk (e.g. ***234345677**). The asterisk indicates that the number is too large to be displayed, and that the system has truncated its most significant digits. At this point you may wish to reset the value.

When you reset a displayed parameter value, you set that value and all other values displayed on the same horizontal line to zero. Parameter values displayed on the same horizontal line refer to the same circuit, circuit group, network interface, line, or slot. You also reset the value of other, non-displayed parameters. Refer to the NCL LIST command (Section 2.11) to identify non-displayed parameters set to zero when you reset a statistics display.

To reset a displayed value, use the `DOWNARROW` (⇓) or `UPARROW` (⇑) to position the cursor on the line containing the value to be reset. Then type `<r>`, and press `RETURN`.

1.3.3 Leaving a Statistics Screen

To leave a statistics screen after examining its contents, press the `LEFTARROW` (⇐) key. The system returns you to the Statistics Screen Menu.

1.4 Circuit Statistics Screen

The Circuit Statistics Screen (shown in Figure 1-2) is not configuration-dependent. Consequently, it is always available for inspection. This screen provides summary traffic volume data for each circuit within your Wellfleet system. If you wish to examine circuit usage at a greater level of detail, you can use the NCL GET command (Section 2.8) to obtain a complete listing of circuit statistics maintained by the system.

- **NAME** lists each individual circuit.
- **RX: Bytes** contains the number of bytes of data received by the circuit.
- **RX: Frames** contains the number of frames received by the circuit. Depending on the circuit type, frames can be IP datagrams, Ethernet packets, X.25 packets, or HDLC frames.
- **RX: Err** contains the number of faulty frames received by the circuit.
- **TX: Bytes** contains the number of bytes of data transmitted by the circuit.
- **TX: Frames** contains the number of frames transmitted by the circuit. Depending on the circuit type, frames can be IP datagrams, Ethernet packets, X.25 packets, or HDLC frames.
- **TX: Err** contains the number of faulty frames sent by the circuit.
- **TOTAL** provides an aggregate system-wide count for each reporting metric.

```
                        Circuit Statistics

      NAME      RX:  Bytes  Frames  Err  TX:  Bytes  Frames  Err
 1. <XXXXXXX>          #       #    #          #       #     #
 2. <XXXXXXX>          #       #    #          #       #     #
 3. <XXXXXXX>          #       #    #          #       #     #
 4. <XXXXXXX>          #       #    #          #       #     #
 5. <XXXXXXX>          #       #    #          #       #     #

      TOTAL            #       #    #          #       #     #

PRESS: 'r' for reset, Down, Up, <- to exit
```

Figure 1-2: Circuit Statistics Screen

```
                      Buffer Usage Statistics

      NAME  MSG:  miss  init  free  min   PKT:  miss  init  free  min
 1. slot #          #     #     #     #         #     #     #     #
 2. slot #          #     #     #     #         #     #     #     #

      TOTAL         #     #     #     #         #     #     #     #


PRESS: 'r' for reset, Down, Up, <- to exit
```

Figure 1-3: Buffer Usage Statistics Screen

1.5 Buffer Usage Statistics Screen

The Buffer Usage Statistics Screen (shown in Figure 1-3) is not configuration-dependent. Consequently, it is always available for inspection. This screen provides data on the allocation, usage, and availability of global memory buffers on each of the Advanced Communications Engine (ACE) processor boards within your Wellfleet system. Such data is useful should you need to troubleshoot your system.

Global memory contains two buffer types. Message buffers facilitate internal, inter-processor communications that take place over the system's VME bus. Packet buffers facilitate external network communications by temporarily storing incoming or outgoing data packets.

- **NAME** lists each individual slot within the Wellfleet cabinet that contains an ACE processor board.
- **MSG: miss** contains the number of times that the system was unable to obtain a message buffer (i.e. all buffers were in use).
- **MSG: init** contains the number of message buffers allocated when the system booted.
- **MSG: free** contains the number of message buffers available for VME data transfers. Because system operations require some overhead, the number of buffers available is somewhat less than the number of allocated buffers.
- **MSG: min** contains the lowest number of message buffers that were available since the system booted. Note that this count is directly related to the **MSG: miss** count. If message buffers were always available (**MSG: min** $\neq 0$), it follows that **MSG: miss** $= 0$.
- **PKT: miss** contains the number of times that the system was unable to obtain a packet buffer (i.e. all buffers were in use).
- **PKT: init** contains the number of packet buffers allocated when the system booted.

- **PKT: free** contains the number of packet buffers available for external data transfers. Because system operations require some overhead, the number of packet buffers available is somewhat less than the number of allocated buffers.
- **PKT: min** contains the lowest number of packet buffers that were available since the system booted. Note that this count is directly related to the **PKT: miss** count. If packet buffers were always available (**PKT: min** $\neq 0$), it follows that **PKT: miss** $= 0$.
- **TOTAL** provides an aggregate system-wide count for each reporting metric.

1.6 Learning Bridge Statistics Screen

Availability of the Learning Bridge Statistics Screen (shown in Figure 1-4) is configuration-dependent. The system provides this screen only if you have installed the Learning Bridge application software module during the system configuration process. This screen provides a circuit group-by-circuit group analysis of Learning Bridge operations.

- **NAME** lists each individual circuit group.
- **Receive** contains the number of frames received by the circuit group. Depending on the circuit type, frames can be Ethernet packets, IP datagrams, or HDLC frames.
- **Forward** contains the number of received frames that were forwarded by the Learning Bridge. Forwarding requires that the Bridge "learned" the destination address.
- **Flood** contains the number of received frames that were flooded by the Learning Bridge. Flooding indicates (1) that the Bridge had not "learned" the destination address at the time of packet reception, or (2) the packet contained a multicast address.
- **Drop** contains the number of received frames that were dropped by the Learning Bridge. Reasons for dropping packets include (but are not limited to):

    a. packet is local to the circuit
    b. flooding is disabled
    c. protocol/source address filtering

- **TOTAL** provides an aggregate system-wide count for each reporting metric.

```
                    Learning Bridge Statistics

    NAME        FRAMES:  Receive   Forward   Flood   Drop
1.  <XXXXXXX>            #         #         #       #
2.  <XXXXXXX>            #         #         #       #
3.  <XXXXXXX>            #         #         #       #
4.  <XXXXXXX>            #         #         #       #

    TOTAL                #         #         #       #

PRESS: 'r' for reset, Down, Up, <- to exit
```

Figure 1-4: Learning Bridge Statistics Screen

```
                    DoD IP Router Statistics

    NAME          PACKETS:  Receive  Transmit  Deliver  Dropped
1.  <nnn.nn.n.nnn>          #        #         #        #
2.  <nnn.nn.n.nnn>          #        #         #        #
3.  <nnn.nn.n.nnn>          #        #         #        #
4.  <nnn.nn.n.nnn>          #        #         #        #

    TOTAL                   #        #         #        #

PRESS: 'r' for reset, Down, Up, <- to exit
```

Figure 1-5: DoD IP Router Statistics Screen

1.7 DoD IP Router Statistics Screen

Availability of the DoD IP Router Statistics Screen (shown in Figure 1-5) is configuration-dependent. The system provides this screen only if you have installed the IP Router application software module during the system configuration process. This screen provides summary traffic volume data for each IP network interface.

- **Name** contains the network interface address in dotted decimal notation.
- **Receive** contains the number of IP datagrams received by the network interface.
- **Transmit** contains the number of IP datagrams transmitted by the network interface.
- **Deliver** contains the number of IP datagrams addressed to the IP Router and delivered, by the router, to one of three upper-layer protocols for processing. The three protocols are ICMP (Internet Control Message Protocol), TCP (Transmission Control Protocol), and UDP (User Datagram Protocol). If you desire more detailed information, you can use the NCL GET command (Section 2.8) to obtain counts of received and transmitted ICMP datagrams by message type.
- **Dropped** contains the number of IP datagrams dropped by the network interface. Dropped datagrams include (but are not limited to) datagrams with faulty checksums and datagrams requiring absent protocols (e.g. telnet requests in the absence of TCP). The interface also drops datagrams as directed by source address and destination address filters established during the configuration process. If you desire more detailed information, you can use the NCL GET command (Section 2.8) to obtain a further breakdown of dropped messages.
- **TOTAL** provides an aggregate system-wide count for each reporting metric.

1.8 DECnet Router Statistics Screen

Availability of the DECnet Router Statistics Screen (shown in Figure 1-6) is configuration-dependent. The system provides this screen only if you have installed the DECnet Router application software module during the system configuration process. This screen provides a circuit group-by-circuit group analysis of DECnet Router operations. If you wish to examine DECnet Router statistics at a greater level of detail, you can use the NCL GET command (Section 2.8) to obtain a complete listing of DECnet statistics maintained by the system.

- **Name** contains the name of the DECnet circuit group.
- **Receive** contains the number of data frames received by the circuit group.
- **Forward** contains the number of data frames transmitted by the circuit group.
- **Drop** contains the number of data frames dropped by the circuit group.
- **TOTAL** provides an aggregate system-wide count for each reporting metric.

```
                    DECnet Router Statistics

     NAME        FRAMES:  Receive  Forward  Drop

1.  <xxxxxxx>              #         #       #
2.  <xxxxxxx>              #         #       #
3.  <xxxxxxx>              #         #       #
4.  <xxxxxxx>              #         #       #
    ------------------------------------------
     TOTAL                 #         #       #

PRESS: 'r' for reset, Down, Up, <- to exit
```

Figure 1-6: DECnet Router Statistics Screen

```
                         T1 Line Statistics

     NAME          Red Rx   Yel Rx   Blue Rx   Yel Tx   Line Errs
    ------------------------------------------------------------
1.  slot#_ds1_#     #        #        #         #        #
2.  slot#_ds1_#     #        #        #         #        #
3.  slot#_ds1_#     #        #        #         #        #
4.  slot#_ds1_#     #        #        #         #        #
    ------------------------------------------------------------
     TOTAL          #        #        #         #        #

PRESS: 'r' for reset, Down, Up, <- to exit
```

Figure 1-7: T1 Line Statistics Screen

1.9 T1 Line Statistics Screen

Availability of the T1 Line Statistics Screen (Figure 1-7) is also configuration-dependent. The system provides this screen only if you have established one or more T1 lines during system configuration. This screen provides a line-by-line analysis of T1 alarm conditions. If you wish to examine T1 alarms at a greater level of detail, you can use the NCL GET command (Section 2.8) to obtain a complete listing of T1 alarm statistics maintained by the system.

- **Name** lists each T1 line by slot number and connector designator.
- **Red Rx** contains the number of received Red alarms. A red alarm indicates that the remote site is not transmitting a valid T1 data frame. After detecting a Red alarm, the system generates a Yellow alarm.
- **Yel Rx** contains the number of received Yellow alarms. A yellow alarm indicates that the remote site is not receiving a valid T1 data frame.
- **Blue Rx** contains the number of received Blue alarms. A blue alarm indicates an all-1's data pattern (two consecutive frames having less than three zeros in the data stream).
- **Yel Tx** contains the number of transmitted Yellow alarms. Since a Yellow alarm is generated by the detection of a Red alarm, this metric and **Red Rx** should be equal.
- **Line Errs** contains the number of bi-polar errors.
- **TOTAL** provides an aggregate system-wide count for each reporting metric.

2. Network Command Language Interpreter

This section of the *Wellfleet Systems Operator's Guide* provides a description of the Network Command Language Interpreter (NCL). NCL is an on-line, dual-function language that:

1. Enables you to acquire detailed information about system operations by providing a series of commands that access the system's hierarchical database.
2. Provides a series of commands that control specific entities within the Wellfleet system.

You access NCL from the Main Menu . Use the UPARROW (⇑) or DOWNARROW (⇓) to position the cursor at **Network Command Language Interpreter**, then press RETURN--or, you may simply press the <2> key. After you press RETURN or type <2>, the system displays the NCL Command Screen (Figure 2-1).

In common with all other Wellfleet system screens, the NCL Command Screen displays a standard banner at the top of the screen. The banner shows installation-specific information (company and system name) along with date/time and session information. The NCL prompt, in the form of the system name followed a colon, appears at the lower left of the screen. A twenty-line area between the banner and the NCL prompt provides a window for data display.

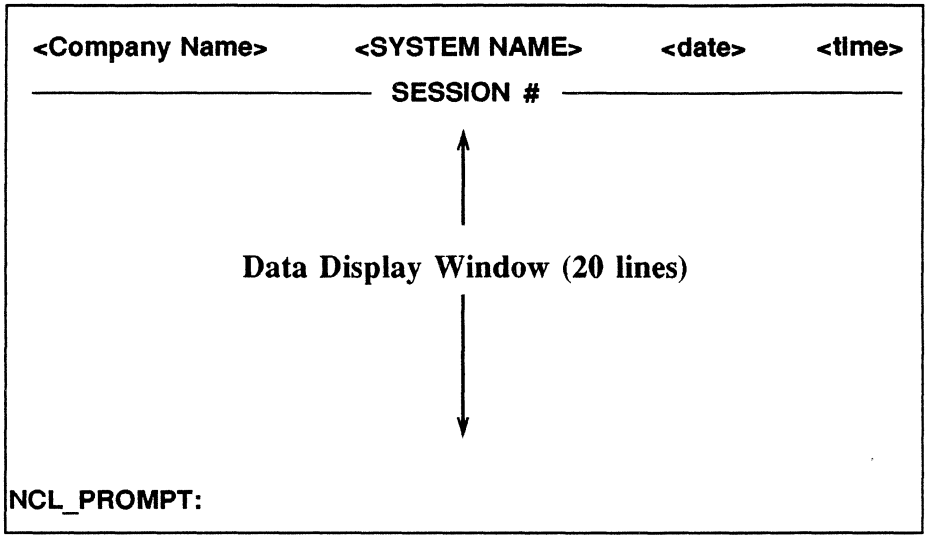

Figure 2-1: NCL Command Screen

The following sections provide descriptions of all NCL commands. Each individual command is described in terms of its function, syntax, and use. All commands require a terminating carriage return, designated by ↵ in the syntax examples.

NOTE

If you replace system diskettes, be sure to use the INSERT and REMOVE commands. Refer to Sections 2.10 and 2.16 for descriptions of these two commands.

2.1 BOOT

Purpose: BOOT re-boots the system.

Abbreviation: BOOT

Syntax: **boot**↵

where:

boot
Is the BOOT command

Example: `boot`↵

Notes: Use the following procedure to re-boot the system:

1. Type <BOOT>, followed by RETURN at the NCL prompt.
2. Enter the system password followed by RETURN in response to the **Enter current password** prompt. If you have not previously assigned a system password, or if you have removed password protection, the system does not prompt for a password.
3. Type <y> at the **Do you want to reboot the system? [y/n]** prompt. The system displays **REBOOTING THE SYSTEM**, and returns you to the Main Menu when it completes the reboot. If you do not wish to reboot the system, type <n> at **Do you want to reboot the system? [y/n]**. The system displays **Boot aborted** and returns you to the NCL prompt.

2.2 COPYTODISK

Purpose: COPYTODISK copies an ASCII file from the system diskette to a backup diskette.

Abbreviation: COPYTODISK

Syntax: **copytodisk <source> <destination>**

where:

copytodisk
Is the COPYTODISK command

source
Is the required name of the ASCII file to be copied

destination
Is the required name of the file on the backup diskette

Examples: **`copytodisk config config.bck`**

Copy the *config* file to a backup diskette under the filename config.bck

`copytodisk log.7 inc_log7.bck`

Copy log.7 to a backup diskette under the filename inc_log7.bck

Notes: Use the following procedure to make a copy of *config*, a log file, or any other ASCII file on the system diskette:

1. At the NCL prompt, enter the COPYTODISK command and specify the source and destination files. **Do not press `RETURN`.**
2. Remove the system diskette.
3. Insert the formatted target diskette.

4. Press RETURN.
5. Wait for the system to complete the copy operation.
6. Remove the target diskette.
7. Insert the system diskette.
8. Press RETURN.

The COPYTODISK command does not support the copying of binary files, nor does it support the use of wildcards in file names. During the copy operation, the system suspends disk logging activity.

2.3 DELETE

Purpose: DELETE removes a file from the system diskette.

Abbreviation: DEL

Syntax: **del <filename>↵**

where:

del
Is the DELETE command

filename
Is the required name of the file to be deleted

Examples: **`del test`↵**

Delete a file called test

`del config.bu`↵

Delete a file called config.bu

`del crash/K9`↵

Delete a file called K9 in the subdirectory crash

Notes: The DELETE command does not prompt you to verify that you wish to delete the file specified by <filename>. Consequently, you must be certain that you do wish to delete the specified file. Once you have deleted a file you cannot recover it.

2.4 DIRECTORY

Purpose: DIRECTORY lists the files in a directory.

Abbreviation: DIR

Syntax:

dir {filename}↵

where:

dir
Is the DIRECTORY command

filename
Is the optional name of a directory file

Examples:

`dir↵`

List all files in the root directory

`dir crash↵`

List all files in the directory file, crash

Notes: The DIR command defaults to the root directory. It provides a tabular listing of files in the root directory or in the directory file specified by {filename}. The listing provides file-specific information in the following format:

<file name> <ext> <size> <date> <time>

<file name> contains the file name.

<ext>, when present, contains the file extension.

<size> contains either a numerical value indicating the size of the file in bytes, or "**dir**" indicating a directory file.

<**date**> and <**time**> contain the date and time of file creation in month/ date/year and hour/minute/second formats.

2.5 DISABLE

Purpose: DISABLE removes an application software module, a circuit, or a managed object from service.

Abbreviation: DIS

Syntax:

dis <sw_entity>↵

where:

dis
Is the DISABLE command

sw_entity
Is the required designator for an application software module as follows:

ip	for the DoD IP Router
ib	for the Learning Bridge
drs	for the DECnet Router

dis <cct.> <name>↵

where:

dis
Is the DISABLE command

cct.
Is the required designator for a circuit

name
Is the required circuit name as entered into the *config* file

dis <mo>↵

where:

dis
Is the DISABLE command

mo
Is the required designator for a managed object (refer to the LIST command for a description of managed object designators)

Examples:

dis ip↵

Disable the IP Router

dis cct.ds1_21↵

Disable a circuit named ds1_21

dis line.slot2_ds1_1↵

Disable the T1 interface that is established by the DS1_1 connector in Slot 2

dis echo↵

Disable TCP echo service

Notes:

When disabling entities you may use system codes to identify specific objects. Refer to the LIST command for a description of system codes. You may also use the asterisk (*) as a wildcard with the DISABLE command. For example:

dis cct.*↵ disable all circuits

dis line.*↵ disable all lines

2.6 ENABLE

Purpose: ENABLE places an application software module, a circuit, or a managed object into service.

Abbreviation: E

Syntax:

e <sw_entity>↵

where:

e
Is the ENABLE command

sw_entity
Is the required designator for an application software module:

ip	for the DoD IP Router
ib	for the Learning Bridge
drs	for the DECnet Router

e <cct.> <name>↵

where:

e
Is the ENABLE command

cct.
Is the required designator for a circuit

name
Is the required circuit name as entered in the *config* file

e <mo>↵

where:

e
Is the ENABLE command

mo
Is the required designator for a managed object (refer to the LIST command for a description of managed object designators)

Examples:

e ip↵

Enable the IP Router

e cct.ds1_21↵

Enable a circuit named ds1_21

e line.slot2_ds1_1↵

Enable the T1 interface that is established by the DS1_1 connector in Slot 2

e echo↵

Enable TCP echo service

Notes:

Use the ENABLE command to restore entities previously removed by DISABLE, or those entities not auto-enabled.

When enabling entities you may use system codes to identify specific objects. Refer to the LIST command for a description of system codes. You may also use the asterisk (*) as a wildcard with the ENABLE command. For example:

e cct.*↵ enable all circuits

e line.*↵ enable all lines

2.7 EXIT

Purpose: EXIT leaves the NCL Interpreter and returns to the Main Menu.

Abbreviation: EXIT

Syntax: **exit**↵

where:

exit
Is the EXIT command

Example: `exit`↵

2.8 GET

Purpose: Display the value of a database parameter.

Abbreviation: G

Syntax: **g <DB_path>**↵

where:

g
Is the GET command

DB_path
Is the required path name to a database parameter

Examples: `g telnet.tx_bytes`↵

Display the contents of the TELNET tx_bytes counter

`g lb.jrb.xmit`↵

Display the contents of the counter that records the number of learning bridge frames transmitted across the circuit named jrb

`g *`↵

Display all database parameters with their current values

Notes: When displaying the value of a parameter you must provide a complete database parameter path. You may use system codes to identify parameters. Refer to the LIST command for a description of database structure and the use of system codes. You may also use the asterisk (*) as a wildcard with the GET command. Table 2-1 shows how to use GET to obtain

complete database parameter values for the managed objects of greatest interest.

Table 2-1: Managed Objects MIB Values

Managed Object	Command Syntax
DECnet Router	`g drs.*↵`
IP Router	`g ip.*↵`
Learning Bridge	`g lb.*↵`
circuits	`g cct.*↵`
T1 lines	`g lines.*↵`

2.9 HELP

Purpose: HELP displays a summary listing of NCL commands.

Abbreviation: H

Syntax: **help**↵

where:

help
Is the HELP command

Example: `help`↵

Notes: The HELP command invokes a screen that provides a summary of NCL commands and syntax.

2.10 INSERT

Purpose: INSERT mounts the system diskette.

Abbreviation: INSERT

Syntax: **insert**↵

where:

insert
Is the INSERT command

Example: `insert`↵

Notes: The INSERT command establishes the logical connection between the operating system and the system drive.

If you want to replace the current system diskette with another, first issue the REMOVE command to dismount the current diskette, then (after swapping diskettes) issue the INSERT command to mount the new diskette.

2.11 LIST

Purpose: LIST displays all or a portion of the system's hierarchical database tree.

Abbreviation: L

Syntax: **l <DB_path>↵**

where:

l

Is the LIST command

DB_path

Is an optional path name to a database variable

Examples: `l↵`

Display managed objects

`l *↵`

Display entire database

`l ip↵`

Display second level of ip database

`l ip.ip_interface↵`

Display third level branch of ip database hierarchy

`l ip.*↵`

Display entire ip database

Notes: The system database is a repository for all data gathered (and used) by the system. Within the database, the system organizes data as a series of hierarchical groups of variables or parameters. At the top level of the

hierarchy are "managed objects", defined as portions of system resources that enable network services. Below the managed objects are an arbitrary number of additional levels that describe the object in terms of instances (multiple occurrences of the object within the system) and attributes (parameter values). Some of these attributes are counters whose values are displayed when you inspect the statistics screens previously described in Section 1 of this guide. Other attributes contain additional information providing a more detailed description of system operations.

The LIST command displays all or a portion of the hierarchical database structure. It tells you what specific data the database contains, and provides you with path names to specific data items. After obtaining the path name, you can use the NCL GET command to obtain actual parameter values.

Table 2-2 shows how to use LIST to obtain complete database displays for the managed objects of greatest interest.

Table 2-2: Managed Objects Display

Managed Object	Command Syntax
DECnet Router	`l drs.*↵`
IP Router	`l ip.*↵`
Learning Bridge	`l lb.*↵`
circuits	`l cct.*↵`
T1 lines	`l lines.*↵`

Figures 2-2 through 2-7 show annotated database hierarchical displays similar to the ones that are generated by the commands listed in Table 2-2. Notice that each figure identifies those parameters that are displayed on system statistics screens, and those parameters that are set to zero when you reset a statistics screen.

The number of managed objects within a system is configuration-dependent. To obtain a list of managed objects resident within your system, type

```
LIST↵
```

at the NCL prompt. Figure 2-8 shows an actual list of managed objects.

The leftmost column in Figure 2-8 lists all managed objects within the system. The **map=** column designates the system slot(s) within which the managed object resides. To translate the map, start with the least significant digit and move to the left converting each individual digit to its binary equivalent. For example the value "24" translates to:

0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0

Each binary digit designates an LN/CN slot with the least significant digit of the binary representation designating Slot 0 and the most significant digit designating Slot 15. Using the list in Figure 2-8, you can see that the IP Router application software module is installed in Slots 2 and 5.

The rightmost column of Figure 2-8, **code=**,

contains a code value, an object-identifier component that corresponds to the variable listed to its left. Subsequent examples illustrate the use of these code values.

Should you wish to inspect a managed object at the next level of detail, use the LIST command followed by either the managed object identifier (from the leftmost column of Figure 2-8) or code value for the managed object (the rightmost column of Figure 2-8). The following paragraphs provide a summary description of database hierarchical structure. They guide you through a portion of the IP Router database. The IP Router is used as an example because it extends downward through five hierarchical levels. Most other managed objects have a less-complex structure.

To inspect the IP Router type one of the following at the NCL prompt.

```
1 ip↵
```

```
1 5 ↵
```

Note on Figure 2-8 that 5 is the code for ip, and can be substituted for ip. Because code identifiers are subject to change with software releases, be sure to verify code values before you use them to access the system database.

After you type `list ip` or `list 5` and press RETURN, the system displays the next (second) level of the IP Router database (shown in Figure 2-9).

Figure 2-9 points to two branches in the IP

Router database. To examine the **ip_interface** branch, type the following at the NCL prompt.

```
1 5.0↵
```

Note that if you must supply the entire database path with each numeral (code value) separated by a period. After you type <list 5.0> and press RETURN, the system displays the ip_interface branch of the next (third) level of the IP Router database (Figure 2-10). Note that this portion consists of a list of IP addresses in dotted decimal notation.

To continue downward through this branch, type the following at the NCL prompt.

```
1 5.0.n↵
```

where n is the code for a specific IP address

After you type <list 5.0.n> and press RETURN, the system displays the next (fourth) level of the IP database (shown in Figure 2-11). This level contains various parameters for the specified interface. If you wish, you can proceed one level deeper into the IP hierarchy by typing the following:

```
1 5.0.n1.n2↵
```

where n_1 is the code for a specific IP address, and n_2 is the code for a specific parameter group (using Figure 2-11 as an example, 3 for **drop**, 6 for **icmp_rx**, and 7 for **icmp_tx**).

Each of these commands displays a specific portion of the hierarchy fifth level as shown in Figure 2-12. Parameters at this level of detail quantify reasons for dropping packets and the types of received and transmitted ICMP messages.

LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4
drs			
	total		
		aged_pkt_loss	
		node_unreach	
		node_out_of_range	
		oversized_pkt_loss	
		pkt_format_error	
		rout_update_loss	
		verif_reject	
	cg		
		cg_names	
			trans_pkts_recv *
			trans_pkts_sent *
			circuit_down
			init_fail
			drop *

Level 4 attributes exist for each circuit group listed in Level 3. Those attributes designated with * are displayed on the DECnet Router Statistics Screen. All circuit-group-specific Level 4 attributes are zeroed when the DECnet Router Statistics Screen is reset.

Figure 2-2: DECnet Router Database Hierarchy

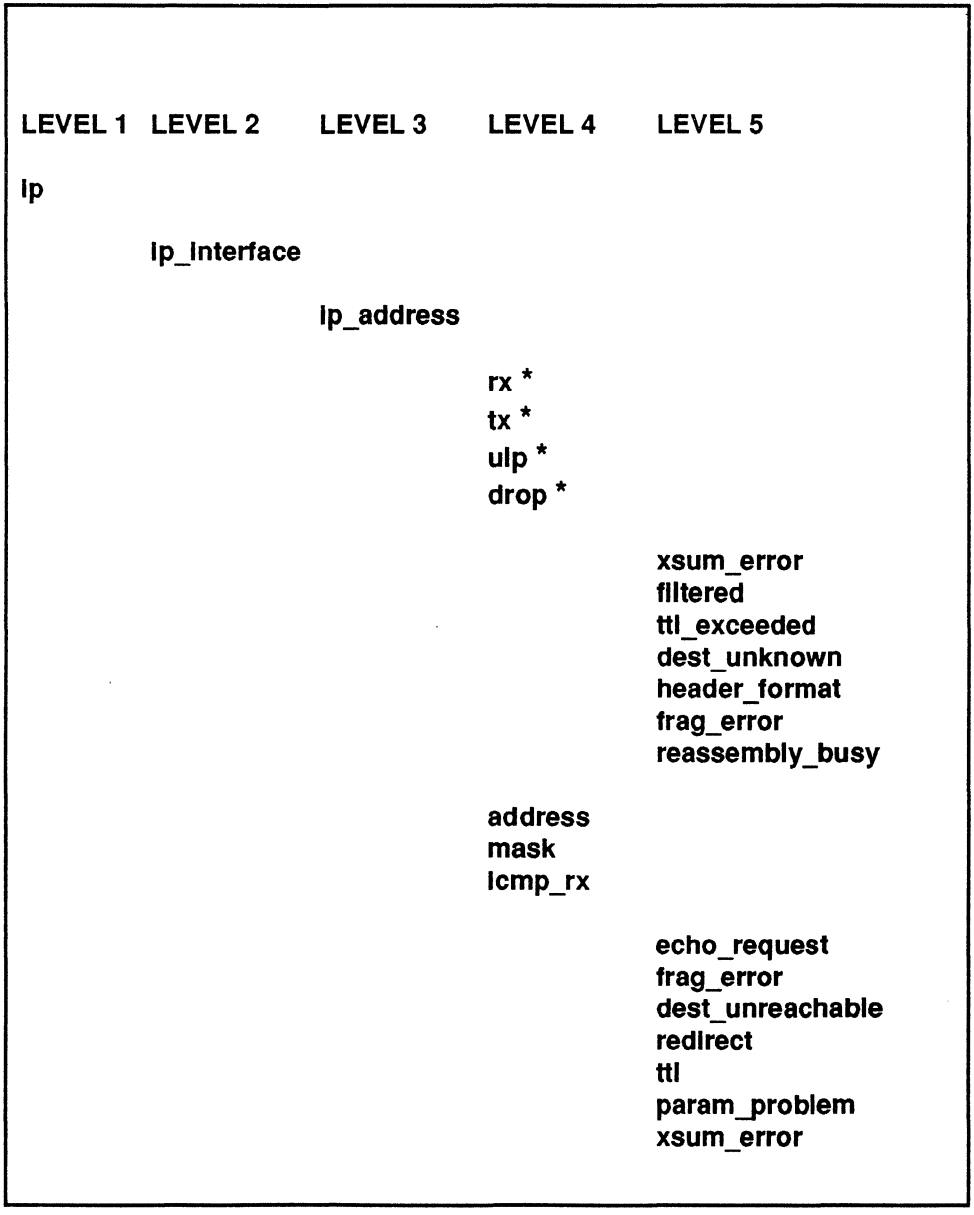

Figure 2-3: IP Router Database Hierarchy (1 of 2)

LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5
			icmp_tx	
				echo_request
				frag_error
				dest_unreachable
				redirect
				ttl
				param_problem
	ip_route_table			
		net_tree		
			cache_hits	
			search_count	
			search_depth	
			node_count	
			node_depth	
			mem_used	

Levels 4 and 5 attributes exist for each IP address listed in Level 3. Those attributes designated with * are displayed on the DoD IP Router Statistics Screen. All IP Address-specific Level 4 attributes (with the exception of **address** and **mask**) and all Level 5 attributes are zeroed when the DoD IP Router Statistics Screen is reset.

Figure 2-3: IP Router Database Hierarchy (2 of 2)

LEVEL 1	LEVEL 2	LEVEL 3
lb		
	circuit_name	
		recv *
		xmit *
		flood *
		drop_listen
		drop_protocol
		drop_src_addr
		drop_dst_addr
		drop_dest_local
		drop_loadbal_noprotcf
		drop_no_cg_from_cgm
		fwd_protocol
		fwd_dst_addr
		fwd_mcast_addr
		fwd_loadbal

Level 3 attributes exist for each circuit group listed in Level 2. Those attributes designated with * are displayed on the Learning Bridge Statistics Screen. All circuit-group-specific Level 3 attributes are zeroed when the Learning Bridge Statistics Screen is reset.

Figure 2-4: Learning Bridge Database Hierarchy

LEVEL 1	LEVEL 2	LEVEL 3
cct		
	circuit_name	
		octets_rx_ok *
		frames_rx_ok *
		total_rx_err *
		octets_tx_ok *
		frames_tx_ok *
		total_tx_err *
		test_cmd_tx
		xid_cmd_tx
		test_rsp_tx
		xid_rsp_tx
		test_cmd_rx
		xid_cmd_rx
		test_rsp_rx
		xid_rsp_rx
		unrecog_pdu
		deferred_tx
		late_colln_tx
		excessv_colln_tx
		babl_error_tx
		net_buffer_tx
		lcar_tx
		uflo_tx
		fcs_error_rx
		alig_error_rx
		lack_resc_error_rx
		to_long_error_rx
		oflo_rx

Figure 2-5: LAN Circuits Database Hierarchy (1 of 2)

LEVEL 1	LEVEL 2	LEVEL 3
		frams_incomp_rx
		merr
		cerr
		dls_ret_rx
		rcv_desc_cnt
		xmt_desc_cnt
		dls_ring_cnt

Level 3 attributes exist for each LAN circuit listed in Level 2. Those attributes designated with * are displayed on the Circuit Statistics Screen. All circuit-specific Level 3 attributes are zeroed when the Circuit Statistics Screen is reset.

Figure 2-5: LAN Circuits Database Hierarchy (2 of 2)

LEVEL 1	LEVEL 2	LEVEL 3
cct		
	circuit_name	
		octets_rx_ok *
		frames_rx_ok *
		total_rx_err *
		octets_tx_ok *
		frames_tx_ok *
		total_tx_err *
		test_cmd_tx
		xid_cmd_tx
		test_rsp_tx
		xid_rsp_tx
		test_cmd_rx
		xid_cmd_rx
		test_rsp_rx
		xid_rsp_rx
		unrecog_pdu
		uflo_tx
		rejects_tx
		lack_resc_error_rx
		oflo_rx
		frams_incomp_rx
		bad_frames_rx
		frmr_frames_rx
		rejects_rx

Figure 2-6: Point-to-Point Circuits Database Hierarchy (1 of 2)

LEVEL 1	LEVEL 2	LEVEL 3
		runts_rx
		t1_tos
		merr
		dls_ret_rx
		rcv_desc_cnt
		xmt_desc_cnt
		dls_ring_cnt

Level 3 attributes exist for each point-to-point circuit listed in Level 2. Those attributes designated with * are displayed on the Circuit Statistics Screen. All circuit-specific Level 3 attributes are zeroed when the Circuit Statistics Screen is reset.

Figure 2-6: Point-to-Point Circuits Database Hierarchy (2 of 2)

LEVEL 1	LEVEL 2	LEVEL 3
line		
	connector	
		excess_bipolar_vio
		bipolar_vio*
		bvcs
		excess_bit_err
		bit_err
		excess_ecs
		ecs
		out_of_frame
		err_sframe
		ryel_alarm *
		rcv_lcar
		rblue_alarm *
		rred_alarm *
		tyel_alarm *

Level 3 attributes exist for each connector listed in Level 2. Attributes designated with * are displayed on the T1 Lines Statistics Screen. All Level 3 attributes are zeroed when the T1 Lines Statistics Screen is reset.

Figure 2-7: T1 Lines Database Hierarchy

```
mgr             map=0004    code=0
driver          map=0034    code=1
cct             map=0034    code=2
lb              map=0010    code=3
drs             map=0010    code=4
ip              map=0024    code=5
svc             map=0024    code=9
dmap            map=0034    code=10
buf             map=0034    code=11
mem             map=0034    code=13
name            map=0004    code=14
timer           map=0034    code=15
alarm           map=0034    code=16
boot            map=0034    code=20
line            map=0004    code=21
tcp             map=0004    code=22
echo            map=0004    code=23
telnet          map=0004    code=24
snmp            map=0004    code=25
mib             map=0024    code=26
hw              map=0034    code=27
```

Figure 2-8: Sample Managed Objects Display

```
ip_interface        map=0004    code=0
ip_route_table      map=0004    code=1
```

Figure 2-9: IP Database Hierarchy (Second Level)

```
192.32.1.94          map=0004    code=1
192.32.1.194         map=0004    code=2
```

Figure 2-10: Sample IP Database Hierarchy (Third Level)

```
rx                   map=0004    code=0
tx                   map=0004    code=1
ulp                  map=0004    code=2
drop                 map=0004    code=3
address              map=0004    code=4
mask                 map=0004    code=5
icmp_rx              map=0004    code=6
icmp_tx              map=0004    code=7
```

Figure 2-11: Sample IP Database Hierarchy (Fourth Level)

```
<drop>xsum_error              map=0004   code=0
<drop>filtered                map=0004   code=1
<drop>ttl_exceeded            map=0004   code=2
<drop>dest_unknown            map=0004   code=3
<drop>header_format           map=0004   code=4
<drop>frag_error              map=0004   code=5
<drop>reassembly-busy         map=0004   code=6

<icmp_rx>echo_request         map=0004   code=0
<icmp_rx>frag_error           map=0004   code=1
<icmp_rx>dest_unreachable     map=0004   code=2
<icmp_rx>redirect             map=0004   code=3
<icmp_rx>ttl                  map=0004   code=4
<icmp_rx>param_problem        map=0004   code=5
<icmp_rx>xsim_error           map=0004   code=6

<icmp_tx>echo_reply           map=0004   code=0
<icmp_tx>frag_error           map=0004   code=1
<icmp_tx>dest_unreachable     map=0004   code=2
<icmp_tx>redirect             map=0004   code=3
<icmp_tx>ttl                  map=0004   code=4
<icmp_tx>param_problem        map=0004   code=5
```

Figure 2-12: Sample IP Database Hierarchy (Fifth Level)

2.12 LOG

Purpose: LOG displays the event log buffer.

Abbreviation: LOG

Syntax: **log**↵

where:

log
Is the LOG command

Example: `log`↵

Notes: The LOG command provides access to the RAM event log. It is functionally equivalent to selecting Event Log from the Main Menu.

By default, the system allocates a 50-item circular (FIFO) buffer to record event messages. If you enable logging during the configuration process, the system also schedules needed resources to periodically write the contents of the event log buffer to a default ASCII file (named *log*) on the system diskette.

When you establish a disk logging session, you specify both the size of the event log buffer and the name of the event file on the system diskette. You also may implement a log-file numbering feature that creates up to 10 log files (identified by a sequential numeric suffix, 0 through 9). With this feature enabled, the system creates a new log file each time it is rebooted, either with the RESET button or with the NCL BOOT command. Prior to creating the new file, it closes the previous log file, appends a numeric suffix to the file

name, and saves the file on the system diskette.

LOG provides immediate access to the event log buffer; the NCL TYPE command provides access to the periodically-updated event log file. Consequently, LOG is the preferred method of obtaining the most recent event log records. TYPE, in contrast, provides the only means of inspecting historical (closed) event log records.

Before displaying the contents of the event log buffer, verify that page mode is enabled (refer to PAGE) so that the buffer contents are displayed one screen at a time.

If the event log exceeds twenty lines, the console displays **-- MORE --** below the last line of displayed data. To view additional data:

1. Press `RETURN` for one more line.
2. Press any other key for an additional screen of data.
3. Press a number from `1` through `9` to display that number of additional lines.
4. Press the `LEFTARROW` key (⇐) to return to the NCL prompt.

When the system reaches the end of the event log buffer, it positions the cursor at the NCL prompt.

2.13 PAGE

Purpose: PAGE enables/disables page mode. In page mode, output is sent to the console one screen at a time.

Abbreviation: PAGE

Syntax: **page**↵

where:

page
Is the PAGE command

Example: **page**↵

Notes: The PAGE command toggles page mode (the default is enabled). Page mode is useful for viewing files greater than twenty lines in length. With page mode enabled, output is sent to the console one screen at a time. Refer to the TYPE and LOG commands for information on viewing files.

2.14 PASSWORD

Purpose: PASSWORD assigns, changes, or removes system password protection.

Abbreviation: PASSWORD

Syntax: **password**↵

where:

password
Is the PASSWORD command

Example: `password`↵

Notes: Use the following procedure to assign a system password:

1. Type `password`, followed by RETURN, at the NCL prompt.
2. Assign a password by entering a string of one to fourteen alphanumeric characters, followed by RETURN, in response to the **Enter new password** prompt.
3. Confirm the password by entering the identical string, followed by RETURN, in response to the **Enter new password again** prompt. After you confirm the password, the system returns you to the NCL prompt.

Use the following procedure to change the system password:

1. Type `password`, followed by RETURN, at the NCL prompt.
2. Enter the current password, followed by RETURN, in response to the **Enter current password** prompt.

3. Enter the new password (an alphanumeric string of from one to fourteen characters), followed by RETURN, in response to the **Enter new password** prompt.
4. Confirm the new password by entering the identical string, followed by RETURN, in response to the **Enter new password again** prompt. After you confirm the password, the system returns you to the NCL prompt.

Use the following procedure to remove password protection:

1. Type `password`, followed by RETURN, at the NCL prompt.
2. Enter the current password, followed by RETURN, in response to the **Enter current password** prompt.
3. Press RETURN in response to the **Enter new password** prompt.
4. Confirm the removal of password protection by pressing RETURN in response to the **Enter new password again** prompt. The system then displays **Password protection removed**, and returns you to the NCL prompt.

2.15 PING

Purpose: PING sends an Internet Control Message Protocol (ICMP) echo request to a specified IP address and waits for a reply.

Abbreviation: PING

Syntax: **ping <remote_host> {count} {timeout}↵**

where:

ping
Is the PING command

remote_host
Is the remote host IP address, in dotted decimal notation

count
Is the optional number of times to repeat the PING command

timeout
Is the optional timeout (in seconds) for each ping

Examples: `ping 192.32.1.62↵`

Pings IP address 192.32.1.62 one time (the default), and waits five seconds (the default) for a response

`ping 192.32.1.62 1000 1↵`

Pings IP address 192.32.1.62 one thousand times, and waits one second for the response to each ping

```
ping 192.32.1.62 2↵
```

Pings IP address 192.32.1.62 twice, and waits five seconds (the default) for the response to each ping

Notes:

PING is a program used within the Internet community to test the reachability of remote hosts. PING transmits an ICMP echo request message to an IP address and waits for a response. If the system receives an echo response within the designated or default interval, the console displays a message indicating that the target address is "alive". If the system does not receive a response within the specified interval, it displays a message indicating that the target did not respond, and prompts you to press any key to continue. The PING command does not support loopback (pinging your own system), broadcast addresses, or timeouts of less than one second.

2.16 REMOVE

Purpose: REMOVE dismounts the system diskette.

Abbreviation: REMOVE

Syntax:

remove↵

where:

remove
Is the REMOVE command

Example:

```
remove↵
```

Notes: The REMOVE command breaks the logical connection between the operating system and the system drive. As a result, the drive is no longer accessible, and any attempt to access the drive (e.g. DIR or TYPE) is unsuccessful.

If you want to replace the current system diskette with another, first issue the REMOVE command to dismount the current diskette, then (after swapping diskettes) issue the INSERT command to mount the new diskette.

2.17 RESET

Purpose: Set a database parameter to zero.

Abbreviation: RESET

Syntax: **reset <DB_path>↵**

where:

reset
Is the RESET command

DB_path
Is the required path name to a database parameter

Examples: **`reset telnet.tx_bytes↵`**

Zero the TELNET tx_bytes counter

`reset lb.jrb.xmit↵`

Zero the counter that records the number of learning bridge frames transmitted on the circuit named jrb

Notes: When resetting a parameter you must provide a complete parameter path. You may use system codes to identify parameters. Refer to the LIST command for a description of system codes. You may also use the asterisk (*) as a wildcard with the RESET command. For example:

`reset cct.jrb.*↵`

Reset all circuit parameters associated with the circuit named jrb

2.18 STAMP

Purpose: STAMP displays the software image stamp.

Abbreviation: STAMP

Syntax:

stamp↵

where:

stamp
Is the STAMP command

Example:

```
stamp↵
```

Notes: You can use the software image stamp to identify your software revision level.

2.19 TELNET

Purpose: TELNET uses the Internet TELNET protocol to establish a connection to a remote host.

Abbreviation: TELNET

Syntax: **telnet <nnn.nnn.nnn.nnn>↵**

where:

telnet
Is the TELNET command

nnn.nnn.nnn.nnn
Is the required IP address, in dotted decimal notation, of the remote host

Example: **`telnet 192.32.1.94↵`**

Establish a TCP connection to a remote host whose IP address is 192.32.1.94

Notes: The TELNET command, which requires the IP Router application software module, enables you to establish a TCP connection with a login server at a remote site. To establish the connection, the command requires the IP address (in dotted decimal notation) of the remote host. Once a connection is established, TELNET passes keystrokes from your system to the remote host.

When attempting to TELNET to a remote Wellfleet system, keep in mind that Wellfleet supports a maximum of two simultaneous TELNET sessions (one incoming and the other outgoing), and a maximum of three simultaneous TCP connections.

2.20 TIME

Purpose: TIME sets the system clock and/or calendar.

Abbreviation: TIME

Syntax: **time <{mm/dd/yy} {hh:mm:ss}>↵**

where:

time
Is the TIME command

mm/dd/yy
Is the optional date in month/day/year format

hh:mm:ss
Is the optional time in military (24 hour) format

Examples: `time 02/29/92 14:15:00↵`

Set the time and date to 2:15 PM on February 29, 1992

`time 12/2/89↵`

Set the date to December 2, 1989

`time 1:00:00↵`

Set the time to 1:00 AM

Notes: The TIME command requires a minimum of one argument (either date or time). It cannot be used without arguments to obtain the current date and time.

2.21 TYPE

Purpose: TYPE reads an ASCII file from the system diskette and displays the file on the console screen.

Abbreviation: TYPE

Syntax: **type <filename>↵**

where:

type
Is the TYPE command

filename
Is the required name of the ASCII file to be displayed

Examples: **`type config↵`**

Display the *config* file

`type log.3↵`

Display the log file log.3

`type crash/k9↵`

Display a file named K9 contained in a directory named crash

Notes: The TYPE command enables the display of all ASCII files (those created with the Configuration Editor, log, and crash files) saved on the system diskette. Before typing a file, you should verify that page mode is enabled (refer to PAGE) so that the file is displayed one screen at a time.

If the file exceeds twenty lines, the console displays **-- MORE --** below the last line of displayed data. To view additional data:

1. Press RETURN for one more line.
2. Press any other key for an additional screen of data.
3. Press a number from 1 through 9 to display that number of additional lines.
4. Press the LEFTARROW (⇐) to return to the NCL prompt.

When the console reaches the last line of the file, it displays **---- 'End of file' ----** and positions the cursor at the NCL prompt.

2.22 ! Command

Purpose: ! repeats the last NCL command.

Abbreviation: !

Syntax: **! {n}↵**

where:

!

Is the ! command

n

Is an optional numeric value that specifies the number of times to repeat the last NCL command

Example: **`get lb.jrb.recv`↵**

Use the GET command to obtain the number of packets received by the Learning Bridge circuit group jrb

!↵

Repeat the above GET command to obtain an updated packet count

3. Event Log

This section of the *Wellfleet Systems Operator's Guide* describes the event messages generated by the system and stored by the system in the event log. It describes the event log's structure, tells you how to access the log, and explains how to interpret log entries. The section also provides an alphabetical listing of the most frequently encountered event messages.

3.1 Event Log Structure

The event log is a circular (FIFO) buffer whose size is defined at system configuration. In the absence of a specifically configured logging session, the event log contains, by default, fifty entries. If you configure a disk logging session, you can specify the size of the event log up to a maximum of 999 entries. When the system generates event messages in excess of the log capacity, the oldest event message is overwritten by the most recent.

Each event log entry is composed of five fields as shown below:

<severity> <date> <time> <object> <event message>

<severity> contains the event message precedence:

1. **M** (for major) indicates the appearance or disappearance of a service.
2. **P** (for performance) indicates that a service, although still present, has degraded.

3. **W** (for warning) indicates that a service has behaved in an unexpected fashion.
4. **I** (for information) indicates routine system events.

<date> contains the date in mm/dd/yy format that the message was placed in the event log.

<time> contains the time in hh:mm:ss format that the message was placed in the event log.

<object> contains the name of the managed object that generated the event message.

<event message> contains the system-generated event message.

3.2 Accessing the Event Log

You access the event log from the Main Menu. Use the UPARROW (⇑) or DOWNARROW (⇓) to position the cursor at **Event Log**, then press RETURN--or, you may simply press the <4> key. After you press RETURN or type <4>, the system displays the event log. Figure 3-1 shows a sample event log display.

The event log header (**Top of Log : First event # n**) precedes the display of log contents. n specifies the sequence number of the event message that is first displayed. The system assigns a sequence number to each event message as it is placed in the event log. A value of n equal to one indicates that the log has not as yet exceeded its capacity. A value greater than one indicates that the log capacity has been exceeded and that the system has begun to overwrite the earliest event messages.

After the event log header the console displays the first nineteen log entries and the following prompt:

PRESS: UP, DOWN, LEFT, RIGHT, RETURN

```
---- Top of Log : First event # 1 ----
I 06/08/89 09:14:24 boot[2]: 'Last booted at 09:09:50 - 6/8/89'
I 06/08/89 09:14:24 boot[2]: 'Boot count = 504'
I 06/08/89 09:14:33 tcp: 'configuration complete'
I 06/08/89 09:14:35 mgr.auto_enable: 'auto-enabling 'cct.B_LINK1"
I 06/08/89 09:14:35 mgr.auto_enable: 'auto-enabling 'cct.B_LINK2"
I 06/08/89 09:14:35 mgr.auto_enable: 'auto-enabling 'cct.LAB_NET"
I 06/08/89 09:14:35 mgr.auto_enable: 'auto enabling 'ip"
I 06/08/89 09:14:35 cct.b_link1: 'Enable requested'
I 06/08/89 09:14:36 mgr.auto_enable: 'auto-enabling 'tcp"
I 06/08/89 09:14:36 ip: 'entity enabled'
I 06/08/89 09:14:39 tcp: 'entity enabled'
I 06/08/89 09:14:39 mgr.auto_enable: 'auto-enabling 'telnet"
I 06/08/89 09:14:40 telnet: 'entity enabled'
I 06/08/89 09:14:40 mgr.auto_enable: 'auto-enabling 'snmp"
I 06/08/89 09:14:40 cct.b_link2: 'Enable requested'
I 06/08/89 09:14:40 cct.lab_net: 'Enable requested'
I 06/08/89 09:14:40 cct.lab_net: 'Providing LLC1 service'
I 06/08/89 09:14:40 boot[2]: 'Board initialized'
I 06/08/89 09:14:40 cct.b_link1: 'Providing LLC2 service to remote'
PRESS: UP, DOWN, LEFT, RIGHT, RETURN
```

Figure 3-1: Event Log Display

To move through the event log display:

1. Press the UPARROW (⇑) or DOWNARROW (⇓) to scroll through the log one entry (line) at a time.
2. Press RETURN to scroll the next screen of log entries.

3. Press the RIGHTARROW (⇒) to move to the end of the log.

4. Press the LEFTARROW (⇐) to return to the Main Menu.

The event log trailer (**Bottom of Log : Last event # n**) follows the last log entry. n designates the sequence number of the last displayed entry.

3.3 Event Log Messages

This section contains an alphabetical list of the most commonly encountered event log messages. For each message, this section lists both the message precedence and the object(s) that generate the message, and provides an explanation of message contents.

Area Reach Chg Area #, Reachable

Precedence: I

Object: drs

Meaning: The DECnet Router has determined that a previously unreachable Area (designated by the area number, #) is now reachable.

Area Reach Chg Area #, Unreachable

Precedence: I

Object: drs

Meaning: The DECnet Router has determined that a previously reachable Area (designated by the area number, #) is now unreachable.

arp: nnn.nn.nn.nn/nnn.nn.nn.nn

Precedence: I

Object: ip.ip_interface

Meaning: ip has entered a host address (designated by the first IP address, **nnn.nn.nn.nn**) in the ARP table for a network identified by the second IP address.

auto-enabling <entity>

Precedence: I

Object: mgr.auto_enable

Meaning: mgr.auto_enable has begun to auto-enable the service designated by **<entity>**. Such services include drs, ip, lb, snmp, tcp, and telnet.

auto-enabling cct.<xxxxxxx>

Precedence: I

Object: mgr.auto_enable

Meaning: mgr.auto_enable has begun to auto-enable the circuit named <xxxxxxx>.

auto-enabling line.slot#_ds1_#

Precedence: I

Object: mgr.auto_enable

Meaning: mgr.auto_enable has begun to auto-enable the line established by connector n on Slot #.

bad configuration, using defaults

Precedence: W

Object: tcp

Meaning: TCP has completed configuration (after rejecting bad user-specified TCP configuration parameters, and substituting default parameters).

Blue alarm cleared

Precedence: W

Object: line.slot#_DS1_n

Meaning: The T1 line established by connector n on Slot # has cleared a blue alarm.

Board initialized

Precedence: I

Object: boot [#]

Meaning: The ACE board in Slot # has completed initialization.

Boot count = nnn

Precedence: I

Object: boot [2]

Meaning: nnn specifies the number of times the system has been booted. This number is stored on the system controller board. If the controller board is changed, this count will reflect the number stored in the new board.

CG Up CG <xxxxxxx>, Adj node=n.nn

Precedence: I

Object: drs

Meaning: The DECnet circuit group designated by <xxxxxxx> is enabled, establishing communications with the adjacent DECnet node designated by n.nn.

Circuit configuration missing

Precedence: W

Object: driver

Meaning: No circuits have been configured for the link module in Slot #.

configuration complete

Precedence: I

Object: tcp

Meaning: TCP has completed configuration (after using good user-specified TCP configuration parameters).

Enable requested

Precedence: I

Object: cct.<xxxxxxx>

Meaning: The circuit named <xxxxxxx> is requesting to be enabled.

entity already disabled

Precedence: W

Object: drs, ip, lb, snmp, tcp, telnet

Meaning: The object generating the message is disabled. Generated if a disabled entity receives a disable command.

entity already enabled

Precedence: W

Object: drs, ip, lb, snmp, tcp, telnet

Meaning: The object generating the message is enabled. Generated if an enabled entity receives an enable command.

entity disabled

Precedence: I

Object: drs, ip, lb, snmp, tcp, telnet

Meaning: The object generating the message has finished disabling in response to a disable command.

entity enabled

Precedence: I

Object: drs, ip, lb, snmp, tcp, telnet

Meaning: The object generating the message is enabled.

Excessive bipolar violations

Precedence: W

Object: line.slot#_DS1_n

Meaning: More than 1000 bipolar violations were received in a 250ms period.

Excessive errors

Precedence: W

Object: line.slot#_DS1_n

Meaning: More than 1000 line errors were received in a 250ms period.

Excessive frame bit errors

Precedence: W

Object: line.slot#_DS1_n

Meaning: More than 1000 frame (synchronization) errors were received in a 250ms period.

Expected module DISABLED, diagnostic failure

Precedence: M

Object: boot [2]

Meaning: The link module connected to the Master ACE (Slot 2) has failed diagnostics.

hash_add: FWD TBL FULL

Precedence: W

Object: lb

Meaning: The learning bridge forwarding table is full.

hash_filter_add: FILTER TBL FULL

Precedence: W

Object: lb

Meaning: The learning bridge filter table is full.

Last booted at <hh/mm/ss> - <mm/dd/yy>

Precedence: I

Object: boot [2]

Meaning: This message indicates when the system was last booted.

Local not hearing from remote

Precedence: P

Object: cct.<xxxxxxx>

Meaning: A disparity exists between the ends of a point-to-point circuit. The Wellfleet remote signal and sense priority requires that both ends of the circuit be designated as "active". This message is generated by an "active" side of a mis-matched circuit.

Local receiving from remote

Precedence: P

Object: cct.<xxxxxxx>

Meaning: Both ends a point-to-point circuit have been properly designated as "active" and the circuit is up.

Module # not as configured

Precedence: W

Object: driver

Meaning: There is a discrepancy between the link module in Slot # and the config file.

network disabled on nnn.nn.nn.nn

Precedence: I

Object: ip.ip_interface

Meaning: An IP network interface, designated by the dotted decimal address (**nnn.nn.nn.nn**) has disabled either in response to a disable command or because the circuit group has gone down.

network enabled on nnn.nn.nn.nn

Precedence: I

Object: ip.ip_interface

Meaning: An IP network interface, designated by the dotted decimal address (**nnn.nn.nn.nn**) has finished enabling.

no configuration, using defaults

Precedence: W

Object: tcp

Meaning: TCP has completed configuration (after using default TCP configuration parameters).

No ethernet circuits configured for slot

Precedence: W

Object: driver

Meaning: The *config* file contains no Ethernet circuit records for the designated Slot.

no network interfaces configured

Precedence: W

Object: ip.ip_interface

Meaning: The *config* file contains no network interface definitions.

Node Reach Chg Node #.##, Reachable

Precedence: I

Object: drs

Meaning: The DECnet Router has determined that a previously unreachable Node (designated by the node designator, #.##) is now reachable.

Node Reach Chg Node #.##, Unreachable

Precedence:

Object: drs

Meaning: The DECnet Router has determined that a previously reachable Node (designated by the node designator, #.##) is now unreachable.

port 23 connected to nnn.nn.nn.nn

Precedence: I

Object: telnet

Meaning: A telnet session has been established through the "well-known" telnet port.

port 23 disconnected from nnn.nn.nn.nn

Precedence: I

Object: telnet

Meaning: A telnet session has been terminated.

Providing LLC1 service

Precedence: I

Object: cct.<xxxxxxx>

Meaning: The circuit named <xxxxxxx> is enabled and providing link level control 1 (LLC1) service. The type of service (LLC1 or LLC2) is specified during system configuration.

Providing LLC2 service to remote

Precedence: I

Object: cct.<xxxxxxx>

Meaning: The circuit named <xxxxxxx> is enabled and providing link level control 2 (LLC2) service. The type of service (LLC1 or LLC2) is specified during system configuration.

received blue alarm

Precedence: W

Object: line.<xxxxxxx>

Meaning: Line <xxxxxxx> received a blue alarm.

received carrier loss

Precedence: W

Object: line.<xxxxxxx>

Meaning: Line <xxxxxxx> lost carrier. The line is down.

received red alarm

Precedence: W

Object: line.<xxxxxxx>

Meaning: Line <xxxxxxx> received a red alarm.

received yellow alarm

Precedence: W

Object: line.<xxxxxxx>

Meaning: Line <xxxxxxx> received a yellow alarm.

Red alarm cleared

Precedence: W

Object: line.slot#_DS1_n

Meaning: The T1 line established by connector n on Slot # has cleared a red alarm.

Responded to reset, service continued

Precedence: W

Object: cct.<xxxxxxx>

Meaning: Service has been established on a point-to-point circuit. Refer to the "**Unexpected remote reset to local**" message.

rip: nnn.nn.nn.nn/nnn.nn.nn.nn, cost C

Precedence: I

Object: ip.ip_interface

Meaning: ip has learned a route to a network (designated by the first IP address, nnn.nn.nn.nn) via a router designated by the second IP address with an cost of C (a value from 1 to 16).

Routing Update Loss

Precedence: I

Object: drs

Meaning: The DECnet Router has dropped a topology packet.

SQE absent (non 802.3 XCVR)

Precedence: W

Object: cct.<xxxxxxx>

Meaning: The LAN circuit named <xxxxxxx> has detected a loss of the Signal Quality Error (SQE) signal. SQE is always absent if an Ethernet standard transceiver rather than an 802.3 transceiver is used.

Too many circuits configured for slot

Precedence: W

Object: driver

Meaning: You have configured more than two T1 circuits for the specified slot.

Too many T1 circuits configured for module

Precedence: W

Object: driver

Meaning: You have configured more than two T1 circuits for the specified line.

Unexpected remote reset to local

Precedence: W

Object: cct.<xxxxxxx>

Meaning: This message occurs at one end of a point-to-point link when the connection is first being established. One end will come up before the other and receive a reset command from the other end when it comes up. The side that comes up first displays this message. This message appears in tandem with a "**Responded to reset, service continued**" message.

Yellow alarm cleared

Precedence: W

Object: line.slot#_DS1_n

Meaning: The T1 line established by connector n on Slot # has cleared a yellow alarm.

12 DeAngelo Drive, Bedford, MA 01730-2204 Tel: (617) 275-2400 Fax: (617) 275-5001