VMS Software

VSI OpenVMS

DECnet Network Management Utilities

Document Number: DO-DNTNMU-01A

Publication Date: April 2024

Operating System and Version: VSI OpenVMS IA-64 Version 8.4-1H1 or higher
VSI OpenVMS Alpha Version 8.4-2L1 or higher

DECnet Network Management Utilities

VMS Software

Preface

1. About VSI

VMS Software, Inc. (VSI) is an independent software company licensed by Hewlett Packard Enterprise to develop and support the OpenVMS operating system.

2. Intended Audience

The *VSI OpenVMS DECnet Network Management Utilities* is a reference to the formats and syntax of NCP and DTS/DTR commands. This manual is for users already familiar with DECnet for OpenVMS concepts. System managers can use NCP to configure, control, monitor, or test DECnet for OpenVMS software.

3. Document Structure

This document consists of four chapters and one appendix:

*   Chapter 1 provides a full description of the NCP utility, including the following topics:

    *   Command syntax

    *   Entering commands

    *   Command prompting

    *   Component and parameter syntax rules

    *   NCP HELP

*   Chapter 2 outlines the following NCP information:

    *   Invoking the utility

    *   Exiting the utility

    *   Directing output

    *   Restrictions or privileges for specific operations

*   Chapter 3 describes NCP commands, including format, parameters qualifiers, and examples.

*   Chapter 4 provides a description of the DTS/DTR utility

*   Appendix A lists DECnet for OpenVMS circuit and line devices.

*   Appendix B includes descriptions of circuit, line and node counters.

*   Appendix C contains descriptions of circuit, line, and link states, and their substates and transitions. This section also describes X25-SERVER module and DTE states, substates, and their transitions.

*   Appendix D includes a summary of NCP events in terms of their class and type.

4. Related Documents

Before using NCP, you should be familiar with the basic networking concepts described in the *VSI OpenVMS DECnet Guide to Networking*. DECnet for OpenVMS concepts and NCP usage information are presented in the *VSI OpenVMS DECnet Networking Manual*, which also describes the procedures needed to configure, manage, test, and monitor your network. Refer to that manual for more information about any NCP command listed in this manual.

The following functional specifications define Network Architecture (NA) Phase IV protocols to which all implementations of DECnet Phase IV adhere:

- *DECnet Network Architecture General Description*

- *Data Communications Message Protocol Functional Specification*

- *Network Services Protocol Functional Specification*

- *Maintenance Operation Protocol Functional Specification*

- *Data Access Protocol Functional Specification*

- *Routing Layer Functional Specification*

- *DNA Session Control Functional Specification*

- *DNA Phase IV Network Management Functional Specification*

- *Ethernet Node Product Architecture Specification*

- *Ethernet Data Link Functional Specification*

5. VSI Encourages Your Comments

You may send comments or suggestions regarding this manual or any VSI document by sending electronic mail to the following Internet address: `<docinfo@vmssoftware.com>`. Users who have VSI OpenVMS support contracts through VSI can contact `<support@vmssoftware.com>` for help with this product.

6. OpenVMS Documentation

The full VSI OpenVMS documentation set can be found on the VMS Software Documentation webpage at https://docs.vmssoftware.com.

7. Conventions

The following conventions are used in this manual:

Convention	Meaning
Ctrl/ *x*	A sequence such as **Ctrl/** *x* indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
PF1 *x*	A sequence such as PF1 *x* indicates that you must first press and release the key labeled PF1 and then press and release another key or a pointing device button.

Convention	Meaning
Return	In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)
...	A horizontal ellipsis in examples indicates one of the following possibilities: • Additional optional arguments in a statement have been omitted. • The preceding item or items can be repeated one or more times. • Additional parameters, values, or other information can be entered.
. . .	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
( )	In command format descriptions, parentheses indicate that you must enclose the options in parentheses if you choose more than one.
[ ]	In command format descriptions, brackets indicate optional choices. You can choose one or more items or no items. Do not type the brackets on the command line. However, you must include the brackets in the syntax for OpenVMS directory specifications and for a substring specification in an assignment statement.
[|]	In command format descriptions, vertical bars separate choices within brackets or braces. Within brackets, the choices are options; within braces, at least one choice is required. Do not type the vertical bars on the command line.
{ }	In command format descriptions, braces indicate required choices; you must choose at least one of the items listed. Do not type the braces on the command line.
bold text	This typeface represents the introduction of a new term. It also represents the name of an argument, an attribute, or a reason.
italic text	Italic text indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error *number*), in command lines (/PRODUCER= *name*), and in command parameters in text (where *dd* represents the predefined code for the device type).
UPPERCASE TEXT	Uppercase text indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
`Monospace type`	Monospace type indicates code examples and interactive screen displays. In the C programming language, monospace type in text identifies the following elements: keywords, the names of independently compiled external functions and files, syntax summaries, and references to variables or identifiers introduced in an example.
-	A hyphen at the end of a command format description, command line, or code line indicates that the command or statement continues on the following line.
numbers	All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated.

Chapter 1. NCP Description

The Network Control Program (NCP) is the utility program used by system and network managers to configure and control DECnet nodes. System managers can also use NCP to monitor network resources and test network components. This manual presents the syntax of the NCP commands used to perform these tasks.

Consult the *VSI OpenVMS DECnet Networking Manual* for detailed descriptions explaining the use of the NCP commands. After you read the appropriate sections of the *VSI OpenVMS DECnet Networking Manual* and know which NCP command (or series of commands) you need, use this manual for the syntax of the commands.

The description of NCP covers the following topics:

• Command syntax

• Entering commands

• Command prompting

• Component and parameter syntax rules

• NCP HELP Facility

The *VSI OpenVMS DECnet Networking Manual* also contains an overview of networking concepts and descriptions of the components that make up the network.

1.1. Command Syntax

To invoke NCP, type RUN SYS$SYSTEM:NCP at the DCL prompt. The *NCP>* prompt indicates that you are using the NCP utility program. When you receive this prompt, you can enter NCP commands.

The NCP command syntax has four parts: a command verb, a component, one or more parameters and, optionally, one or more qualifiers. Table 1.1 is an example of this syntax.

Table 1.1. Command Syntax Example

Command Verb	Component	Parameter	Qualifier
SHOW	ACTIVE LINES	CHARACTERISTICS	TO filespec
	KNOWN LINES	COUNTERS	
	LINE line-id	STATUS	
		SUMMARY	

For each command, you must supply a command verb and a component. Some commands also require one or more parameters. Command qualifiers are optional. Unless otherwise indicated, you can specify parameters in any order.

1.2. Entering Commands

Enter NCP commands as verbs, components, parameters, and qualifiers separated by spaces or tabs. To continue a long command to the next line, use the standard continuation line convention (a hyphen as the last character in the line). The *NCP>* prompt is not displayed on continuation lines; instead, continuation lines prompt with an underscore character. For example:

```
NCP> SET EXECUTOR –
_ ADDRESS 11 –
_ INCOMING TIMER 30 –
_ STATE ON
NCP>
```

In command files, use an exclamation point (!) to designate a comment line. NCP ignores hyphens within and at the end of a comment line. However, NCP does not ignore lines beginning with an exclamation point if they follow a command line ending with a hyphen.

When entering an NCP command, you can truncate any command verb, component name, parameter name, or qualifier name to its fewest unique letters. For example, the following two commands perform the same function:

```
NCP> SET EXECUTOR ADDRESS 11 STATE ON
NCP> SET EXE AD 11 STA ON
```

For clarity, all examples show the full command syntax.

Recalling Commands

You can recall NCP commands that you previously entered by pressing CTRL/B or the arrow keys in response to the *NCP>* prompt.

You can press CTRL/B or the up-arrow key up to 20 times to recall the last 20 commands you have typed. After you recall an NCP command, you can edit it and then reenter it. You can press the down-arrow key to recall the next (most recent) command in the NCP command recall buffer.

Using Wildcard Characters to Enter Commands

You can use the asterisk (*) and the percent sign (%) as wildcard characters in an NCP command line to represent NCP components by a general name, rather than specify each component name individually.

You can use the wildcard characters to represent the following component names:

- Node name

- Node address

- Circuit name

- Line name

- Object name

- Events

The asterisk wildcard represents one or more characters, while the percent sign represents a single character.

Rules for Using Wildcard Characters

The following rules define how you can use wildcard characters with component names:

- If the component name is a string, the wildcard character may occur at any location in the string. For example:

  ```
  NCP> LIST NODE ST%R STATUS
  ```

```
NCP> SHOW OBJECT M* CHARACTERISTICS
```

The first command requests a list of status information for all nodes with four-letter node names beginning with "ST" and ending with "R." The second command requests a listing of characteristics for all objects with names beginning with "M."

- For node addresses, which are represented by the format *area-number.node-number*, only the *node-number* portion of the node address (the numeral on the right side of the period) can contain a wildcard. For example, the following command sets a COUNTER TIMER value of 45 seconds for all nodes in area 4:

```
NCP> SET NODE 4.* COUNTER TIMER 45
```

  Specifying a node address such as *.5 is invalid because the *area-number* cannot contain a wildcard.

- In a node address, a wildcard character cannot be combined with a numeral to represent a partial *node-number*. The node addresses 4.* and 4.% contain valid uses of the wildcard characters, but the node addresses 4.%2 and 4.1* are invalid.

- For events, which are represented by the format *class.type*, only the *type* portion of the event (the numeral on the right side of the period) can contain a wildcard. For example, the following command specifies that all class 2 events are to be logged:

```
NCP> SET KNOWN LOGGING EVENTS 2.*
```

- Except in the case of events, only component names can contain wildcards. Parameter values cannot contain wildcards. The following command is invalid because the circuit name UNA-* is not the component name in the command. Rather, it is a parameter used to modify the component named BOSTON. Only component names can be represented by wildcard characters.

```
NCP> SET NODE BOSTON SERVICE CIRCUIT UNA-*
    !INVALID COMMAND
```

  The component name EVENT is used as a parameter to the LOGGING commands and can contain wildcard characters, as long as only the *type* portion of the event number (the numeral to the right of the period) contains the wildcard.

  For example, the following command clears logging to the logging file for all class 2 events:

```
NCP> CLEAR LOGGING FILE EVENTS 2.*
```

- Unit numbers of circuit and line devices can contain wildcard characters, but device names of circuits and lines cannot contain wildcard characters. Circuit and line devices are typically identified by the format *dev-c*, where *dev* is a mnemonic device name, and *c* is a device unit number. In the following example, the asterisk replaces the unit number in this request for circuit information for all DMC devices:

```
NCP> SHOW CIRCUIT DMC-*
```

  However, the *device-name* portion of a circuit or line name cannot contain wildcard characters. Therefore, the following commands are invalid:

```
NCP> SHOW CIRCUIT D* STATUS   !INVALID COMMAND
NCP> SHOW LINE %NA-0 SUMMARY !INVALID COMMAND
```

Note that substituting a wildcard character for an entire component name is equivalent to specifying the command component KNOWN. For example:

```
NCP> SHOW NODE * STATUS
```

This command is equivalent to the following command:

```
NCP> SHOW KNOWN NODES STATUS
```

1.3. Command Prompting

NCP prompts for selected components and parameters if you do not supply them when you enter the command. These components and parameters are those that you use most often. Each prompt is either a list of valid choices for that position in the command string or the name of a parameter and a description of the value required.

Note

Prompting does not occur when NCP receives input from a command file. Parameter names must be supplied in command files.

Each prompt consists of two parts: the component or parameter desired, and its format. The syntax is as follows:

```
component (format):
```

The component part indicates the name of either the component or parameter. The format part indicates the range of acceptable values. The format part presents command words in uppercase, value classes in lowercase as descriptive strings, and numeric ranges as two numbers separated by a hyphen. Commas indicate alternatives that are mutually exclusive.

Prompting does not guarantee that you are entering a valid command. Even if parameters are required to make the command meaningful, prompting does not demand a response when prompting for parameter values or names, as the following example indicates:

```
NCP> SET OBJECT NML     RET
Object number                 (0-255): RET
File name            (filename.type): RET
Privileges  (List of VMS privileges): RET
User ID             (1-16 characters): RET
Password             (1-8 characters): RET
Account             (1-16 characters): RET
Proxy access INCOMING, OUTGOING, BOTH, NONE:) RET
%NCP-F-NOPARM, no parameters
NCP>
```

1.3.1. Exiting the Prompting Sequence

You can exit a prompting sequence by pressing Ctrl/Z. When you do so, the NCP> prompt returns, and you can then enter another NCP command. To exit NCP completely, press Ctrl/Z at the NCP> prompt.

The following example shows how to interrupt a prompting sequence and exit NCP.

```
NCP> SET     RET

(CIRCUIT, EXECUTOR, KNOWN, LINE,
   LOGGING, MODULE, NODE, OBJECT): LINE
Line ID                      (dev-c-u.t): FZA-1
Number of receive buffers      (1-1024): CTRL/Z
Interrupt
```

```
$
```

If after responding to one or more prompts you want to exit the prompting sequence but retain whatever information you have provided in the database, you can respond to the next prompt with _DONE, as follows:

```
NCP> DEFINE OBJECT        RET
Object Name              (12 characters): TALOS
Object Number              (0-255): 101
File name               (filename): TALOS.EXE
Privileges (List of VMS privileges): _DONE
NCP>
```

In this example, NCP makes a new entry in the permanent database for object TALOS. Because _DONE is used in response to a prompt, NCP writes no other parameter information to the database for this object, and NCP exits the prompting sequence. Note that the NML object returns a success message whenever it modifies the permanent database successfully.

Using _DONE may not be successful, however, if you do not provide all of the necessary parameters for a given component. For example:

```
NCP> DEFINE OBJECT        RET
Object Name              (12 characters): RIGEL
Object Number              (0-255): RET
File name               (filename): RIGEL.EXE
Privileges (List of VMS privileges): _DONE
%NCP-I-NMLRSP, listener response - Parameter missing, number
Object = RIGEL
NCP>
```

This example is similar to the previous example except that the object number is omitted. Because each object must have an object number defined in the database, and because none currently exists for object RIGEL, NCP does not update the database with this information.

1.3.2. Prompting Examples

The following examples illustrate three types of prompting that NCP provides:

• Prompts to supply missing component

```
NCP> CLEAR       RET
(CIRCUIT, EXECUTOR, KNOWN, LINE, LOGGING, MODULE, NODE, OBJECT):
```

No component is specified for the CLEAR command. The prompt format is a list of valid components separated by commas. Note that the plural component KNOWN is included in this list.

• Prompts to supply parameter values

```
NCP> CLEAR LINE       RET
LINE ID (dev-c-u):
```

NCP prompts for the *line-id* for this component because it is not provided with the command.

The following example shows variations of the prompting scheme for parameter values:

```
NCP> SET EXECUTOR       RET
Node address              (1.1-63.1023): 2.11
State        (ON, OFF, SHUT, RESTRICTED): ON
```

```
System id string (32 characters:) "DECnet for OpenVMS"
.
.
```

* Prompts to select parameter names

```
NCP> CLEAR LINE     RET
Line ID         (dev-c-u.t): FZA-1
All line parameters (Y,N):
```

NCP prompts for the selected parameter names whose values are to be removed from the database. In this case, no parameter name is given; consequently, NCP asks if all values should be removed. The Y parameter indicates that all values should be removed. Using N or RETURN causes NCP to provide a different response.

```
NCP> CLEAR NODE BOSTON     RET
All node parameters (Y,N): RET
Host node           (Y,N): Y
.
.
.
Service password    (Y,N): Y
Tertiary loader     (Y,N): N
```

Because RETURN is the response to the first prompt, NCP prompts for all of the parameter names.

1.4. Component and Parameter Syntax Rules

When you enter NCP commands, many components, parameters, and qualifiers require you to supply additional information. For the most part, their syntax follows a standard set of rules. Exceptions to these rules are documented in the description of the component, parameter, or qualifier to which they apply.

The syntax of the various component-name, parameter, and qualifier values is summarized in Table 1.2. In the following table, all numeric values are in decimal and have a range of 0 to 65,535 unless otherwise specified:

Table 1.2. NCP Syntax Summary

account	A string of up to 39 alphanumeric and hyphen characters. OpenVMS operating systems do not currently use this string for access control.
area-number	A decimal value in the range 1 to 63 to be specified in the beginning of the *node-address* and separated from the *node number* by a period. If you do not specify an area number, the area number of the executor is used. The default area number for the executor is 1.
circuit-id	A string of characters whose exact syntax is that for a DECnet for OpenVMS circuit. The string may also be that for an X.25 circuit identification. For X.25 circuits, circuit-id is a string of up to 15 characters beginning with "X25-". Circuit identification takes one of the following formats: dev-c dev-c-u dev-c.t dev-c-u.t

	dev	Is a device name. Refer to the description of *device-type* for a list of device mnemonic names. For a complete list of possible DECnet for OpenVMS circuit devices and descriptions, see Appendix A.
	c	Is a decimal number (0 or a positive integer) designating the device's hardware controller.
	u	Is a decimal unit or circuit number (0 or a positive integer) included only if more than one unit is associated with the controller.
	t	Is a decimal number (0 or a positive integer) that identifies a tributary on a multipoint circuit. This is a logical tributary number, not to be confused with the tributary address used to poll the tributary.
cpu-type	A string of characters consisting of one of the following: DECSYSTEM1020 PDP11 PDP8 VAX	
dest-name	An identification string consisting of 1 to 16 alphanumeric characters.	
device-type	A string of characters representing the mnemonic name for the device. Appendix A in this book contains a list of devices, current as of this publication date. The following list represents typical devices: 	BNA	DMF	TT
CI	DMP	TX
DMB	QNA	UNA
DMC	SVA			
dte-address	A string of up to 15 decimal digits. See the *Public Network Information Guide* for the exact format of this parameter on your network.	
E-address	A string of 12 hexadecimal digits, represented by 6 bytes separated by hyphens (for example, AA–00–04–00–AB–04). The string indicates an Ethernet address. The bytes are ordered from left to right as transmitted and received on the Ethernet.	
event-list	A list of event types for a given class in the format class.type. When specifying an event list, you may specify only one class; however, you can specify a range of types by using commas and hyphens, for example, 4.3–5,7–10. You can use the asterisk wildcard character in an event list, but only to replace an event type. The following table provides examples of these formats. | **Event List**	**Meaning**
4.4	Identifies event class 4, type 4.
4.5–7	Identifies event class 4, types 5 through 7.
4.5,7–9,11	Identifies event class 4, types 5, 7 through 9, and 11. Note that types must be specified in ascending order.
4.*	Identifies all event types for class 4 events.		
filespec	A VMS file specification string in the following general format:	

	`node-spec::device:[directory]filename.type;version` Logical names are permitted. Refer to the *VSI OpenVMS DCL Concepts Manual* for the format of a VMS file specification, and the *VSI OpenVMS Guide to OpenVMS File Applications* for more information about specifying files.
h-address	A string of 6 pairs of hexadecimal digits separated by hyphens (for example 08–00–2B–2B–AF–F3). The string represents the hardware address of a LAN device controller.
hex-byte	A string of exactly 2 hexadecimal digits.
hex-password	A string of up to 8 hexadecimal digits.
hex-value	A string of up to 32 hexadecimal digits.
id-string	A string of up to 32 characters. If the string includes spaces or tabs, enclose it within quotation marks.
line-id	A string of characters whose exact syntax is that for a DECnet for OpenVMS line identification. The string may also be that for an X.25 line identification. For OpenVMS operating systems, line-id takes one of the following formats: dev-c dev-c-u

	dev	Is a device name. Refer to the description of *device-type* for a list of device mnemonic names. For a complete list of possible DECnet for OpenVMS line devices and descriptions, see Appendix A.
	c	Is a decimal number (0 or a positive integer) designating the device's hardware controller.
	u	Is a decimal unit or line number (0 or a positive integer) included only if more than one unit is associated with the controller.

microseconds	A decimal numeric value.
milliseconds	A decimal numeric value.
network-name	For X.25 circuits, specifies a 1- to 16-character alphanumeric identifier that indicates the network to be used.
node-address	A numeric value in the range 1.1 to 63.1023, composed of an area number to the left of the period followed by a node number to the right of the period. (The node number indicates the address of the node within the specified area.) If the area number is not supplied, the area number of the executor node is used. The default area number for the executor is 1.
node-id	Either a *node-name* or a *node-address*.
node-name	A string of up to six alphanumeric characters containing at least one alphabetic character.
node-spec	A node-id followed by optional access control information in the following format: node-id"user password" node-id"user password account"
node-type	A string of characters consisting of one of the following:

	Nonrouting III Nonrouting IV Routing IV Routing III[1] Area[1]
number	A decimal numeric value.
object-name	A string of up to 12 printable characters, consisting of alphanumeric characters, the dollar sign ($), or the underscore (_).
p-address	A string of 6 pairs of hexadecimal digits separated by hyphens (for example AA–00–04–00–03–10). The value is the unique physical address that the node has set for itself or, if the node has not set a physical address, the hardware address associated with the node's device controller.
password	A string of up to 39 printable characters.
privilege-list	A list of VMS privilege names delimited by space characters.
seconds	A decimal numeric value.
sink-name	A string of alphanumeric characters that specifies the name of an event sink. The logging console name is a string in the form TTcnn (if a terminal) or a file specification if it is a file that receives events in the console format. The logging file name is a file specification. The logging monitor name is a 1- to 9-character name. If the string includes spaces or tabs, enclose it in quotation marks.
software-id	A string of up to 16 characters.
software-type	A string of characters consisting of one of the following: MANAGEMENT FILE SECONDARY LOADER SYSTEM TERTIARY LOADER
tributary-address[1]	A numeric value in the range of 0 to 255.
user-id	A string of up to 39 alphanumeric and hyphen characters. The operating system may include additional restrictions; for example, OpenVMS systems limit the string to 12 characters.

[1]VAX only

Quotation mark delimiters are valid for the *node-spec* format. In addition, you can use quotation marks as delimiters when providing receive and transmit passwords for the SET NODE and DEFINE NODE commands. For example:

```
NCP> SET NODE TRANSMIT PASSWORD "HI OpenVMS"
```

Also, use quotation marks to define a software identification string with text and spaces for the IDENTIFICATION parameter of the SET EXECUTOR command. For example:

```
NCP> SET EXECUTOR IDENTIFICATION "VMS HOST SYSTEM"
```

1.5. NCP HELP Facility

NCP provides an extensive HELP Facility that contains information about each NCP command and its parameters and qualifiers, including general examples of its use. The information in the HELP file is tree structured, which makes it easy to retrieve information quickly and efficiently. The HELP command

description in the Command Section provides examples to illustrate the use of the HELP command for a typical terminal session.

Use the HELP command to display information at various levels of detail. For example, each HELP display includes a list of command words under an "Additional Information Available" section. These command words serve as topics for the next level of information accessible by the HELP command. In the general format of the HELP command shown in the following example, the topic is a command word from the list:

```
NCP> HELP
Information available:
CLEAR   Commands  CONNECT  COPY  DEFINE   DISCONNECT  EVENTS
EXIT    HELP      LIST     LOAD  LOOP     PARAMETERS  Prompting
PURGE   SET       SHOW     TELL  TRIGGER  ZERO
Topic?
```

You can abbreviate command words when you enter the command. When you enter a particular level of detail, that level becomes the current level for accessing information. Through the use of the prompts *Topic?* and *Subtopic?*, you can return to a higher or lower level of information, respectively. The highest level of information is the general description of the HELP Facility.

Chapter 2. Using NCP

The Network Control Program (NCP) is a DECnet for OpenVMS utility that accepts commands to configure, control, monitor, and test a DECnet network.

Entering Commands

To invoke NCP, enter the following DCL command:

```
$ RUN SYS$SYSTEM:NCP
```

NCP returns the following prompt:

```
NCP>
```

Alternatively, you can enter a single NCP command from DCL by first entering a DCL string assignment statement. For example:

```
$ NCP=="$NCP"
$ NCP SHOW EXECUTOR STATUS
Node Volatile Status as of 12-JUL-1992 11:57:59

Executor node      = 4.306 (INTER)
State              = on
Physical address   = AA-00-04-00-32-11
Active links       = 2
Delay              = 0
$
```

NCP executes the SHOW EXECUTOR STATUS command and returns control to DCL.

To exit from an NCP session, type EXIT or press CTRL/Z after the NCP> prompt.

Output for the SHOW and LIST commands is normally displayed on the default output device, SYS$OUTPUT. Alternatively, you may direct output to a specified file using the TO qualifier with the SHOW or LIST command.

Certain NCP commands require privileges, as shown in Table 2.1. For DECnet purposes, an account is considered privileged if any privileges other than NETMBX and TMPMBX are held.

Table 2.1. Privileges for NCP Operations

Operation Required	Privileges
Start the network	ACNT, CMKRNL, SYSNAM, and DETACH
Perform task-to-task communication	NETMBX is required to assign a channel to the NET device. TMPMBX is required to optionally associate a temporary mailbox with a network channel.
Create a logical link	NETMBX
Declare a name or object number in a user task	SYSNAM (Refer to *VSI OpenVMS DECnet Networking Manual*for information about user tasks.)
CLEAR parameters from the volatile database	NETMBX and OPER
Issue CONNECT commands	NETMBX

Operation Required	Privileges
COPY KNOWN NODES	NETMBX, TMPMBX and OPER (SYSPRV is also needed if access to the permanent node database is required or if the WITH option is used.)
DEFINE parameters in the permanent database	SYSPRV and OPER[1]
DISCONNECT LINKs	NETMBX and OPER
LIST parameters kept in the permanent database	SYSPRV[1]
LIST service passwords kept in the permanent database	SYSPRV and OPER[1]
LIST receive passwords, transmit passwords, and object and executor access control passwords kept in the permanent database	BYPASS
Issue LOAD commands	NETMBX, TMPMBX and OPER
Use LOOP CIRCUIT	NETMBX, TMPMBX and OPER
Use LOOP EXECUTOR	NETMBX and TMPMBX
PURGE parameters from the permanent database	SYSPRV and OPER[1]
SET parameters kept in the volatile database	NETMBX and OPER
Issue SET EXECUTOR NODE	Requires NETMBX and TMPMBX privileges on the local node, and NML on the executor node must have appropriate privileges to perform the commands issued after the SET EXECUTOR NODE command.
Use "SET component ALL" to load parameters from the permanent database to the volatile database	NETMBX, OPER and SYSPRV[1]
SHOW parameters kept in the volatile database	NETMBX
SHOW service passwords kept in the volatile database	NETMBX and OPER
SHOW receive passwords, transmit passwords, and access control passwords kept in the volatile database	NETMBX and BYPASS
Issue TELL commands	Requires NETMBX and TMPMBX on the local node, and NML on the executor node must have sufficient privileges to perform the command that follows TELL.
Issue TRIGGER commands	NETMBX, TMPMBX and OPER
ZERO counters	NETMBX and OPER

[1]You can perform operations requiring access to the permanent database without the SYSPRV privilege if you have read and write access to all configuration database files or hold BYPASS privilege. However, VSI recommends that you protect these network configuration files from unauthorized access by requiring SYSPRV to access these files.

2.1. Hints for Using Commands

You can abbreviate any command verb, component, parameter, or qualifier as long as the abbreviation is not ambiguous.

Commands that have the same components and parameters, yet different command verbs—depending upon whether they access the volatile or the permanent database—are listed together in Chapter 1. Examples of commands listed together are CLEAR and PURGE, SET and DEFINE, and SHOW and LIST, where CLEAR, SET, and SHOW verbs apply to the volatile database and PURGE, DEFINE and LIST verbs apply to the permanent database. When two commands are grouped together, components and parameters are described for the command that accesses the volatile database. Typically, the actions described for the volatile database also apply to the permanent database. However, if certain commands apply only to a particular database or if a different action is required for the volatile and permanent databases, these exceptions are specifically noted in the command description.

In general, the NCP commands and parameters are common to most network configurations. Exceptions to this general rule are noted under the command or parameter to which they apply. The VAX margin icon indicates functions supported only on VAX systems. Other restrictions are noted as part of the command or parameter description.

___

Note

Although NCP may accept a particular command syntax, this is no guarantee that some other DECnet implementation supports the function.

___

For a complete description of the network components specified in NCP commands, see the *VSI OpenVMS DECnet Networking Manual*. That manual also explains how to use the NCP commands to manage the DECnet network.

Chapter 3. NCP Commands

This chapter presents NCP command descriptions. Commands are presented in alphabetical order with the associated command parameters and appropriate examples.

NCP allows for management of the local node and for remote management of any DECnet node that you specify as the executor. When you issue NCP commands, there are two ways to specify the executor — the location at which the command executes:

- Begin individual NCP commands with the TELL prefix.

- Set the executor with the SET EXECUTOR NODE command. All subsequent NCP commands execute at the remote node until you:

  - Issue another SET EXECUTOR NODE command,

  - Issue a CLEAR EXECUTOR NODE command, or

  - Exit NCP.

To enter one or two NCP commands for remote execution, it is probably easier to use the TELL prefix. To execute multiple NCP commands at a remote node, first issue the SET EXECUTOR NODE command, specifying that node.

Note

NCP commands and command parameters are not all supported on every hardware platform. In some cases, NCP allows you to set and display an unsupported parameter. However, the unsupported parameter is not functional and has no effect on the operation of the DECnet software.

CLEAR/PURGE CIRCUIT

CLEAR/PURGE CIRCUIT — The CLEAR CIRCUIT command resets circuit parameters in the volatile database to the default value (if any), or removes them from the volatile database. The PURGE CIRCUIT command removes circuit parameters from the permanent database.

Format

CLEAR   [circuit-component parameter [...]]

PURGE   [circuit-component parameter [...]]

Circuit Components

CIRCUIT
KNOWN CIRCUITS

Circuit Components

CIRCUIT circuit-id

Resets to the default value (if any) or removes from the database the parameters for a specific circuit. The *circuit-id* specifies the name of the circuit you want to clear.

KNOWN CIRCUITS

Resets to the default value (if any) or removes from the database the parameters for all known circuits.

Command Parameters

ACTIVE BASE

Applies only to DDCMP CONTROL circuits. Resets to its default value the ACTIVE BASE value in the database.

ACTIVE INCREMENT

Applies only to DDCMP CONTROL circuits. Resets to its default value the ACTIVE INCREMENT value in the database.

ALL

Removes from the database the specified circuit or all known circuits. The circuit must be in the OFF state before you specify this parameter.

BABBLE TIMER

Applies only to DDCMP CONTROL circuits. Resets to its default value the BABBLE TIMER value in the database.

COUNTER TIMER

Cancels the logging timer. This cancellation prevents any further circuit counter logging for the indicated circuit or circuits.

DEAD THRESHOLD

Applies only to DDCMP CONTROL circuits. Resets to its default value the DEAD THRESHOLD count in the database.

DTE

Applies only to DLM circuits. Allows the X.25 call to be set up using any local DTE.

DYING BASE

Applies only to DDCMP CONTROL circuits. Resets to its default value the DYING BASE value in the database.

DYING INCREMENT

Applies only to DDCMP CONTROL circuits. Resets to its default value the DYING INCREMENT value in the database.

DYING THRESHOLD

Applies only to DDCMP CONTROL circuits. Resets to its default value the DYING THRESHOLD count in the database.

INACTIVE BASE

Applies only to DDCMP CONTROL circuits. Resets to its default value the INACTIVE BASE value in the database.

INACTIVE INCREMENT

Applies only to DDCMP CONTROL circuits. Resets to its default value the INACTIVE INCREMENT value in the database.

INACTIVE THRESHOLD

Applies only to DDCMP CONTROL circuits. Resets to its default value the INACTIVE THRESHOLD value in the database.

MAXIMUM BUFFERS

Applies only to DDCMP CONTROL circuits. Resets to its default value the MAXIMUM BUFFERS value in the database.

MAXIMUM RECALLS

Applies only to X.25 DLM circuits. Resets to its default value the limit on the number of call retries.

MAXIMUM ROUTERS

Applies only to Ethernet circuits. Resets to its default value the MAXIMUM ROUTER value in the database.

MAXIMUM TRANSMITS

Applies only to DDCMP CONTROL circuits. Resets to its default value the MAXIMUM TRANSMIT count in the database.

NETWORK

Applies only to purging X.25 PVCs and X.25 DLM circuits. Removes the circuit's network parameters from the permanent database.

NUMBER

Applies only to incoming DLM circuits. Removes the restriction on accepting calls only from a particular remote DTE number.

RECALL TIMER

Applies only to X.25 DLM circuits. Resets to its default value the RECALL TIMER value in the database.

ROUTER PRIORITY

Applies only to Ethernet circuits. Resets to its default value the priority assigned to the router in the database.

TRANSMIT TIMER

Applies only to DDCMP CONTROL circuits. Resets to its default value the TRANSMIT TIMER value in the database.

Example

1. `NCP> CLEAR CIRCUIT DMC-0 ALL`

   This command removes all parameter entries for circuit DMC–0 in the volatile database. As a result, the circuit no longer exists for the local DECnet software.

2. `NCP> CLEAR CIRCUIT DMP-0.4 COUNTER TIMER`

   This command removes the COUNTER TIMER parameter for circuit DMP–0.4 from the volatile database.

3. `NCP> PURGE CIRCUIT X25-ANDIES ALL`

   This command removes all parameter entries in the permanent database for the X.25 permanent virtual circuit (PVC) called X25-ANDIES.

4. `NCP> CLEAR CIRCUIT UNA-* ALL`

   This command clears all UNA circuits and the associated parameters for all UNA circuits in the volatile database.

CLEAR/PURGE EXECUTOR

CLEAR/PURGE EXECUTOR — The CLEAR EXECUTOR command resets selected local node parameters to the default value (if any) or removes them from the volatile database on the local node. The PURGE EXECUTOR command removes selected local node parameters from the permanent database on the local node. You cannot clear the BUFFER SIZE or STATE parameter from the volatile database.

Format

CLEAR EXECUTOR parameter [...]

PURGE EXECUTOR parameter [...]

Command Parameters

ALIAS INCOMING

Removes from the database the ALIAS INCOMING parameter for the local node.

ALIAS MAXIMUM LINKS

Resets to its default value the ALIAS MAXIMUM LINKS value for the local node in the database.

ALIAS NODE

Removes from the database the alias node identifier for the local node.

ALL

Removes from the permanent database all executor parameters.

AREA MAXIMUM COST

Resets to its default value the AREA MAXIMUM COST value for the local node in the database.

AREA MAXIMUM HOPS

Resets to its default value the AREA MAXIMUM HOPS value for the local node in the database.

BROADCAST ROUTING TIMER

Resets to its default value the BROADCAST ROUTING TIMER value in the database.

BUFFER SIZE

Removes from the permanent database the BUFFER SIZE value.

COUNTER TIMER

Removes from the database the logging timer to prevent any further circuit counter logging for the local node.

DELAY FACTOR

Resets to its default value the DELAY FACTOR value for the local node in the database.

DELAY WEIGHT

Resets to its default value the DELAY WEIGHT value for the local node in the database.

IDENTIFICATION

Removes from the database the identification string for the local node.

INACTIVITY TIMER

Resets to its default value the INACTIVITY TIMER value for the local node in the database.

INCOMING PROXY

Resets to its default value the INCOMING PROXY value for the local node in the database.

INCOMING TIMER

Resets to its default value the INCOMING TIMER value for the local node in the database.

MAXIMUM ADDRESS

Resets to its default value the MAXIMUM ADDRESS value for the local node in the database.

MAXIMUM AREA

Resets to its default value the MAXIMUM AREA value for the local node in the database.

MAXIMUM BROADCAST NONROUTERS

Resets to its default value the MAXIMUM BROADCAST NONROUTERS value in the database.

MAXIMUM BROADCAST ROUTERS

Resets to its default value the MAXIMUM BROADCAST ROUTERS value in the database.

MAXIMUM BUFFERS

Resets to its default value the MAXIMUM BUFFERS value for the local node in the database.

MAXIMUM CIRCUITS

Resets to its default value the MAXIMUM CIRCUITS value for the local node in the database.

MAXIMUM COST

Resets to its default value the MAXIMUM COST value for the local node in the database.

MAXIMUM HOPS

Resets to its default value the MAXIMUM HOPS value for the local node in the database.

MAXIMUM LINKS

Resets to its default value the MAXIMUM LINKS value for the local node in the database.

MAXIMUM PATH SPLITS

Resets to its default value the MAXIMUM PATH SPLITS value for the local node in the database.

MAXIMUM VISITS

Resets to its default value the MAXIMUM VISITS value for the local node in the database.

NONPRIVILEGED item

Removes from the database the specified nonprivileged access control information. Specify any or all of the following items for removal:

ACCOUNT
PASSWORD
USER

OUTGOING PROXY

Resets to its default value the OUTGOING PROXY value for the local node in the database.

OUTGOING TIMER

Resets to its default value the OUTGOING TIMER value for the local node in the database.

PATH SPLIT POLICY

Resets to its default value the PATH SPLIT POLICY value for the local node in the database.

PIPELINE QUOTA

Resets to its default value the PIPELINE QUOTA value for the local node in the database.

PRIVILEGED item

Removes from the database the specified privileged access control information. Specify any or all of the following items for removal:

ACCOUNT
PASSWORD
USER

RETRANSMIT FACTOR

Resets to its default value the RETRANSMIT FACTOR value for the local node in the database.

ROUTING TIMER

Resets to its default value the ROUTING TIMER value for the local node in the database.

SEGMENT BUFFER SIZE

Resets to its default value the SEGMENT BUFFER SIZE value in the database.

STATE

Removes from the permanent database the local node's state.

SUBADDRESSES

Removes from the database the value of the range of local DTE subaddresses.

TYPE

Resets to its default value the type of the local node in the database.

Example

1. `NCP> PURGE EXECUTOR IDENTIFICATION INCOMING TIMER`

   This command removes the local node's identification string and incoming timer value from the permanent database.

2. `NCP> PURGE EXECUTOR COUNTER TIMER`

   This command removes the local node counter timer value from the permanent database.

3. `NCP> CLEAR EXECUTOR AREA MAXIMUM COST`

   This command resets to its default value the AREA MAXIMUM COST parameter in the volatile database for the local node whose type is AREA. AREA MAXIMUM COST is the total cost allowed from the local node to any other level 2 router.

CLEAR EXECUTOR NODE

CLEAR EXECUTOR NODE — The CLEAR EXECUTOR NODE command clears the default executor designation for all NCP commands. The executor of commands becomes the local node. You

cannot use the TELL prefix with the CLEAR EXECUTOR NODE command. After you enter the CLEAR EXECUTOR NODE command, NML, linked with NCP on the local node, performs all NML operations. NML uses the privileges of your current process instead of the default nonprivileged or privileged accounts.

Format

```
CLEAR EXECUTOR NODE
```

Parameters

None.

Example

```
NCP> SET EXECUTOR NODE TRNTO"GRAY MARY"...
NCP> CLEAR EXECUTOR NODE
```

The first command sets the executor node to node TRNTO. The second command clears the default executor node designation specified previously by the SETEXECUTOR NODE command.

CLEAR/PURGE LINE

CLEAR/PURGE LINE — The CLEAR LINE command resets line parameters to the default value (if any) or removes the entries and parameters from the volatile database on the local node or DTE. For an X.25 line, the default value depends on the profile that is associated with the NETWORK parameter. The PURGE LINE command removes line parameters from the permanent database on the local node or DTE.

Format

CLEAR   [line-component parameter [...]]

PURGE   [line-component parameter [...]]

Line Components

LINE
KNOWN LINES

Line Components

LINE line-id

Identifies the specific line whose parameters are to be reset to their default values (if any) or removed from the database.

KNOWN LINES

Resets the parameters to the default value (if any) or removes the m for all known lines from the database.

Command Parameters

ALL

Removes the specified line or all known lines from the database. The line must be in the OFF state before you specify this parameter.

COUNTER TIMER

Resets the logging timer to zero. This prevents any further line counter logging for the indicated line(s).

DEAD TIMER

Applies only to DDCMP lines. Resets the DEADTIMER parameter to its default value in the database.

DELAY TIMER

Applies only to DDCMP lines. Removes the DELAY TIMER parameter from the database.

HANGUP

Applies only to asynchronous DDCMP lines. Removes the HANGUP parameter from the database.

HOLDBACK TIMER

Applies only to X.25 lines. Removes the HOLDBACK TIMER parameter from the database, causing the X.25 software to acknowledge received data immediately.

LINE SPEED

Applies only to asynchronous DDCMP lines. Removes the LINE SPEED parameter from the database.

MAXIMUM RETRANSMITS

Applies only to X.25 lines. Resets the maximum number of retransmissions of a frame to the default value for the specified line or all known lines in the database.

RECEIVE BUFFERS

Applies only to DDCMP lines. Resets the RECEIVE BUFFERS parameter to its default value in the database.

SCHEDULING TIMER

Applies only to DDCMP lines. Resets the SCHEDULING TIMER parameter to its default value in the database.

STREAM TIMER

Applies only to DDCMP lines. Resets the STREAMTIMER parameter to its default value in the database.

SWITCH

Applies only to asynchronous DDCMP lines. Removes the SWITCH parameter from the database.

TRANSMIT PIPELINE

Applies only to DMR11 lines. Removes the TRANSMIT PIPELINE parameter from the database.

Example

1. `NCP> CLEAR LINE DMC-0 ALL`

   This command removes line DMC–0 from the volatile database.

2. `NCP> CLEAR LINE DUP-0 ALL`

   This command removes the line called DUP–0 from the volatile database.

3. `NCP> CLEAR KNOWN LINES MAXIMUM RETRANSMITS`

   This command resets the maximum number of retransmissions of a frame for all known lines in the volatile database. The parameter is reset to its default value.

4. `NCP> CLEAR LINE DMC-* ALL`

   This command removes all DMC lines from the volatile database.

CLEAR/PURGE LOGGING

CLEAR/PURGE LOGGING — The CLEAR LOGGING command removes logging parameters from the volatile database on the local node or DTE. The PURGE LOGGING command removes logging parameters from the permanent database on the local node or DTE.

Format

CLEAR   [logging-component parameter]

PURGE   [logging-component parameter]

Logging Components

KNOWN LOGGING
LOGGING FILE
LOGGING CONSOLE
LOGGING MONITOR

Logging Components

KNOWN LOGGING

Removes from the database the parameters for all known logging.

LOGGING CONSOLE

Removes from the database all parameters for logging to the console.

LOGGING FILE

Removes from the database all parameters for logging to a file.

LOGGING MONITOR

Removes from the database all parameters for logging to a monitor.

Command Parameters

ALL

Removes all parameters from the database. After you enter the CLEAR/PURGE LOGGING command with this parameter, the logging component is no longer recognized by the network.

CIRCUIT circuit-id

Removes from the database all logging parameters for the specified circuit.

LINE line-id

Removes from the database all logging parameters for the specified line.

MODULE name

Removes from the database all logging parameters for the specified module. Note that the word MODULE is optional. Module names include the following:

X25-PROTOCOL
X25-SERVER
X29-SERVER
X25-ACCESS

NODE node-id

Removes from the database all logging parameters for the specified node.

Example

```
NCP> CLEAR LOGGING CONSOLE ALL
```

This command clears logging to the logging console, after which the logging component is no longer recognized by the network.

CLEAR/PURGE LOGGING EVENTS

CLEAR/PURGE LOGGING EVENTS — The CLEAR LOGGING EVENTS command removes source-related logging parameters from the volatile database on the local node or DTE. The PURGE LOGGING EVENTS command removes source-related logging parameters from the permanent database on the local node or DTE. If no source is specified in the command, the source specific events will not be affected.

Format

CLEAR   [logging-component parameter [...]]

PURGE   [logging-component parameter [...]]

Logging Components

KNOWN LOGGING
LOGGING FILE
LOGGING CONSOLE
LOGGING MONITOR

Logging Components

KNOWN LOGGING

Removes from the database the parameters for all known logging.

LOGGING CONSOLE

Removes from the database the parameters for logging to the console.

LOGGING FILE

Removes from the database the parameters for logging to a file.

LOGGING MONITOR

Removes from the database the parameters for logging to a monitor program.

Command Parameters

EVENTS event-list

Removes from the database an event class and type or all event classes and types. You can associate any of the following source parameters when you specify the EVENTS parameter.

CIRCUIT circuit-id	Identifies the circuit to which the events apply. Events will not be logged for this circuit.
LINE line-id	Identifies the line to which the events apply. Events will not be logged for this source.
MODULE name	Identifies the module to which the events apply. Events will not be logged for this source. Module names include the following: X25-PROTOCOL X25-SERVER X29-SERVER X25-ACCESS Note that the word MODULE is optional.
NODE node-id	Identifies the node to which the events apply. Events will not be logged for this source.
SINK	Identifies the node where the event logger runs to log the events. There are two possibilities:
	EXECUTOR	Identifies the local node

	NODE node-id	Identifies the node at which events are being logged
	Only events being logged to this node are removed from the database. If this parameter is omitted, the local node is assumed to be the sink node affected.	

Example

1. `NCP> CLEAR LOGGING FILE EVENTS 2.*`

   This command clears logging to the logging file for all class 2 events.

2. `NCP> CLEAR LOGGING CONSOLE KNOWN EVENTS LINE DMC-1`

   This command clears logging to the console for all known events currently being logged for line DMC–1.

CLEAR/PURGE LOGGING NAME

CLEAR/PURGE LOGGING NAME — The CLEAR LOGGING NAME command removes the name of the logging sink from the volatile database on the local node or DTE. The PURGE LOGGING NAME command removes the name of the logging sink from the permanent database on the local node or DTE. You must enter a SET LOGGING *component* STATE OFF command before attempting to use the CLEAR command.

Format

CLEAR   [logging-component NAME]

PURGE   [logging-component NAME]

Logging Components

KNOWN LOGGING
LOGGING FILE
LOGGING CONSOLE
LOGGING MONITOR

Logging Components

KNOWN LOGGING

Indicates that the name of the logging sink for all known logging is to be removed from the database.

LOGGING CONSOLE

Indicates that the name of the logging console is to be removed from the database.

LOGGING FILE

Indicates that the name of the logging file is to be removed from the database.

LOGGING MONITOR

Indicates that the name of the logging monitor is to be removed from the database.

Example

```
NCP> CLEAR LOGGING CONSOLE NAME
```

This command clears the name of the logging console from the volatile database.

CLEAR/PURGE MODULE X25-ACCESS

CLEAR/PURGE MODULE X25-ACCESS — The CLEAR MODULE X25-ACCESS command removes from the volatile database network names and parameters used by the host node to access a specified X.25 network or networks through a connector node. The PURGE MODULEX25-ACCESS command removes these same network names and parameters from the permanent database. Note that the word MODULE is optional.

Format

CLEAR MODULE X25-ACCESS   [qualifier parameter [...]]

PURGE MODULE X25-ACCESS   [qualifier parameter [...]]

Qualifiers

NETWORK
KNOWN NETWORKS

Qualifiers

NETWORK network-name
KNOWN NETWORKS

> The name of a specific network or all of the network names to be removed from the database. After you specify the NETWORK qualifier,you should follow it with one or more of these parameters:

ACCOUNT	Removes user account access control information from the database.
ALL	Removes all X25-ACCESS module parameters from the database.
PASSWORD	Removes the password access control information from the database.
USER	Removes the user identification from the access control information in the database.

Example

1. `NCP> CLEAR MODULE X25-ACCESS NETWORK TELENET1 ALL`

   This command removes the network TELENET1 from the volatile database in the local node. The local node can no longer connect to X.25 network TELENET1 through the connector node.

2. `NCP> CLEAR MODULE X25-ACCESS NETWORK TELENET1 PASSWORD`

This command removes the password access control information for the network TELENET1 from the volatile database in the local node.

CLEAR/PURGE MODULE X25-PROTOCOL

CLEAR/PURGE MODULE X25-PROTOCOL — The CLEAR MODULE X25-PROTOCOL command removes the protocol module component from the volatile database or resets individual parameters. (Parameters are reset to their default values; the network values are specified by the PROFILE parameter of the SET MODULE X25-PROTOCOL command.) The PURGE MODULE X25-PROTOCOL command removes the parameters of the protocol module component from the permanent database. Note that the word MODULE is optional. Use separate CLEAR or PURGE MODULE X25-PROTOCOL commands to clear or purge the network, the DTE, or the group. You may not be able to change the values of some of the network parameters (see the *Public Network Information Guide*).

Format

CLEAR MODULE X25-PROTOCOL   [[qualifier] parameter [...]]

PURGE MODULE X25-PROTOCOL   [[qualifier] parameter [...]]

Qualifiers

DTE
KNOWN DTES
GROUP
KNOWN GROUPS
NETWORK
KNOWN NETWORKS

Qualifiers

DTE dte-address
KNOWN DTES

Removes parameters for the specified DTE or all known DTEs in the database. If you specify the DTE qualifier, you must associate with it either of the following parameters.

NETWORK network-name	Selects the DTE(s) specified on a particular network. With the PURGE command, you can omit the network part of the qualifier if there is only one network defined.
KNOWN NETWORKS	Selects the DTE(s) on all known networks.

Additionally, you must follow the DTE qualifier with one or more of the following parameters:

ALL	Removes the specified DTE or all known DTEs from the database.
CALL TIMER	Resets the call timer to its default value in the database.
CLEAR TIMER	Resets the clear timer to its default value in the database.

COUNTER TIMER	Removes the counter timer for the specified DTE or all known DTEs from the database.
DEFAULT DATA	Resets the default packet size to its default value in the database.
DEFAULT WINDOW	Resets the default window size to its default value in the database.
INTERRUPT TIMER	Resets the interrupt timer to its default value in the database.
MAXIMUM CIRCUITS	Resets the maximum number of virtual circuits allowed at once on the DTE to its default value in the database.
MAXIMUM CLEARS	Resets to the default value in the database the maximum number of attempts to clear a circuit.
MAXIMUM DATA	Resets the maximum packet size to its default value in the database.
MAXIMUM RESETS	Resets to the default value in the database the maximum number of attempts to reset a circuit.
MAXIMUM RESTARTS	Resets to the default value in the database the maximum number of attempts to restart the DTE.
MAXIMUM WINDOW	Resets the maximum window size to its default value in the database.
RESET TIMER	Resets the reset timer to its default value in the database.
RESTART TIMER	Resets the restart timer to its default value in the database.

GROUP group-name
KNOWN GROUPS

Removes parameters for the specified closed user group (CUG) or bilateral closed user group (BCUG), or all known groups, in the database. If you specify a group qualifier, you must follow it with the following parameter:

| ALL | Removes all parameters for the specified group or all known groups. |

NETWORK network-name
KNOWN NETWORKS

Removes all DTEs, groups, and parameters for the specified network or all known networks in the database. You must specify the following parameter:

| ALL | Removes all parameters for the network or networks. |

Example

1. NCP> CLEAR MODULE X25-PROTOCOL GROUP ESECUG ALL

This command removes the closed user group ESECUG from the volatile database.

2. `NCP> PURGE MODULE X25-PROTOCOL GROUP ESECUG ALL`

This command removes the parameters of the closed user group ESECUG from the permanent database. When the system is rebooted, the group no longer exists for VAX P.S.I.

3. `NCP> CLEAR MODULE X25-PROTOCOL DTE 123789456 NETWORK PSS1 ALL`

This command removes the specified DTE belonging to the network PSS1 from the volatile database.

4. `NCP> PURGE MODULE X25-PROTOCOL DTE 123789456 KNOWN NETWORKS ALL`

This command removes all parameter entries for all the DTEs of that DTE number, for all networks in which the DTE exists, from the permanent database. When the network is restarted, the DTE no longer exists for VAX P.S.I.

5. `NCP> CLEAR MODULE X25-PROTOCOL ALL`

This command removes all network, DTE, group, and X25-PROTOCOL module parameters from the volatile database.

6. `NCP> CLEAR MODULE X25-PROTOCOL NETWORK PSS1 ALL`

This command removes all DTEs and groups on network PSS1 and then removes the network name itself.

7. `NCP> CLEAR MODULE X25-PROTOCOL KNOWN DTES -`
   `_ KNOWN NETWORKS COUNTER TIMER`

This command removes the COUNTER TIMER parameter for all DTEs on all networks from the volatile database.

CLEAR/PURGE MODULE X25-SERVER/X29-SERVER

CLEAR/PURGE MODULE X25-SERVER/X29-SERVER — The CLEAR MODULE X25-SERVER or CLEAR MODULE X29-SERVER command resets or removes the X.25 or X.29 call handler from the volatile database or removes individual parameters. The PURGE MODULE X25-SERVER or PURGE MODULEX29-SERVER command removes the parameters of the X.25 or X.29 call handler from the permanent database. Note that the word MODULE is optional.

Format

CLEAR MODULE X25-SERVER [[qualifier] parameter [...]]

CLEAR MODULE X29-SERVER [[qualifier] parameter [...]]

PURGE MODULE X25-SERVER [[qualifier] parameter [...]]

PURGE MODULE X29-SERVER [[qualifier] parameter [...]]

Qualifiers

DESTINATION
KNOWN DESTINATIONS

Qualifiers

DESTINATION dest-name
KNOWN DESTINATIONS

Identifies the destination or all known destinations whose parameters are to be reset or removed from the database. If you specify a destination qualifier, you must follow it with one or more of the following parameters.

ACCOUNT	Removes from the database the account for the specified destination or all known destinations.
ALL	Removes from the database all parameters for the specified destination or all known destinations.
CALL MASK	Removes from the database the call mask for the specified destination or all known destinations.
CALL VALUE	Removes from the database the call value for the specified destination or all known destinations.
CALLED ADDRESS	Removes from the database the called DTE address for the specified destination or all known destinations.
EXTENSION MASK	Removes from the database the extension mask for the specified destination or all known destinations.
EXTENSION VALUE	Removes from the database the extension value for the specified destination or all known destinations.
GROUP	Removes from the database the group name for the specified destination or all known destinations.
INCOMING ADDRESS	Removes from the database the incoming address for the specified destination or all known destinations.
NETWORK	Removes from the database the network name for the specified destination or all known destinations.
OBJECT	Removes from the database the object identification for the specified destination or all known destinations.
PASSWORD	Removes from the database the password for the specified destination or all known destinations.
PRIORITY	Resets to the lowest value the priority for the specified destination or all known destinations.

RECEIVING DTE	Removes from the database the receiving DTE address for the specified destination or all known destinations.
REDIRECT REASON	Removes from the database the redirect reason code for the specified destination or all known destinations.
SENDING ADDRESS	Removes from the database the sending DTE address for the specified destination or all known destinations.
SUBADDRESSES	Removes from the database the subaddress range for the specified destination or all known destinations.
USER	Removes from the database the user identification for the specified destination or all known destinations.

Command Parameter

COUNTER TIMER

Resets the module counter timer to 0. This is the only parameter with which you are not required to supply a destination qualifier.

Example

1. `NCP> CLEAR MODULE X25-SERVER KNOWN DESTINATIONS ALL`

   This command removes all X.25 destinations from the volatile database.

2. `NCP> PURGE MODULE X25-SERVER KNOWN DESTINATIONS ALL`

   This command removes all parameters for all X.25 destinations from the permanent database.

CLEAR/PURGE NODE

CLEAR/PURGE NODE — The CLEAR NODE command removes node parameters from the volatile database on the local node. The PURGE NODE command removes node parameters from the permanent database on the local node. To remove a loop node name from the database, use the CLEAR/ PURGENODE node-id command and specify either the ALL parameter (with the loop node name or address as the node-id), or the CIRCUIT parameter.

Format

CLEAR   [node-component parameter [...]]

PURGE   [node-component parameter [...]]

Node Components

NODE

KNOWN NODES

Node Components

NODE node-id

Identifies the node (local or remote) whose parameters are to be removed from the database.

KNOWN NODES

Removes from the database the specified parameters for all known nodes.

Command Parameters

ALL

Removes from the database all parameters for the specified node or all known nodes. The component is no longer recognized by the network.

Caution

The PURGE KNOWN NODES ALL command purges both the executor and remote node databases.

COUNTER TIMER

Removes the COUNTER TIMER value from the database.

CPU

Removes the node's CPU identification from the database.

DIAGNOSTIC FILE

Removes from the database the identification of the downline loading diagnostics file.

DUMP ADDRESS

Removes from the database the upline dump address of the adjacent node.

DUMP COUNT

Removes the upline dump count from the database.

DUMP FILE

Removes the upline dump file identification from the database.

HARDWARE ADDRESS

Removes the hardware Ethernet address from the database.

HOST

Removes from the database the identification of the host node.

INBOUND

Applies only to nodes connected by DDCMP point-to-point circuits. Removes the INBOUND parameter from the database.

LOAD ASSIST AGENT

Removes from the database the identification of the VMS image used to assist a downline load. The LOAD ASSIST AGENT parameter applies to downline load operations to Local Area VAX clusters only.

LOAD ASSIST PARAMETER

Removes from the database the value to be passed to the load assist agent. The LOAD ASSIST PARAMETER value applies to downline load operations to Local Area VAX clusters only.

LOAD FILE

Removes from the database the identification of the downline loading file.

MANAGEMENT FILE

Removes from the database the identification of the downline loading management information file.

NAME

Removes from the database the node name parameter for the node address.

NONPRIVILEGED item

Removes the specified nonprivileged access control information from the database. Specify any or all of the following items for removal:

ACCOUNT
PASSWORD
USER

PRIVILEGED item

Removes the specified privileged access control information from the database. Specify any or all of the following items for removal:

ACCOUNT
PASSWORD
USER

RECEIVE PASSWORD

Removes from the database the receive password expected from the remote node during a Routing initialization sequence.

SECONDARY LOADER

Removes from the database the identification of the secondary downline loading file.

SERVICE CIRCUIT

Removes from the database the circuit parameter associated with the node for downline loading purposes.

SERVICE DEVICE

Removes the service device type from the database.

SERVICE NODE VERSION

Removes from the database the DECnet–VAX software version number of the node to be downline loaded.

SERVICE PASSWORD

Removes from the database the password parameter required to trigger the bootstrap mechanism of the node to be downline loaded.

SOFTWARE IDENTIFICATION

Removes the identification of the software from the database.

SOFTWARE TYPE

Removes the initial load software program type from the database.

TERTIARY LOADER

Removes from the database the identification of the tertiary downline loading file.

TRANSMIT PASSWORD

Removes from the database the transmit password sent to the remote node during a Routing initialization sequence.

Example

1. `NCP> CLEAR NODE TRNTO ALL`

   This command removes all parameter entries for node TRNTO from the volatile database. As a result, the node no longer exists for local DECnet–VAX software.

2. `NCP> CLEAR NODE 14 NAME...`
   `NCP> SET NODE 15 NAME DENVER`

   The first command disassociates the name DENVER from node 14. The second command associates the same name with node 15.

3. `NCP> CLEAR NODE 5.14 NAME...`
   `NCP> SET NODE 5.15 NAME DENVER`

   The first command disassociates the name DENVER from node 5.14. The second command associates the same name with node 5.15. Use this example in place of the previous example when the network is divided into multiple areas.

4. `NCP> CLEAR NODE BOSTON RECEIVE PASSWORD`

This command removes the receive password from the record for the local node from the volatile database.

5. `NCP> CLEAR NODE 5.* NAME`

This command disassociates the node names from their addresses for all nodes in area 5.

CLEAR/PURGE NODE CIRCUIT

CLEAR/PURGE NODE CIRCUIT — The CLEAR NODE CIRCUIT command removes a loop node from the volatile database on the executor node. The PURGE NODE CIRCUIT command removes a loop node from the permanent database on the executor node.

Format

CLEAR   [node-component CIRCUIT]

PURGE   [node-component CIRCUIT]

Node Components

NODE
KNOWN NODES

Node Components

NODE node-id

Identifies the loop node to be removed from the database.

KNOWN NODES

Indicates that all loop nodes are to be removed from the database.

Example

```
NCP> SET NODE TESTER CIRCUIT DMC-0...
NCP> CLEAR NODE TESTER CIRCUIT
```

The first command sets a loop node for circuit DMC–0. The second command removes the loop node from the volatile database.

CLEAR/PURGE OBJECT

CLEAR/PURGE OBJECT — The CLEAR OBJECT command removes from the volatile database object parameters on the local node. The PURGEOBJECT command removes from the permanent database object parameters on the local node.

Format

CLEAR   [object-component parameter [...]]

PURGE [object-component parameter [...]]

Object Components

KNOWN OBJECTS
OBJECT

Object Components

KNOWN OBJECTS

Indicates that parameters for all known objects are to be removed from the database.

OBJECT object-name

Identifies the object whose parameters are to be removed from the database.

Command Parameters

ACCOUNT

Removes the account access control information from the database.

ALIAS INCOMING

Removes from the database the ALIAS INCOMING parameter for the specified object or all known objects. Does not apply to VAX P.S.I.

ALIAS OUTGOING

Removes from the database the ALIAS OUTGOING parameter for the specified object or all known objects. Does not apply to VAX P.S.I.

ALL

Removes from the database all parameters for the specified object or all known objects. After you enter this parameter,the component is no longer recognized by the network.

FILE

Removes the name of the object's startup command file from the database.

OUTGOING CONNECT PRIVILEGES

Removes from the database the privileges required to make an outbound connection to the object. Does not apply to VAX P.S.I.

PASSWORD

Removes the password access control information from the database.

PRIVILEGES

Removes from the database the privileges required to access the object. Does not apply to VAX P.S.I.

PROXY

Removes from the database the proxy access information for the object. Does not apply to VAX P.S.I.

USER

Removes the user access control information from the database.

Example

```
NCP> CLEAR OBJECT DTR ALL
```

This command removes all parameter entries for the DTR object from the volatile database. As a result, the object no longer exists for the local DECnet–VAX software.

CONNECT NODE

CONNECT NODE — The CONNECT NODE command sets up a logical connection between the host node and the console interface on a specified target node. Both the host node and the target node must be on the same Ethernet circuit. To define default information for the target node in the volatile database,use the NCP command SET NODE to specify the SERVICE CIRCUIT, SERVICE PASSWORD, and HARDWARE ADDRESS parameters for the target node. Once you set these default parameters, you can override them for the target node by specifying new parameters in the CONNECT command.

Format

CONNECT   [node-component [parameter] [...]]

Node Component

NODE

Node Component

NODE node-id

Identifies the node name or address of the target node to be connected to the local node.

Command Parameters

PHYSICAL ADDRESS E-address

Specifies the Ethernet physical address of the target node. The value is the Ethernet address that the target node has set for itself or, if the target node has not set an Ethernet address, the HARDWARE ADDRESS parameter associated with the target node in the executor node's volatile database.

SERVICE PASSWORD hex-password

Identifies the password required to create the logical link between the host node and the target node. The password is a 16-digit hexadecimal number in the range 0 to FFFFFFFFFFFFFFFF.

VIA circuit-id

Specifies the circuit to be used to create the logical link between the host node and the target node. This circuit must be an Ethernet circuit.

Example

```
NCP> CONNECT NODE RTRDEV SERVICE PASSWORD FEFEFEFEFEFEFEFE -
_ VIA UNA-0 PHYSICAL ADDRESS AA-00-04-00-38-04
```

This command connects the host node to the console interface on the target node RTRDEV specifying the service password FEFEFEFEFEFEFEFE, the service circuit UNA–0, and the Ethernet physical address AA–00–04–00–38–04.

CONNECT VIA

CONNECT VIA — The CONNECT VIA command sets up a logical connection between the host node and the console interface on a target node using the specified circuit. If the DECnet node address of the target node is not known, use the CONNECT VIA command instead of the CONNECT NODE command. Both the host node and the target node must be on the same Ethernet circuit.

Format

CONNECT VIA   [circuit-id parameter [...]]

Command Parameters

PHYSICAL ADDRESS E-address

Specifies the Ethernet physical address of the target node. The value is the Ethernet address that the target node has set for itself. This parameter is required.

SERVICE PASSWORD hex-password

Identifies the password required to create the logical link between the host node and the target node. The password is a hexadecimal number in the range 0 to FFFFFFFFFFFFFFFF.

Example

```
NCP> CONNECT VIA UNA-0 PHYSICAL ADDRESS AA-00-04-00-38-00
```

This command connects the host node to the console interface on the target node by specifying the circuit over which the connection is to be made and the Ethernet physical address of the target node.

COPY KNOWN NODES

COPY KNOWN NODES — The COPY KNOWN NODES command updates the node database on the local node. You can copy the volatile or permanent node database from a remote node to either or both the volatile and permanent node databases on the local node. You also have the option of clearing or purging the node database on the local node before beginning the copy operation. Only the node name and node address are copied. Anode entry will not be copied into the node database if it would result in the association of two names with one address or two addresses with one name. You cannot use the TELL prefix with the COPY KNOWN NODES command.

Format

COPY   [node-component parameter [qualifier] [...]]

Node Component

KNOWN NODES

Qualifiers

USING
WITH
TO

Node Component

KNOWN NODES

Indicates that names and addresses of all known nodes stored in the database of the specified remote node are to be copied.

Command Parameter

FROM node-id

Specifies the remote node from which node database information is to be copied. The remote node can be any node in the network to which you have access. The word FROM is optional.

Qualifiers

USING option

Specifies the node database on the remote node from which the information is to be copied. There are two possible options:

VOLATILE	Indicates that the volatile database on the remote node is to be copied.
PERMANENT	Indicates that the permanent database on the remote node is to be copied.

The default is VOLATILE.

TO option

Specifies the node database on the local node to which the information is to be copied. There are three possible options:

VOLATILE	Indicates that the information is to be copied to the volatile database on the local node.
PERMANENT	Indicates that the information is to be copied to the permanent database on the local node.
BOTH	Indicates that the information is to be copied to both the volatile and permanent databases on the local node.

The default is VOLATILE.

WITH option

Clears or purges the node database on the local node before the copy operation is performed. Retains the executor node characteristics and the name and address of the remote node from which the node information is to be copied. The node database to be cleared or purged is the local database to which the information will be copied. There are two options:

CLEAR	Clears the volatile node database at the local node.
PURGE	Purges the permanent node database at the local node.

Note that you can actually specify either CLEAR or PURGE for either database or for both databases.

If you do not specify the WITH qualifier, the node entries copied are added to the existing node database(s).

Example

```
1. NCP> LIST KNOWN NODES
   Known Node Permanent Summary as of 30-DEC-1988 13:50:20
   Executor node = 2.20 (ROBIN)
   State                   = on
   Remote node =   2.21 (THRUSH)
   No information available
   Remote node =   2.22 (LARK)
   No information available

   NCP> TELL LARK LIST KNOWN NODES
   Known Node Permanent Summary as of 30-DEC-1989 13:50:20
   Executor node = 2.22 (LARK)
   State                   = on
   Remote node =   2.20 (ROBIN)
   No information available
   Remote node =   2.23 (DOVE)
   No information available

   NCP> COPY KNOWN NODES FROM LARK USING PERMANENT -
   _ TO PERMANENT WITH PURGE
   %NCP-I-SUCCESS - Success

   Remote node = 2.21 (THRUSH)
   %NCP-I-RECDELET, Database entry deleted
   %NCP-I-SUCCESS - Success

   Remote node = 2.22 (LARK)
   %NCP-I-RECDELET, Database entry deleted
   %NCP-I-SUCCESS - Success

   Executor node = 2.20 (ROBIN)
   %NCP-I-RECDELET, Database entry deleted

   NCP> LIST KNOWN NODES
```

```
Known Node Permanent Summary as of 30-DEC-1989 14:01:05
Executor node = 2.20 (ROBIN)
State                   = on
Remote node =   2.22 (LARK)
No information available

Remote node =   2.23 (DOVE)
No information available
```

This copy command copies the node information from the permanent node database on node LARK into the permanent node database on the local node (ROBIN). The node database is purged before the copy operation is begun.

2.
```
NCP> SHOW KNOWN NODES
Known Node Volatile Summary as of 30-DEC-1989 13:50:20
Executor node = 2.20 (ROBIN)
State                   = on
Identification          = DECnet-VAX V5.0,  VMS V5.0
Active links            = 2
   Node            State      Active  Delay   Circuit      Next node
                              Links
 2.21 (THRUSH)                                QNA-0        2.19 (EGRET)
 2.22 (LARK)                                  QNA-0        2.19 (EGRET)

NCP> TELL LARK SHOW KNOWN NODES
Known Node Volatile Summary as of 30-DEC-1989 13:53:27
Executor node = 2.22 (LARK)
State                   = on
Identification          = DECnet-VAX V5.0,  VMS V5.0
Active links            = 2
   Node            State      Active  Delay   Circuit      Next node
                              Links
 2.20 (ROBIN)                                 QNA-0        2.19 (EGRET)
 2.23 (DOVE)                                  QNA-0        2.19 (EGRET)

NCP> COPY KNOWN NODES FROM LARK
NCP> SHOW KNOWN NODES
Known Node Volatile Summary as of 30-DEC-1989 14:01:05
Executor node = 2.20 (ROBIN)
State                   = on
Identification          = DECnet-VAX V5.0,  VMS V5.0
Active links            = 2
   Node            State      Active  Delay   Circuit      Next node
                              Links
 2.21 (THRUSH)                                QNA-0        2.19 (EGRET)
 2.22 (LARK)                                  QNA-0        2.19 (EGRET)
 2.23 (DOVE)                                  QNA-0        2.19 (EGRET)
```

This COPY command copies the node information from the volatile node database on node LARK into the volatile node database on the local node (ROBIN). In this example, the volatile node database on the local node is not cleared before the copy operation is begun.

DISCONNECT LINK

DISCONNECT LINK — The DISCONNECT LINK command disconnects logical links active on the local node.

Format

DISCONNECT   [link-component [qualifier]]

Link Components

LINK
KNOWN LINKS

Qualifier

WITH NODE

Link Components

KNOWN LINKS

Indicates that all links are to be disconnected.

LINK number

Identifies the particular logical link to be disconnected. Use a number in the range of 1 to 65,535.

Qualifier

WITH NODE node-id

Identifies the node with which all logical links are to be disconnected. The *node-id* is either a node name or a node address. You can use this command verb only with the KNOWN LINKS component.

Example

1.  ```
    NCP> SHOW KNOWN LINKS
    Known Link Volatile Summary as of 30-DEC-1989 15:53:42
       Link        Node           PID        Process      Remote link  Remote
      user
       4104   11.9 (DALLAS)     20600090  GRAY              3116    MAIL
       8229    3.5 (TRNTO)      2060014C  DAVIS             7217    NML
       3125    2.17 (NYC)       2060023D  REMACP            5175    THOMPSON_1
    NCP> DISCONNECT KNOWN LINKS WITH NODE DALLAS
    ```

    The first command, SHOW KNOWN LINKS, identifies the link addresses. The second command disconnects all known links to remote node DALLAS.

2.  ```
    NCP> SET EXECUTOR NODE DENVER...
    NCP> DISCONNECT KNOWN LINKS
    NCP> CLEAR EXECUTOR NODE
    ```

    The first command sets the executor node to DENVER. The DISCONNECT KNOWN LINKS command disconnects the link to NML on DENVER. As a result, you must either enter the SET EXECUTOR NODE command again, or you must enter the CLEAR EXECUTOR NODE command to reestablish communication with the local node's NML.

HELP

HELP — The HELP command displays general information about NCP commands and parameters. See Section 1.5 for more information about the NCP HELP Facility.

Format

HELP [topic...]

Command Parameter

topic

A command word listed in the HELP display. You may specify up to eight topics separated by spaces or tabs.

Example

1.  ```
    NCP> HELP
    Information available:
        CLEAR       Commands  CONNECT    COPY       DEFINE     DISCONNECT
     EVENTS
        EXIT        HELP      LIST       LOAD       LOOP       PARAMETERS
     Prompting
        PURGE       SET       SHOW       TELL       TRIGGER    ZERO
    Topic?
    ```

    This command displays all command verbs for which further information exists.

2.  ```
    NCP> HELP CLEAR CIRCUIT
    CLEAR

    CIRCUIT
        The CLEAR CIRCUIT command resets circuit parameters in the volatile
     database to the default value (if any), or removes them from the
     volatile database on the local node or DTE.
        The PURGE CIRCUIT command always removes circuit parameters from the
     permanent database on the local node or DTE.

        CLEAR   KNOWN CIRCUITS          (parameters...)
                CIRCUIT circuit-id

    Additional information available:
        KNOWN CIRCUITS        CIRCUIT circuit-id   ACTIVE BASE
        ACTIVE INCREMENT      ALL        BABBLE TIMER         COUNTER TIMER
        DEAD THRESHOLD        DYING BASE DYING INCREMENT       DYING
     THRESHOLD
        INACTIVE BASE         INACTIVE INCREMENT   INACTIVE THRESHOLD
        MAXIMUM BUFFERS       MAXIMUM RECALLS      MAXIMUM ROUTERS
        MAXIMUM TRANSMITS     NETWORK    RECALL TIMER          ROUTER
     PRIORITY
        TRANSMIT TIMER        Examples
    ```

    This command provides a description of the NCP command CLEAR CIRCUIT and displays command words for which further information exists.

3. ```
NCP> HELP SHOW
   SHOW
     Use the SHOW command to display information from the volatile database
    on the executor node. Use the LIST command to display information from
    the permanent database on the executor node.

   Additional information available:
     CHARACTERISTICS   COUNTERS   EVENTS   STATUS   SUMMARY
     TO file-id AREA   CIRCUIT   EXECUTOR   LINE   LINKS
    LOGGING
      MODULE   NODE   OBJECT
```

This command provides a description of the NCP command SHOW and displays command words for which further information exists.

# LOAD NODE

LOAD NODE — The LOAD NODE command downline loads software to a target node, using either the identified circuit (if specified) or the circuit obtained from the volatile database. Any parameter left unspecified in the command defaults to whatever entry is specified in the volatile database on the executor node.

## Format

**LOAD**   [node-component parameter [...]]

### Node Component

NODE

## Node Component

### NODE node-id

Identifies the node name or address of the target node to be downline loaded.

## Command Parameters

### ADDRESS node-address

Identifies the address that the target node is to use (when it comes up).

### CPU cpu-type

Identifies the node's CPU type. There are four possibilities:

DECSYSTEM1020
PDP11
PDP8
VAX

### FROM filespec

Identifies the file specification of the load file containing the system software to be downline loaded.

**HOST node-id**

Identifies the default host that the target node is to use (when it comes up).

**LOAD ASSIST AGENT filespec**

Specifies the VMS image that defines the system software to be downline loaded to the target node. The load assist agent can be used to tailor the system software to be loaded. The LOAD ASSIST AGENT parameter applies to downline load operations to Local Area VAX clusters only.

**LOAD ASSIST PARAMETER item**

Specifies a parameter to be passed to a load assist agent. The LOADASSIST PARAMETER value applies to downline load operations to Local Area VAXclusters only.

**MANAGEMENT FILE filespec**

Specifies a file containing the management information to be downline loaded to the target node.

**NAME node-name**

Identifies the name that the target node is to use (when it comes up).

**PHYSICAL ADDRESS E-address**

Applies only to nodes on Ethernet circuits. Specifies the Ethernet address that the target node to be downline loaded currently uses to identify itself. The value is the Ethernet address that the target node has set for itself or, if the target node has not set an Ethernet address, the HARDWAREADDRESS parameter if associated with the target node in the executor node's volatile database.

**SECONDARY LOADER filespec**

Identifies the file specification of a secondary load file containing the software to be downline loaded.

**SERVICE DEVICE device-type**

Identifies the target node's line controller for the service line over which the operation is to take place. The possibilities areas follows:

| BNA | DA | DL | DLV | DMB | DMC |
|-----|-----|-----|-----|-----|-----|
| DMF | DMP | DMR | DMV | DP | DQ |
| DTE | DU | DUP | DV | DZ | KDP |
| KDZ | KL | PCL | QNA | SVA | UNA |

Only the synchronous port on the DMF service device may be used.

**SERVICE PASSWORD hex-password**

Identifies the password required to trigger the boots trap mechanism on the target node. The password is a hexadecimal number. For DDCMP circuits, it is in the range 0 to FFFFFFFF; for Ethernet circuits, it is in the range 0 to FFFFFFFFFFFFFFFF.

**SOFTWARE IDENTIFICATION software-id**

Is the ID of the software to be downline loaded.

**SOFTWARE TYPE software-type**

Identifies the particular file type to be loaded first. There are four possibilities:

MANAGEMENT FILE
SECONDARY LOADER
SYSTEM
TERTIARY LOADER

**TERTIARY LOADER filespec**

Identifies the file specification of a tertiary load file containing the software to be downline loaded.

**VIA circuit-id**

Identifies a circuit over which the loading sequence is to take place.

# Example

1. `NCP> LOAD NODE NYC HOST BOSTON VIA DMC-2`

   This command initiates a downline load operation for node NYC over circuit DMC–2. When loaded, node NYC has node BOSTON as its default host specification.

2. `NCP> LOAD NODE BANGOR`

   This command initiates a downline load operation for node BANGOR. Any required default information is retrieved from the volatile database on the executor node.

3. `NCP> LOAD NODE SWIFT PHYSICAL ADDRESS AA-00-04-00-07-04`

   This command initiates a downline load operation for node SWIFT on an Ethernet circuit. The executor node uses the Ethernet physical address specified in the command to address node SWIFT.

# LOAD VIA

LOAD VIA — The LOAD VIA command downline loads software to a target node using the specified circuit. The target node identification is obtained from the volatile database on the executor node. If the target node is on an Ethernet circuit, you must specify the PHYSICAL ADDRESS parameter in this command.

# Format

**LOAD VIA**  [circuit-id parameter [...]]

# Command Parameters

**ADDRESS node-address**

Identifies the address that the target node is to use (when it comes up).

**CPU cpu-type**

Identifies the node's CPU type. There are four possibilities:

DECSYSTEM1020
PDP11
PDP8
VAX

**FROM filespec**

Identifies the file specification of the load file containing the system software to be downline loaded.

**HOST node-id**

Identifies the default host that the target node is to use (when it comes up).

**LOAD ASSIST AGENT filespec**

Specifies the VMS image that defines the system software to be downline loaded to the target node. The load assist agent can be used to tailor the system software to be loaded. The LOAD ASSIST AGENT parameter applies to downline load operations to Local Area VAX clusters only.

**LOAD ASSIST PARAMETER item**

Specifies a parameter to be passed to a load assist agent. The LOADASSIST PARAMETER value applies to downline load operations to Local Area VAXclusters only.

**MANAGEMENT FILE filespec**

Specifies a file containing the management information to be downline loaded to the target node.

**NAME node-name**

Identifies the name that the target node is to use (when it comes up).

**PHYSICAL ADDRESS E-address**

Applies only to nodes on Ethernet circuits. Specifies the Ethernet address that the target node to be downline loaded currently uses to identify itself. The value is the Ethernet address that the target node has set for itself or, if the target node has not set an Ethernet address, the HARDWAREADDRESS parameter if associated with the target node in the executor node's volatile database.

**SECONDARY LOADER filespec**

Identifies the file specification of a secondary load file containing the software to be downline loaded.

**SERVICE DEVICE device-type**

Identifies the target node's line controller for the service line over which the operation is to take place. The possibilities are as follows:

| BNA | DA | DL | DLV | DMB | DMC |
|-----|-----|-----|-----|-----|-----|
| DMF | DMP | DMR | DMV | DP | DQ |
| DTE | DU | DUP | DV | DZ | KDP |
| KDZ | KL | PCL | QNA | SVA | UNA |

Only the synchronous port on the DMF service device may be used.

**SERVICE PASSWORD hex-password**

Identifies the password required to trigger the bootstrap mechanism on the target node. The password is a hexadecimal number. For DDCMP circuits,it is in the range 0 to FFFFFFFF; for Ethernet circuits, it is in the range 0 to FFFFFFFFFFFFFFFF.

**SOFTWARE IDENTIFICATION software-id**

Is the ID of the software to be downline loaded.

**SOFTWARE TYPE software-type**

Identifies the particular file type to be loaded first. There are four possibilities:

MANAGEMENT FILE
SECONDARY LOADER
SYSTEM
TERTIARY LOADER

**TERTIARY LOADER filespec**

Identifies the file specification of a tertiary load file containing the software to be downline loaded.

# Example

1. `NCP> LOAD VIA DMP-0.4`

   This command initiates a downline load operation over circuit DMP-0.4. Any required default information is retrieved from the volatile database on the executor node. NCP scans the node database until it finds a node whose service circuit matches the load circuit.

2. `NCP> LOAD VIA DMC-0 SOFTWARE TYPE SECONDARY LOADER`

   This command initiates a downline load operation over circuit DMC–0, commencing with the secondary loader file.

3. `NCP> LOAD VIA UNA-0 PHYSICAL ADDRESS AA-00-04-00-07-04`

   This command initiates a downline load operation over Ethernet circuit UNA–0, to the target node whose Ethernet physical address is specified in the command.

# LOOP CIRCUIT

LOOP CIRCUIT — The LOOP CIRCUIT command tests a specified circuit in the network by causing test blocks of data to be transmitted over the specified circuit. The parameters are optional and can be entered in any order.

# Format

`LOOP`   [circuit-component [parameter] [...]]

## Circuit Component

CIRCUIT

# Circuit Component

**CIRCUIT circuit-id**

Does not apply to X.25 circuits. Identifies the circuit for loop back testing.

# Command Parameters

**ASSISTANT NODE node-id**

Applies only to Ethernet circuits. Identifies the name of the node or the address of the node that is to perform the role of loopback assistant for Ethernet third-party loop testing. You can use this parameter instead of the ASSISTANT PHYSICAL ADDRESS parameter.

**ASSISTANT PHYSICAL ADDRESS E-address**

Applies only to Ethernet circuits. Identifies the Ethernet physical address of the node that is to perform the role of loopback assistant for Ethernet third-party loop testing. If HELP is included in this command,you must specify ASSISTANT PHYSICAL ADDRESS. The address cannot be a multicast address. You can use this parameter instead of the ASSISTANT NODE parameter.

**COUNT number**

Specifies the number of blocks to be sent during loopback testing over the circuit. The count must be a decimal integer in the range of 1 through 65,535. If you omit this parameter, only one block is looped.

**HELP help-type**

Applies only to Ethernet circuits. Indicates the amount of assistance to be provided during Ethernet loopback testing by the assistant node, whose address is specified in the ASSISTANT PHYSICAL ADDRESS or *node-id* as specified in the ASSISTANT NODE parameter. There are three possible values for *help-type*.

| | |
|---|---|
| FULL | The assistant node relays the request and the reply between the executor node and the destination node. |
| RECEIVE | The executor node sends the request directly to the destination node, which relays the reply to the assistant node for transmission to the executor node. |
| TRANSMIT | The assistant node relays the request to the destination node, which replies directly to the executor node. |

If you specify HELP, you must also specify ASSISTANT PHYSICAL ADDRESS or ASSISTANT NODE.

**LENGTH number**

Specifies the length (in bytes) of the blocks to be sent during loopback testing. The length must be a decimal integer in the range of 1 through 65,535. Note that the LENGTH parameter must be less than 50 for the DMC–11 circuit operating in either controller loopback mode or cable loopback (a DMC–11 with an attached loopback cable). If you omit this parameter, a block length of 40 bytes is used.

**NODE node-id**

Applies only to Ethernet circuits. Identifies the destination node to be used for loopback testing of the specified Ethernet circuit. You can use this parameter instead of the PHYSICAL ADDRESS parameter.

**PHYSICAL ADDRESS E-address**

Applies only to Ethernet circuits. Identifies the Ethernet physical address of the destination node to be used for loopback testing of the specified Ethernet circuit.

**WITH data-type**

Specifies the type of binary information to be sent during testing. If you omit this parameter, a combination of ones and zeros (the MIXED data type) is sent. There are three types of data:

MIXED
ONES
ZEROS

# Example

1. `NCP> LOOP CIRCUIT DMC-0 COUNT 5 LENGTH 20`

   This command initiates a circuit-level loopback test over circuit DMC–0. The software loops five messages, 20 bytes in length with mixed binary information.

2. `NCP> LOOP CIRCUIT UNA-0 PHYSICAL ADDRESS AA-00-04-00-FF-04`

   This command initiates an Ethernet circuit-level loopback test with anode whose Ethernet physical address is AA–00–04–00–FF–04.

3. `NCP> LOOP CIRCUIT UNA-0 NODE 224`

   This command initiates an Ethernet circuit-level loopback test with anode whose address is 224. In this example, the NODE parameter with a *node-id* value of 224 is used in place of the PHYSICAL ADDRESS parameter.

4. `NCP> LOOP CIRCUIT UNA-0 PHYSICAL ADDRESS -`
   `_ AA-00-04-00-12-04 ASSISTANT NODE GULL HELP RECEIVE`

   This command initiates an Ethernet circuit-level loopback test by transmitting a message directly to a node whose Ethernet physical address is AA–00–04–00–12–04. Additionally, the command requests the assistance of an Ethernet node named GULL to receive the message from the target node and then relay the message back.

# LOOP EXECUTOR

LOOP EXECUTOR — The LOOP EXECUTOR command tests the executor node in the network by causing test data to be transmitted to the executor node. The parameters are optional and can be entered in any order. You can supply explicit access control information for the LOOP EXECUTOR command.

## Format

**LOOP EXECUTOR** [[parameter] [...]]

## Command Parameters

**ACCOUNT account**

Identifies the user's account for access control verification for the executor node.

**COUNT number**

Specifies the number of blocks to be sent during loopback testing over the executor node. The count must be a decimal integer in the range of 1 through 65,535. If you omit this parameter, only one block is looped.

**LENGTH number**

Specifies the length (in bytes) of the blocks to be sent during loopback testing. The length must be a decimal integer in the range of 1 through 65,535. If you omit this parameter, a block length of 40 bytes is used.

**PASSWORD password**

Identifies the user's password for access control verification for the executor node.

**USER user-id**

Specifies the user's identification for access control verification for the executor node.

**WITH data-type**

Specifies the type of binary information to be sent during testing. If you omit this parameter, a combination of ones and zeros (the MIXED data type) is sent. There are three types of data:

MIXED
ONES
ZEROS

# Example

1. `NCP> LOOP EXECUTOR`

   This command initiates a loopback test on the executor node.

# LOOP LINE

LOOP LINE — The LOOP LINE command initiates loopback tests of X.25 lines only. The parameters are optional and can be entered in any order. Before beginning loopback testing of an X.25 line, use the SET LINE command to set the STATESERVICE and CONTROLLER LOOPBACK parameters for internal loopback testing or the STATE SERVICE and CONTROLLER NORMAL parameters for all other loopback testing.

# Format

`LOOP` [line-component [parameter] [...]]

**Line Component**

LINE

## Line Component

**LINE line-id**

Applies only to VAX P.S.I. Identifies the X.25 line for loopback testing. Specify a line name using the *dev-c-u* format for the *line-id*.

## Command Parameters

**COUNT number**

Specifies the number of blocks to be sent during loopback testing over the line. The count must be a decimal integer in the range of 1 through 65,535. If you omit this parameter, only one block is looped.

**LENGTH number**

Specifies the length (in bytes) of the blocks to be sent during loopback testing. The length must be a decimal integer in the range of 1 through 65,535. If you omit this parameter, a block length of 40 bytes is used.

**WITH data-type**

Specifies the type of binary information to be sent during testing. If you omit this parameter, a combination of ones and zeros (the MIXED datatype) is sent. There are three types of data:

MIXED
ONES
ZEROS

## Example

1. `NCP> LOOP LINE DUP-0`

   This command initiates a line-level loopback test over X.25 line DUP–0. The software loops one message, 40 bytes in length, with mixed binary information.

2. `NCP> LOOP LINE DUP-1 COUNT 10 WITH ONES`

   This command initiates a line-level loopback test over X.25 line DUP–1. The software loops 10 messages, 40 bytes in length, with all binary ones information.

# LOOP NODE

LOOP NODE — The LOOP NODE command tests a specified node (other than the executor node) in the network by causing test blocks of data to be transmitted to the node. The parameters are optional and can be entered in any order. You can supply explicit access control information for the LOOP NODE command. To test the executor node, use the LOOP EXECUTOR command.

## Format

**LOOP**   [node-component [parameter] [...]]

**Node Component**

NODE

# Node Component

**NODE node-id**

Does not apply to VAX P.S.I. Identifies the node on which the loopback mirror is to run.

# Command Parameters

**ACCOUNT account**

Identifies the user's account for access control verification for the designated node.

**COUNT number**

Specifies the number of blocks to be sent during loopback testing over the node. The count must be a decimal integer in the range of 1 through 65,535. If you omit this parameter, only one block is looped.

**LENGTH number**

Specifies the length (in bytes) of the blocks to be sent during loopback testing. The length must be a decimal integer in the range of 1 through 65,535. Note that the LENGTH parameter must be less than 50 for the DMC−11 circuit operating in either controller loopback mode or cable loopback (a DMC−11 with an attached loopback cable). If you omit this parameter, a block length of 40 bytes is used.

**PASSWORD password**

Identifies the user's password for access control verification for the designated node.

**USER user-id**

Specifies the user's identification for access control verification for the designated node.

**WITH data-type**

Specifies the type of binary information to be sent during testing. If you omit this parameter, a combination of ones and zeros (the MIXED datatype) is sent. There are three types of data:

MIXED
ONES
ZEROS

# Example

1. ```
   NCP> SET NODE TESTER CIRCUIT DMC-0
   NCP> LOOP NODE TESTER...
   NCP> CLEAR NODE TESTER CIRCUIT
   ```

   The first command creates a loop node name (TESTER) for the associated circuit, the second command initiates a node-level loopback test with the loop node name, and the third command removes the loop node name from the volatile database when the loop test is completed.

SET CIRCUIT ALL

SET CIRCUIT ALL — The SET CIRCUIT ALL command updates the volatile copy of the database on the local node with all the circuit parameters stored for a particular circuit or for all known circuits in the permanent database on the local node.

Format

SET   [circuit-component ALL]

Circuit Components

CIRCUIT
KNOWN CIRCUITS

Circuit Components

CIRCUIT circuit-id

Instructs NCP to update the parameters for a specific circuit.

KNOWN CIRCUITS

Instructs NCP to update all parameters for all known circuits.

Example

1. `NCP> SET KNOWN CIRCUITS ALL`

   This command loads all permanent database entries stored for all known circuits into the volatile database.

SET/DEFINE CIRCUIT

SET/DEFINE CIRCUIT — The SET CIRCUIT command creates or modifies circuit parameters in the volatile database on the local node. The DEFINE CIRCUIT command creates or modifies circuit parameters in the permanent database on the local node. The circuit must be in the OFF state before you modify any parameters other than COST, COUNTER TIMER, STATE, or VERIFICATION.

Format

SET   [circuit-component parameter [...]]

DEFINE   [circuit-component parameter [...]]

Circuit Components

CIRCUIT
KNOWN CIRCUITS

Circuit Components

CIRCUIT circuit-id

Identifies the circuit whose parameters are to be updated.

KNOWN CIRCUITS

Indicates that parameters for all known circuits are to be updated.

Command Parameters

ACTIVE BASE base

Applies only to DDCMP CONTROL circuits. Sets the base priority of an ACTIVE tributary after the tributary has been polled. You can set a separate base for each of the indicated polling states. The base value must be a decimal integer from 0 to 255. The default is 255.

ACTIVE INCREMENT increment

Applies only to DDCMP CONTROL circuits. Sets the increment value added to the ACTIVE tributary priority each time the scheduling timer expires. The *increment* value must be a decimal integer from 0 to 255. The default is 0.

BABBLE TIMER milliseconds

Applies only to DDCMP CONTROL circuits. Represents the number of milliseconds that a selected tributary or remote half-duplex station is allowed to transmit. The *milliseconds* value must be a decimal integer in the range 1 to 65,535. The default is 6000 (6 seconds).

CHANNEL number

Applies only to X.25 and DLM permanent virtual circuits (PVCs). Identifies the logical channel number for the X.25 PVCs. Specify a value in the range 1 to 4095. When you specify an X.25 PVC or a DLM PVC for the first time, this parameter is mandatory.

COST cost

Specifies the routing cost of the circuit. Messages travel between nodes along the path with the smallest total cost. The *cost* value must be a decimal integer in the range 1 to 25. The default value is 10.

COUNTER TIMER seconds

Specifies the number of seconds that the circuit counter timer will run. When the counter timer expires, a circuit counter logging event occurs. The *seconds* value must be a decimal integer in the range 0 to 65,535. If no value is set for COUNTER TIMER, the circuit counters are not logged automatically.

DEAD THRESHOLD count

Applies only to DDCMP CONTROL circuits. Defines the number of times to poll the active, inactive, or dying tributary before changing that tributary's polling state to DEAD because of receive timeouts. The *count* value must be a decimal integer in the range 0 to 255. The default count is 8.

DTE dte-address

Applies only to X.25 PVCs and DLM circuits. Identifies the local DTE for the circuit. Specify a decimal integer of 1 to 15 digits. See the *Public Network Information Guide* for the format of the address on your network. This parameter is optional for DLM SVCs and mandatory when you specify an X.25 PVC or DLM PVC for the first time.

DYING BASE base

Applies only to DDCMP CONTROL circuits. Sets the base priority of a DYING tributary after the tributary has been polled. You can set a separate base for each of the indicated polling states. The *base* value must be a decimal integer from 0 to 255. The default is 0.

DYING INCREMENT increment

Applies only to DDCMP CONTROL circuits. Sets the increment value added to the DYING tributary priority each time the scheduling timer expires. The *increment* value must be a decimal integer from 0 to 255. The default is 16.

DYING THRESHOLD count

Applies only to DDCMP CONTROL circuits. Specifies the number of times to poll the active or inactive tributary before changing that tributary's polling state to dying because of receive timeouts. The *count* value must be a decimal integer in the range 0 to 255. The default count is 2.

HELLO TIMER seconds

Specifies the frequency of Routing Hello messages sent to adjacent nodes on the circuit. The *seconds* value must be a decimal integer in the range 0 to 8191. The default value is 15. The value of the read-only circuit parameter LISTEN TIMER is three times the value of the HELLO TIMER parameter.

INACTIVE BASE base

Applies only to DDCMP CONTROL circuits. Sets the base priority of an inactive tributary after the tributary has been polled. You can set a separate base for each of the indicated polling states. The *base* value must be a decimal integer from 0 to 255. The default is 0.

INACTIVE INCREMENT increment

Applies only to DDCMP CONTROL circuits. Sets the increment value added to the inactive tributary priority each time the scheduling timer expires. The *increment* value must be a decimal integer from 0 to 255. The default is 64.

INACTIVE THRESHOLD count

Applies only to DDCMP CONTROL circuits. Specifies the number of times to poll the active tributary before changing that tributary's polling state to inactive because of no data response. The *count* value must be a decimal integer in the range 0 to 255. The default count is 8.

MAXIMUM BUFFERS count

Applies only to DDCMP CONTROL circuits. Specifies the maximum number of buffers from a common buffer pool that the tributary can use. If you do not set this parameter, there is no common buffer pool and the higher level will explicitly supply the buffers. The *count* value must be a decimal integer in the range up to 254 or the word UNLIMITED.

MAXIMUM DATA count

Applies only to X.25 PVCs and DLM circuits. Specifies the maximum packet size for the X.25 circuit. This value must be at least 5 bytes less than the MAXIMUM BLOCK value that you specify in the SET or DEFINE LINE command and must be a power of 2. Specify a value in the range 16 to 4096 bytes. By default, the *packet size* takes the value specified by the PROFILE parameter of the SET or DEFINE

MODULE X25-PROTOCOL command. See the *Public Network Information Guide* for the network value of this parameter.

MAXIMUM RECALLS count

Applies only to DLM outgoing switched virtual circuits (SVCs). Specifies the maximum number of call retries generated on an outgoing DLM circuit before the circuit is declared nonoperational. The range of possible values for MAXIMUM RECALLS is 0 to 255. A *count* value of 0 specifies an infinite number of call retries. The default value is 100.

MAXIMUM ROUTERS number

Applies only to Ethernet circuits. Specifies the maximum number of routers (other than the executor node) allowed by the Routing layer on this circuit. Use a number in the range 1 to 33. The default value is 33.

MAXIMUM TRANSMITS count

Applies only to DDCMP CONTROL circuits. Specifies the maximum number of data messages that can be transmitted at one time. The *count* value must be a decimal integer in the range 1 to 255. The default is 4.

MAXIMUM WINDOW count

Applies only to X.25 PVCs and DLM circuits. Specifies the maximum window size allowed for the X.25 circuit—that is, the maximum number of packets for which outstanding acknowledgments are allowed. Specify a value in the range 1 to 127. By default, *window size* takes the value specified by the PROFILE parameter of the SET or DEFINE MODULE X25-PROTOCOL command. See the *Public Network Information Guide* for the network value of this parameter.

NETWORK network-name

Applies only to X.25 PVCs and DLM circuits. Specifies the network that the circuit runs through. If only one network is set up, this parameter is not required. Otherwise, the parameter is mandatory.

NUMBER dte-address

Applies only to DLM switched virtual circuits. For outgoing circuits, specifies the remote DTE address and subaddress that this DLM circuit will call. For incoming circuits, specifies the address of the remote DTE from which this circuit will accept calls. If the NUMBER parameter is not specified for an incoming circuit, a call from any remote DTE can be directed to this circuit. Specify a decimal integer of 1 to 15 digits.

OWNER EXECUTOR

Applies only to DLM circuits. Identifies the circuit owner as the Routing layer. This parameter is mandatory for a DLM circuit.

POLLING STATE polling-state

Applies only to DDCMP CONTROL circuits. Identifies the multipoint polling state of the tributary. There are five possible states:

AUTOMATIC	The tributary's state varies according to the operation of the polling algorithm. This is the default.

ACTIVE	The tributary is locked in the ACTIVE state.
DEAD	The tributary is locked in the DEAD state.
DYING	The tributary is locked in the DYING state.
INACTIVE	The tributary is locked in the INACTIVE state.

RECALL TIMER seconds

Applies only to outgoing DLM SVCs. Sets a timer whose expiration causes a remote DTE to be called again during an attempt to set up a DLM circuit. Specify a decimal integer up to a maximum of 65,535.

ROUTER PRIORITY number

Applies only to Ethernet circuits. Specifies the priority this router (the executor node on this circuit) is to have in the selection of a designated router for this circuit. Use a value in the range 0 to 127. The default is 64.

SERVICE service-mode

Specifies whether service operations (loading and loop testing) are allowed for the circuit. There are two possible modes:

DISABLED	The circuit cannot be put into service state and cannot perform service functions. The default mode is DISABLED.
ENABLED	The circuit can be put into service state and perform service functions.

STATE circuit-state

Specifies the circuit's operational state. There are three possible states:

OFF	The circuit is not in use.
ON	The circuit is available for normal use or service functions.
SERVICE	The circuit is available for service functions only.

TRANSMIT TIMER milliseconds

Defines the number of milliseconds to delay between data message transmits. The *milliseconds* value must be a decimal integer up to 65,535. The default is 0.

TRIBUTARY tributary-address

Applies only to DDCMP CONTROL circuits. Defines the data link physical tributary address of the circuit. The *tributary-address* value must be a decimal integer up to a maximum of 255.

TYPE X25

Applies only to X.25 circuits. Specifies the type of circuit.

USAGE usage-type

Applies only to X.25 circuits. Defines the usage type of an X.25 virtual circuit. There are three possible usage types:

INCOMING	Applies only to DLM switched virtual circuits (SVCs). The circuit is used only for incoming calls.
OUTGOING	Applies only to DLM SVCs. The circuit is used only for outgoing calls.
PERMANENT	Mandatory for X.25 PVCs. The circuit is permanently connected to a remote DTE and does not need to be switched dynamically.

VERIFICATION option

Applies only to synchronous and asynchronous circuits. Requires the remote node to send its routing initialization password. There are three options:

DISABLED	Does not require the remote node to send its routing initialization password. This is the default.
ENABLED	Requires the remote node to send its routing initialization password.
INBOUND	Applies to any DDCMP point-to-point circuit. Specifies that the executor node expects to receive a routing initialization password for verification from a remote node before a connection is made between the nodes. The executor is prohibited from sending its routing initialization password to the remote node. This parameter is specified automatically for dynamic asynchronous DDCMP circuits. If you specify the VERIFICATION INBOUND parameter for a circuit, you must specify the INBOUND node parameter (by using the SET/DEFINE NODE command) for the remote node.

Description

If you plan to run DECnet over a CI, you must first install the driver CNDRIVER. To do this, add the following lines to the LOADNET.COM command procedure in SYS$MANAGER:

```
$ RUN SYS$SYSTEM:SYSGENCONNECT CNA0/NOADAPTER
```

Refer to the *VSI OpenVMS DECnet Networking Manual* for a table of circuit parameters grouped according to circuit type.

Example

1. `NCP> SET CIRCUIT DMC-0 COST 4 STATE ON`

   This command sets the circuit STATE to ON and the circuit cost to 4 for circuit DMC–0.

2. `NCP> SET CIRCUIT UNA-0 STATE ON MAXIMUM ROUTERS 5`

   This command sets Ethernet circuit UNA–0 to ON and sets the maximum number of routers permitted on the circuit to 5.

3. `NCP> DEFINE CIRCUIT X25-ANDIES CHANNEL 3 DTE 123789456 -`
   `_ NETWORK PSDND MAXIMUM DATA 1024 MAXIMUM WINDOW 7 USAGE PERMANENT STATE`
   ` ON`

   This command defines an X.25 PVC called X25-ANDIES with a logical channel number of 3 on the local DTE with address 123789456 on network PSDND. The circuit has a maximum packet size of 1024 bytes and a maximum window size of 7.

4. `NCP> DEFINE CIRCUIT X25-DLM-0 NUMBER 456789012 NETWORK PSDNA –`
   `_ USAGE OUTGOING STATE ON`

   This command defines a DLM SVC called X25-DLM-0 connected to the remote DTE on network PSDNA with address 456789012, and used for outgoing calls.

5. `NCP> DEFINE CIRCUIT X25-DLM1 NUMBER 12345677 NETWORK PSDNB –`
   `_ USAGE INCOMING STATE ON`

   This command defines a DLM SVC called X25-DLM-1 on network PSDNB that accepts only incoming calls from the remote DTE with address 12345677.

6. `NCP> SET CIRCUIT DMC-* COST 5`

   This command sets the circuit cost to 5 for all DMC circuits.

SET EXECUTOR ALL

SET EXECUTOR ALL — The SET EXECUTOR ALL command updates the volatile copy of the local node's database with all the local node parameters stored for that node in its permanent database.

Format

SET EXECUTOR ALL

Parameters

None.

Example

1. `NCP> SET EXECUTOR ALL`

   This command loads all permanent database entries stored for the local node into the volatile database.

SET/DEFINE EXECUTOR

SET/DEFINE EXECUTOR — The SET EXECUTOR command creates or modifies parameters in the volatile database that controls the network on the local node. The DEFINE EXECUTOR command creates or modifies parameters in the permanent database that controls the network on the local node. After the local node's state is set to ON, you cannot change the ADDRESS, ALIASNODE, ALIAS INCOMING, BUFFER SIZE, NAME, or TYPE parameter for the local node. If the local node whose state is ON is connected to an Ethernet circuit whose state is ON, you cannot change the MAXIMUM CIRCUITS parameter for the local node. You cannot use the TELL prefix with the SET EXECUTOR command.

Format

SET EXECUTOR [parameter [...]]

DEFINE EXECUTOR [parameter [...]]

Command Parameters

ADDRESS node-address

Establishes a node address for the local node, in the following format:

```
area-number.node-number
```

where:

area-number	Is in the range 1 to 63.
node-number	Is in the range 1 to 1023.

If you do not specify *area-number*, the default value is 1. You need not supply the area number in the *node-address* if your node is in area 1. When you configure the local node, this parameter is required.

ALIAS INCOMING option

Specifies whether the local node accepts incoming connect requests directed to the alias node identifier specified for the local node. The alias node identifier is described under the ALIAS NODE parameter. There are two options for ALIAS INCOMING:

DISABLED	Specifies that the local node will not accept incoming connect requests directed to the alias node identifier.
ENABLED	Specifies that the local node will accept incoming connect requests directed to the alias node identifier. This is the default if an alias node identifier has been specified.

ALIAS MAXIMUM LINKS number

Specifies the maximum number of logical links for the local node that can use the alias node identifier. The alias node identifier is described under the ALIAS NODE parameter. The maximum value for ALIAS MAXIMUM LINKS is 200. The default value is 32.

ALIAS NODE node-id

Establishes a cluster alias node identifier for use by the local node. The *node-id* is a DECnet node identifier that can be either anode name or a node address. This alias permits the local node to be associated with a cluster node identifier common to some or all nodes in the cluster, in addition to its own unique *node-id*. If you do not specify this parameter, the local node is not associated with a cluster alias node identifier. If a node name is to be used as the alias *node-id*, the node name must previously have been defined in the database.

AREA MAXIMUM COST number

Applies only to an executor node whose type is AREA. Specifies the maximum total path cost allowed from the executor to any other level 2 routing node (area router). You can specify a decimal value in the range 1 to 1022. The default is 1022.

AREA MAXIMUM HOPS number

Applies only to an executor node whose type is AREA. Specifies the maximum number of routing hops allowable from the executor to any other level 2 routing node. You can specify a decimal value in the range 1 to 30. The default is 30.

BROADCAST ROUTING TIMER seconds

Specifies the maximum amount of time allowed between routing updates on Ethernet circuits. When the timer expires before a routing update occurs, a routing update is forced. The routing update produces a routing configuration message for each adjacent node. You can specify a number in the range 1 to 65,535. The default is 40.

BUFFER SIZE number

Specifies in bytes the size of the receive buffers, thereby controlling the maximum size of NSP message segments that can be received and forwarded. (The size includes protocol overhead down to and including the End Communication layer, but does not include the Data Link layer overhead.) This buffer size applies to all circuits known to the executor. Use a value up to a maximum of 65,535. The default value is equal to the value of the SEGMENT BUFFER SIZE, if specified; otherwise the default is 576.

COUNTER TIMER seconds

Specifies a timer whose expiration causes a node counter logging event.

DEFAULT ACCESS option

Assigns the default access to all nodes that do not have a specific node ACCESS entry in the volatile database. There are four options:

BOTH	Allows incoming and outgoing logical link connections. This is the default.
INCOMING	Allows logical link connections from the remote node.
NONE	Does not allow incoming or outgoing logical link connections to this node.
OUTGOING	Allows the local node to initiate connections to the remote node, but does not allow connections from the remote node.

If you have OPER privilege on the local system, you can override the default access restriction specified in this parameter.

DELAY FACTOR number

Specifies the number by which to multiply one-sixteenth of the estimated round trip delay to a node to set the retransmission timer to that node. Use a number up to a maximum of 255. If you do not set this parameter,the default value is 80.

DELAY WEIGHT number

Specifies the weight to apply to a new round-trip delay data point when updating the estimated round-trip delay to a node. Use a number in the range up to a maximum of 255. If you do not set this parameter, the default value is 5.

IDENTIFICATION id-string

Specifies a text string that describes the executor node (for example, "VMS Host System"). The string can be a maximum of 32 characters. If it contains blanks or tabs, you must enclose the string in quotation marks. If you do not set this parameter, the default value is DECnet–VAX V5.n VMSV5.n.

INACTIVITY TIMER seconds

Specifies the maximum duration of inactivity (no data in either direction) on a logical link before the node checks to see if the logical link still works. If you do not set this parameter, the default value is 60.

INCOMING PROXY option

Indicates whether proxy login requests present on incoming logical links are to be honored. There are two options for INCOMING PROXY:

DISABLED	Ignores all incoming proxy requests and instead relies exclusively on access control information supplied in the connect requests to validate the logical link.
ENABLED	Invokes the appropriate proxy, based on the source user, source node, and supplied access control information (if any). This is the default.

Note that proxy access characteristics established in the object database take preference over the proxy access characteristics established in the executor database.

INCOMING TIMER seconds

Specifies the maximum amount of elapsed time between the time a connection is received for a process and the time that process accepts or rejects the connection. For very busy systems, use a value in the range of 45 to 60 seconds. Otherwise use a value of 30 seconds. The default value is 45.

MAXIMUM ADDRESS number

Defines the largest node address and, consequently, the greatest number of nodes that can be addressed by the local node. Use as small a number as possible. The default value is 1023.

If you receive an "invalid value" error, you probably have not allocated enough nonpaged pool space. You must lower the MAXIMUM ADDRESS value or increase the VMS NPAGEDYN value. (Use the System Generation Utility to change the NPAGEDYN parameter.)

MAXIMUM AREA number

Applies only to an executor node whose type is AREA. Specifies the largest area number and, therefore, the greatest number of areas that can be known about by the executor node's Routing layer. You can specify a decimal value up to a maximum of 63. The default is 63.

MAXIMUM BROADCAST NONROUTERS number

Specifies the maximum total number of nonrouting nodes (end nodes) the executor node can have on its Ethernet circuits. Use a number up to a maximum of 65,535. The default value is 64.

MAXIMUM BROADCAST ROUTERS number

Specifies the maximum total number of routers the executor node can have on its Ethernet circuits. Use a number up to a maximum of 65,535. The default value is 32.

MAXIMUM BUFFERS number

Specifies the maximum number of buffers in the transmit buffer pool. DECnet normally allocates only what it needs. At minimum, use a value that is 15 times the square root of the number of lines. Increase this value if you experience congestion loss. The default value is 100.

MAXIMUM CIRCUITS number

Defines the maximum number of routing circuits that the local node can use. The number must be in the range 1 to 127. The default value is 16.

MAXIMUM COST number

Specifies the maximum total path cost allowed from the local node to any node. The path cost is the sum of the circuit costs along a path between two nodes. Use as small a number as possible in the range of 1 to 1022. The default is 1022.

MAXIMUM HOPS number

Specifies the maximum routing hops from the local node to any other reachable node. A hop is the logical distance over a circuit between two adjacent nodes. Use as small a number as possible in the range of 1 to 30, and be sure that this value is less than or equal to the MAXIMUM VISITS parameter. The default value is 30.

MAXIMUM LINKS number

Specifies the maximum logical link count for the local node. A reasonable range for most networks is 25 to 50. The maximum value for MAXIMUMLINKS is 3885. The default value of MAXIMUM LINKS is 32.

MAXIMUM PATH SPLITS number

Indicates the maximum number of equal cost paths to a given destination node among which the packet load may be split. The default value is 1.

MAXIMUM VISITS number

Specifies the maximum number of nodes a message can visit before it is received by the destination node. Use a number in the range of the value of the MAXIMUM HOPS parameter to 63. You should specify a number that is twice the MAXIMUM HOPS value. The default value is 63.

NAME node-name

Specifies the node name to be associated with the executor node identification. You can assign only one name to a node address or node identification.

NONPRIVILEGED item

Specifies nonprivileged inbound access control information for the node. Associate any of the following parameters with the NONPRIVILEGED parameter:

ACCOUNT account	Identifies the account for the default nonprivileged DECnet account on the executor node.
PASSWORD password	Identifies the password for the default nonprivileged DECnet account on the executor node.
USER user-id	Identifies the user name for the default nonprivileged DECnet account on the executor node.

OUTGOING PROXY option

Indicates whether proxy login may be used on outgoing connect requests. There are two options for OUTGOING PROXY.

DISABLED	Specifies that proxy login is not requested on any outgoing logical links.
ENABLED	Specifies that proxy login is requested on outgoing logical links. This is the default.

Note that proxy access characteristics established in the object database take preference over the proxy access characteristics established in the executor database.

OUTGOING TIMER seconds

Specifies the timeout value for the elapsed time between the moment a connection is requested and the moment that connection is acknowledged by the destination node. A value in the range of 30 to 60 seconds is recommended. The default is 45.

PATH SPLIT POLICY policy

Specifies the policy for equal cost path splitting of network traffic. There are two values for PATH SPLIT POLICY:

INTERIM	Specifies that traffic will be split over all equal cost paths while forcing packets for individual network sessions to follow the same paths in order to guarantee that packets will be received by the destination node in the correct order. The INTERIM value should be set if some of the nodes in the network do not support out-of-order packet caching. (DECnet–VAX Version 4.5 and lower DECnet–VAX versions do not support out-of-order packet caching.)
NORMAL	Specifies that all traffic will be split equally overall equal cost paths to a destination node. All destination nodes must support out-of-order packet caching (supported by DECnet–VAX Version 4.6 or higher);otherwise, network performance may suffer. This is the default.

PIPELINE QUOTA quota

Specifies the maximum number of bytes of nonpaged pool that DECnet will use for transmission over logical links. Use this parameter for multibuffering at the NSP level. The default value is 10000 bytes. For satellite communications, a value of 6000 or greater is recommended.

PRIVILEGED item

Specifies privileged inbound access control information for the node. Associate any of the following parameters with the PRIVILEGED parameter:

ACCOUNT account	Identifies the account for the default privileged DECnet account on the executor node.
PASSWORD password	Identifies the password for the default privileged DECnet account on the executor node.
USER user-id	Identifies the user name for the default privileged DECnet account on the executor node.

These parameters are not needed unless the PRIVILEGES parameter is used explicitly in the object database.

RETRANSMIT FACTOR number

Defines the maximum number of times any given message (except a connect initiate message) will be retransmitted before the logical link is disconnected. If you do not set this parameter, the default value is 10.

ROUTING TIMER seconds

Specifies the maximum amount of elapsed time before a routing update is forced on non-Ethernet circuits. The routing update produces a routing configuration message for each adjacent node. You can use a number up to a maximum of 65,535. If you do not set this parameter, the default value is 600.

SEGMENT BUFFER SIZE number

Specifies in bytes the maximum size of transmit buffers, thereby controlling the maximum size NSP message segment that can be transmitted.(This value is the maximum size message the End Communications layer can transmit; it does not include Data Link layer overhead.) Use a value up to a maximum of 65,535. The default value is equal to the value of BUFFER SIZE, if specified; otherwise, the default is 576.

The SEGMENT BUFFER SIZE is always less than or equal to the BUFFER SIZE. The two values are normally equal but may differ to permit the network manager to alter buffer sizes on all nodes without interruption of service.

STATE node-state

Specifies the operational state of the local node. There are four possible states:

OFF	Allows no new logical links, terminates existing links, and stops route-through traffic.
ON	Allows logical links.
RESTRICTED	Allows no new inbound links from other nodes.
SHUT	Allows no new logical links, does not destroy existing links, and goes to the OFF state when all logical links are disconnected.

If you have OPER privilege, you can override the state value specified in this parameter.

SUBADDRESSES range

Applies only to DLM circuits. Specifies a range of local DTE subaddresses that the Routing layer accepts as X.25 DLM calls. VAX P.S.I. routes all incoming X.25 calls within the specified subaddress range to the Routing layer to be handled as DLM calls.

TYPE node-type

Indicates the type of the executor node. There are three possible node types:

AREA
NONROUTING IV
ROUTING IV

The default depends upon the DECnet–VAX license registered. If the full function kit is installed, the default is ROUTING IV; if the end node kit is installed, the default (and only possible value) is NONROUTING IV.

A routing node has full routing capability. A nonrouting node (or end node) can deliver packets to or receive them from any node, but cannot route packets from other source nodes through to destination nodes.

An area node is a level 2 router that can route packets between areas.

Description

You can use the SET and DEFINE EXECUTOR commands to create or modify local node parameters, including certain parameters common to both the local and remote nodes. Refer to the *VSI OpenVMS DECnet Networking Manual* for a table of node parameters that indicates those used for the local node only and those common to the local and remote nodes.

Example

1. `NCP> SET EXECUTOR ADDRESS 2.11`

   This command sets the local node's address to 2.11.

2. `NCP> SET NODE 2.13 NAME CLUSTR...`
   `NCP> SET EXECUTOR ALIAS NODE CLUSTR`

   The SET NODE command establishes a node address 2.13 with the associated nodename CLUSTR. The SET EXECUTOR ALIAS NODE command then establishes the nodename CLUSTR as the alias node identifier.

3. `NCP> SET EXECUTOR ALIAS INCOMING ENABLED`

   This command specifies that incoming connect requests directed to the alias node identifier are permitted.

4. `NCP> SET EXECUTOR ADDRESS 11...`
   `NCP> SHOW EXECUTOR SUMMARY`
   `Node Volatile Summary as of 30-DEC-1989 15:40:20`
   `Executor node = 1.11 (BOSTON)`
   `State               = off`
   `Identification      = VMS HOST SYSTEM`

   This command sets the local node's address to 1.11. As the display indicates,the system default value is 1 if the *area-number* is not specified.

5. `NCP> SET EXECUTOR STATE ON`

   This command turns on the local node.

6. `NCP> SET EXECUTOR NAME BOSTON`

   This command sets the local node's name to BOSTON.

SET EXECUTOR NODE

SET EXECUTOR NODE — The SET EXECUTOR NODE command sets the default executor for all NCP commands. The executor is the node on which the network management listener (NML) runs to perform these commands. You cannot use the TELL prefix with the SET EXECUTOR NODE command.

Format

SET EXECUTOR [parameter [...]]

Command Parameter

NODE node-spec

Specifies a node name or address optionally followed by access control information as specified for VMS operating systems. Use one of the following formats:

```
node-idnode-id"user password account"
```

The *node-spec* may be a logical name. You can override access control in a logical name by specifying access control information explicitly in the command.

Associate any or all of the following optional access control parameters with this parameter:

ACCOUNT account	Identifies the user's account for access control verification at the designated executor node.
PASSWORD password	Identifies the user's password for access control verification at the designated executor node.
USER user-id	Identifies the user's identification for access control verification at the designated executor node.

Description

For this command, you must enter the command verb SET EXECUTOR followed by the NODE *node-spec* parameter. You can then enter the optional access control parameters in any order.

Also, if you have implemented area routing on your node and you specify a node address, you must use the area-number prefix with the node address. Refer to the following examples.

If the SET EXECUTOR NODE command fails, you must explicitly use the CLEAREXECUTOR NODE or the SET EXECUTOR NODE command with a different nodename or address; otherwise, NCP continues to try to establish a logical link to NML running on the first executor node named.

Example

1. NCP> SET EXECUTOR NODE 5.14...
   NCP> CLEAR EXECUTOR NODE

   The first command sets the executor to node 5.14. The second resets the executor to the local node.

2. NCP> SET EXECUTOR NODE 14"GRAY MARY"

   This command uses access control information to set the executor node to node 1.14. This example assumes a default area number of 1 for the executor node.

3. NCP> SET EXECUTOR NODE TRNTO USER GRAY PASSWORD MARY

   This command uses an alternate access control format to set the executor to node TRNTO.

4. `NCP> SET EXECUTOR NODE 0"GRAY MARY"`

   This command uses access control information to cause NML to run under user GRAY's current privileges instead of the privileges of the NCP user.

5. `$ DEFINE NODE$MY   TRNTO"""GRAY MARY"""`
   `$ RUN SYS$SYSTEM:NCP`
   `NCP> SET EXECUTOR NODE NODE$MY`

   The first command defines a logical name (NODE$MY) for the given *node-spec*.When running NCP, you can use this logical name with the SET EXECUTOR NODE command. Note that you must use three sets of quotation marks to delimit access control information (if supplied) in the *node-spec* for this example.

SET KNOWN PROXIES ALL

SET KNOWN PROXIES ALL — The SET KNOWN PROXIES ALL command clears the contents of the volatile proxy database and rebuilds the volatile proxy database from the contents of the permanent proxy database.

Format

SET   [proxy-component ALL]

Proxy Component

KNOWN PROXIES

Proxy Component

KNOWN PROXIES

Indicates that all proxy values are to be updated.

Example

1. `NCP> SET KNOWN PROXIES ALL`

   This command rebuilds the volatile database of proxy values from the permanent database.

SET LINE ALL

SET LINE ALL — The SET LINE ALL command updates the volatile database on the local node or DTE with all the line parameters stored for a particular line in the permanent database on the local node or DTE.

Format

SET   [line-component ALL]

Line Components

LINE

KNOWN LINES

Line Components

LINE line-id

Identifies the line whose parameters are to be updated.

KNOWN LINES

Indicates that all parameters for all known lines are to be updated.

Example

1. `NCP> SET KNOWN LINES ALL`

   This command loads all permanent database entries stored for all known lines into the volatile database.

SET/DEFINE LINE

SET/DEFINE LINE — The SET LINE command creates or modifies line parameters in the volatile database on the local node. The DEFINE LINE command creates or modifies line parameters in the permanent database on the local node. A line must be in the OFF state in order for all but the COUNTER TIMER, SERVICE TIMER, and STATE parameters to be changed.

Format

SET   [line-component parameter [...]]

DEFINE   [line-component parameter [...]]

Line Components

LINE
KNOWN LINES

Line Components

LINE line-id

Identifies the line for which specified parameters are to be created or modified in the database.

KNOWN LINES

Indicates that the specified parameters for all known lines are to be created or modified in the database.

Command Parameters

BUFFER SIZE number

Specifies in bytes the size of receive buffers for the specified line, thereby controlling the maximum size of NSP message segments that can be received from or forwarded to an adjacent node that has

accepted the line buffer size. Use a value up to a maximum of 65,535. For Ethernet lines, a default value of 1498 bytes is provided. For all other types of line, the default is the executor BUFFER SIZE value (as specified in the SET EXECUTOR command).

You can use the line parameter BUFFER SIZE to increase the size of NSP messages for logical links over this line.

CLOCK clock-mode

Applies only to synchronous DDCMP lines. Specifies the hardware clock mode for the line. There are two values for *clock-mode*:

EXTERNAL	For normal clock operating mode. The clock signal is supplied externally to the controller.
INTERNAL	For use of the clock in test mode. Setting this value causes the line device to supply a clock signal that will allow all transmitted messages to be looped back from outside the device. Note that,in order to use this parameter, the operator may have to connect a loopback plug in place of the normal line.

CONTROLLER mode

Specifies the controller mode for the line. There are two possible modes:

LOOPBACK	Internal device loopback mode
NORMAL	Normal operating mode, which is the default

COUNTER TIMER seconds

Specifies a timer whose expiration causes a line counter logging event. Specify a decimal integer up to a maximum of 65,535.

DEAD TIMER milliseconds

Applies only to DDCMP CONTROL lines. Specifies the number of milliseconds between polls of one of the sets of dead tributaries. The *milliseconds* value must be a decimal integer up to a maximum of 65,535. The default is 10,000 (10 seconds).

DELAY TIMER milliseconds

Applies only to DDCMP CONTROL lines. Defines the minimum number of milliseconds to delay between polls. This timer limits the effect of a very fast control station on slower tributaries. The *milliseconds* value must be a decimal integer up to a maximum of 65,535. If you do not set this parameter, there is no delay.

DUPLEX mode

Does not apply to Ethernet lines. Specifies the hardware duplex mode of the line. There are two possible modes:

FULL	Full-duplex (default)
HALF	Half-duplex

HANGUP option

Applies only to asynchronous DDCMP lines. Indicates whether the modem signals are dropped when the line is shut down. There are two possible options:

DISABLED	Indicates that modem signals should not be dropped when the line is shut down. This is the default for static asynchronous DDCMP lines.
ENABLED	Indicates that modem signals should be dropped when the line is shut down.

This parameter is supplied automatically for dynamic asynchronous DDCMP lines. The default is HANGUP ENABLED if the /HANGUP qualifier was specified for the DCL command SET TERMINAL, and HANGUP DISABLED if /NOHANGUP was specified.

HOLDBACK TIMER milliseconds

Applies only to X.25 lines. Specifies the maximum delay before the X.25 software acknowledges successful receipt of data. If you do not specify this parameter, then the acknowledgment is sent immediately. If you specify a holdback timer, performance may be improved by allowing the acknowledgment to be included with a subsequent data message transmitted. This parameter cannot be set to a value larger than one half the value of the RETRANSMIT TIMER parameter.

INTERFACE interface-code

Applies only to X.25 lines using the ISO8208 profile. Specifies the way in which the X.25 level 2 software acts when it uses the line. There are two values for the *interface-code*:

DCE	Specifies that the software operates as a DCE.
DTE	Specifies that the software operates as a DTE.

The default code is DTE.

LINE SPEED number

Applies only to asynchronous DDCMP lines. Specifies the speed of the line in baud. This parameter must be set to the same value on both sides of an asynchronous DDCMP connection. It is specified automatically for dynamic asynchronous DDCMP lines. If not specified, the value of this parameter is equal to the current speed of the line.

MAXIMUM BLOCK count

Applies only to X.25 lines. Specifies the maximum size in bytes of the frame for the line. This value must be at least 5 bytes larger than the maximum packet size you specify in the SET CIRCUIT (for PVCs) or SET MODULE X25-PROTOCOL (for SVCs) command. Specify a value in the range 21 to 4103. By default, the size of the frame takes the value specified by the PROFILE parameter of SET MODULE X25-PROTOCOL commands. See the *Public Network Information Guide* manual for the network value of this parameter.

MAXIMUM RETRANSMITS count

Applies only to X.25 lines. Specifies the maximum number of retransmissions of a frame on the specified line. Specify a value up to a maximum of 255. By default, the retransmission value takes the value specified by the PROFILE parameter of the SET MODULE X25-PROTOCOL command. See the *Public Network Information Guide* for the network value of this parameter.

MAXIMUM WINDOW count

Applies only to X.25 lines. Specifies the maximum number of frames for which outstanding acknowledgments are allowed. Specify a value up to a maximum of 127. By default, the outstanding frame value takes the value specified by the PROFILE parameter of the SET MODULE X25-PROTOCOL command. See the *Public Network Information Guide* for the network value of this parameter.

MICROCODE DUMP filespec

Applies only to X.25 lines. Dumps the microcode of the specified KMS11, KMS1P, or KMV1A device to the indicated file. By default, the output file takes the following format:

```
SYS$ERRORLOG:fileid.DMP
```

The KMS/KMV Dump Analyzer (PSIKDA) is the utility program used to process the dump file created by the MICROCODE DUMP parameter. Refer to the *VAX P.S.I. Problem Solving Guide* for more information about PSIKDA.

NETWORK network-name

Applies only to X.25 lines. Specifies the network to which the line is connected. If only one network is set up, that network is the default. Otherwise, the NETWORK parameter is mandatory.

[PROTOCOL protocol-name]

Defines the Data Link protocol to be used on this line. The following values can be used for *protocol-name*:

DDCMP CONTROL	Specifies this line as a multipoint control station. You can specify multiple circuits for CONTROL lines, but each circuit must have a unique physical tributary address.
DDCMP DMC	Specifies that this line is in DMC emulator mode. DMC is similar to POINT, except that DMC uses an older version of DDCMP (Version 3.2). This protocol should be set for the local line when the remote line is a DMC. Note that this protocol is valid only when a DMP11 or DMV11 is being used.
DDCMP POINT	Defines this line as one end of a point-to-point DDCMP connection. You may specify only one circuit per POINT line.
DDCMP TRIBUTARY	Specifies that this line is a tributary end of a DDCMP multipoint group. You may specify only one circuit per TRIBUTARY line.
ETHERNET	Specifies that this line uses the Ethernet protocol.

Default line protocols based on line names are as follows:

BNA	ETHERNET
CI	No protocol specified
DMB	DDCMP POINT
DMC/DMR	DDCMP POINT
DMF	DDCMP POINT
DMP/DMV	DDCMP POINT

DPV	LAPB
QNA	ETHERNET
SVA	ETHERNET
UNA	ETHERNET

RECEIVE BUFFERS number

Specifies the length of the line's receive queue. Use a value in the range 1 to 32. A value in the range 2 to 4 is adequate for line speeds of less than 56 kilobits/second. Line speeds of 1 megabit/second may require eight or more buffers depending on the observed error rate. For X.25 lines, specify a value in the range 2 to 32. The default is 4, which is normally adequate for DUP, DPV, and DMF lines. However, note that KMX, KMY,and KMV lines can use 8 buffers.

[RETRANSMIT TIMER milliseconds]

Does not apply to Ethernet lines. For a DDCMP multipoint line,specifies the maximum amount of time allowed to elapse before a retransmission is necessary. This is the amount of time a control station will wait for a tributary to respond. The number of milliseconds must be a decimal integer up to a maximum of 65,535. The default value is the network default.

For an X.25 line, this parameter specifies the time before a frame is retransmitted. By default, the time takes the value specified in the PROFILE parameter of the SET MODULE X25-PROTOCOL command. See the *Public Network Information Guide* for the network value of this parameter. The RETRANSMIT TIMER cannot be set to a value smaller than twice the value of the HOLDBACK TIMER parameter (if specified).

[SCHEDULING TIMER milliseconds]

Applies only to DDCMP CONTROL lines. Specifies the number of milliseconds between recalculation of tributary polling priorities. The number of milliseconds must be a decimal integer up to a maximum of 65,535. The default is 50.

[SERVICE TIMER milliseconds]

Does not apply to X.25 lines. Specifies the maximum amount of time allowed to elapse before a receive request completes while performing service operations on a line. The default value is 4000 (4 seconds).

[STATE line-state]

Specifies the line's operational state. The possible states include the following:

OFF	The line is not in use.
ON	The line is available for normal use or service functions.
SERVICE	The line is available for active service functions. Applies only to X.25 lines.

STREAM TIMER milliseconds

Applies only to DDCMP CONTROL lines. Defines the number of milliseconds that a tributary or half-duplex remote station is allowed to hold the line. The number of milliseconds must be a decimal integer up to a maximum of 65,535. The default is 1000 (1 second).

SWITCH option

Applies only to asynchronous DDCMP lines. Forces the line currently being used as a DECnet asynchronous communications line to be converted back to a terminal line. There are two values for *option*:

DISABLED	The line is not switched to a terminal line. This is the default for static lines.
ENABLED	The line is switched to a terminal line after it is disconnected from the network (when the channel to the network is deassigned). This is the default for dynamic lines.

TRANSMIT PIPELINE count

Applies only to DMR11 lines. Specifies the maximum number of DDCMP messages for which outstanding acknowledgments are allowed. Specify a value in the range 1 to 32. By default, the value for outstanding DDCMP messages is 7. To avoid excessive use of system memory, do not arbitrarily set this value higher than necessary. Refer to the *VSI OpenVMS DECnet Networking Manual* for the procedure for determining an optimum TRANSMIT PIPELINE value.

Description

Note that no protocol is specified for a CI line. The CI uses its own private protocols for communication between nodes. If you plan to run DECnet over a CI, you must install the driver CNDRIVER. To do this, add the following lines to the LOADNET.COM command procedure in SYS$MANAGER:

```
$ RUN SYS$SYSTEM:SYSGENCONNECT CNA0/NOADAPTER
```

Refer to the *VSI OpenVMS DECnet Networking Manual* for a table of line parameters grouped according to line type.

Example

1. `NCP> SET LINE UNA-0 STATE ON`

   This command sets Ethernet line UNA–0 to the ON state.

2. `NCP> SET LINE DMC-0 DUPLEX FULL STATE ON`

   This command sets line DMC–0 to the ON state in full duplex mode.

3. `NCP> SET LINE DMC-0 PROTOCOL DDCMP POINT`

   This command sets the line protocol to DDCMP POINT for line DMC–0.

4. ```
   NCP> SET LINE DUP-1 -
   _ RETRANSMIT TIMER 500 MAXIMUM RETRANSMITS 10 -
   _ MAXIMUM BLOCK 133 MAXIMUM WINDOW 2 STATE ON
   ```

   This command sets up a line called DUP–1 that uses the LAPB protocol and has an associated DUP11–DA synchronous line interface with controller number 1. If a frame is not acknowledged within 500 milliseconds, it is retransmitted, and this operation is to be performed a maximum of 10 times. The maximum frame size is 133 bytes, and the maximum number of frames for which outstanding acknowledgments are allowed is 2. The line is operational.

5. `NCP> SET LINE KMX-0-0 MICROCODE DUMP BARRY`

This command dumps the microcode of the KMX to the file SYS$ERRORLOG:BARRY.DMP.

6. `NCP> SET LINE KMV-0 NETWORK PSS STATE ON`

    This command sets up the line KMV-0 to the PSS network.

7. `NCP> SET LINE DMF-* CONTROLLER LOOPBACK`

    This command sets the controller mode to loopback for all DMF lines.

SET LOGGING ALL

SET LOGGING ALL — The SET LOGGING ALL command updates the volatile copy of the database on the executor node with all the logging parameters stored for the logging component in the permanent database on the executor node. For VAX P.S.I., the SET LOGGING ALL command updates the volatile database at the local DTE with the logging parameters for the logging component in the permanent database at the local DTE.

Format

SET [logging-component ALL]

Logging Components

KNOWN LOGGING

Indicates that all parameters for all known logging sinks are to be updated.

LOGGING CONSOLE

Indicates that all parameters for the logging console are to be updated.

LOGGING FILE

Indicates that all parameters for the logging file are to be updated.

LOGGING MONITOR

Indicates that all parameters for the logging monitor are to be updated.

Example

1. `NCP> SET LOGGING CONSOLE ALL`

    This command loads all permanent database entries stored for the logging console into the volatile database.

SET/DEFINE LOGGING EVENTS

SET/DEFINE LOGGING EVENTS — The SET LOGGING EVENTS command creates or modifies source-related logging parameters in the volatile database on the local node. The DEFINELOGGING EVENTS command creates or modifies source-related logging parameters in the permanent database on the local node. The logging state (in the SET/DEFINE LOGGING STATE command) must be set to

ON for all logging. Source-related and sink-related parameters are mutually exclusive; you cannot use parameters from both categories in a single command.

Format

SET   [logging-component parameter [...]]

DEFINE   [logging-component parameter [...]]

Logging Components

KNOWN LOGGING
LOGGING FILE
LOGGING CONSOLE
LOGGING MONITOR

Logging Components

KNOWN LOGGING

Indicates that the specified parameters for all known logging are to be created or modified in the database.

LOGGING CONSOLE

Indicates that the specified parameters for the logging console are to be created or modified in the database.

LOGGING FILE

Indicates that the specified parameters for the logging file are to be created or modified in the database.

LOGGING MONITOR

Indicates that the specified parameters for the logging monitor are to be created or modified in the database.

Command Parameters

[EVENTS event-list]

Indicates a specific event or all known events to be logged. The EVENTS parameter may be followed by one or more of the following source-related parameters:

CIRCUIT circuit-id	Identifies the circuit from which events are logged.
LINE line-id	Identifies the line from which events are logged.
MODULE name	Indicates that the logging parameters for the specified source are to be created or modified in the database. Possible module names are as follows: X25-PROTOCOL X25-SERVER X29-SERVER X25-ACCESS

NODE node-id	Identifies the node from which events are logged.	
SINK node	Identifies the node that is to receive events. There are two possibilities:	
	EXECUTOR	Identifies the local node (default).
	NODE node-id	Identifies a node in the network.

Description

For a table of the source-related parameters used with the SET/DEFINE LOGGING EVENTS command, refer to the *VSI OpenVMS DECnet Networking Manual*. For a description of event-lists, refer to Section 1.4. For a summary of event class and types and information about specific events that VMS operating systems will log, refer to Appendix C.

Example

1. NCP> SET LOGGING MONITOR LINE DMC–1 KNOWN EVENTS

   This command causes all events for line DMC–1 to be logged on the console.

2. NCP> SET LOGGING CONSOLE KNOWN EVENTS SINK NODE TRNTO

   This command causes all events generated locally to be logged to the logging console on remote node TRNTO.

3. NCP> SET LOGGING CONSOLE EVENTS 4.* SINK EXECUTOR

   This command causes all class 4 events to be logged on the console of the executor node.

SET/DEFINE LOGGING STATE

SET/DEFINE LOGGING STATE — The SET LOGGING STATE command creates or modifies sink-related logging parameters in the volatile database on the local node. The DEFINELOGGING STATE command creates or modifies sink-related logging parameters in the permanent database on the local node. The STATE parameter is required for all logging.

Format

SET   [logging-component parameter [...]]

DEFINE   [logging-component parameter [...]]

Logging Components

KNOWN LOGGING
LOGGING FILE
LOGGING CONSOLE
LOGGING MONITOR

Logging Components

KNOWN LOGGING

Indicates that the specified parameters for all known logging are to be created or modified in the database.

LOGGING CONSOLE

Indicates that the specified parameters for the logging console are to be created or modified in the database.

LOGGING FILE

Indicates that the specified parameters for the logging file are to be created or modified in the database.

LOGGING MONITOR

Indicates that the specified parameters for the logging monitor are to be created or modified in the database.

Command Parameters

NAME sink-name

Identifies the name of a console, file, or monitor program to which events will be logged. For the monitor sink, all network operator terminals (enabled by the command REPLY/ENABLE=NETWORK) receive formatted events. You can change this parameter only if the sink is in the OFF state.

STATE sink-state

Specifies the operational state of the sink. There are three possible states:

HOLD	The sink is temporarily unavailable and events are queued.
OFF	The sink is unavailable for receiving events; they are discarded.
ON	The sink is available for receiving events.

Description

Refer to the *VSI OpenVMS DECnet Networking Manual* for a table of sink-related logging parameters and their functions.

Example

1. `NCP> SET LOGGING CONSOLE NAME SYS$MANAGER:EVENTS.LOG`

   This command establishes a logging file, EVENTS.LOG, on the local node.

2. `NCP> SET LOGGING MONITOR STATE ON`

   This command turns on logging to OPCOM, the default VMS monitor. All operator terminals that were previously declared as network operator terminals (with REPLY/ENABLE=NETWORK) will now receive all formatted events.

SET/DEFINE MODULE CONFIGURATOR

SET/DEFINE MODULE CONFIGURATOR — The SET MODULE CONFIGURATOR command creates or modifies the parameters of the Ethernet configurator module component in the volatile

database at the local node on an Ethernet circuit. The DEFINE MODULE CONFIGURATOR command creates or modifies the parameters of the configurator module component in the permanent database. You must specify a qualifier following the CONFIGURATOR command verb. Note that the word MODULE is optional. The configurator module constructs a list of systems active on the Ethernet circuits specified.

Format

SET MODULE CONFIGURATOR   [[qualifier] parameter]

DEFINE MODULE CONFIGURATOR   [[qualifier] parameter]

Qualifiers

CIRCUIT
KNOWN CIRCUITS

Qualifiers

CIRCUIT circuit-id

Applies only to Ethernet circuits. Specifies that configurator information on the circuit identified by *circuit-id* is to be stored in the database.

KNOWN CIRCUITS

Applies only to Ethernet circuits. Specifies that configurator information on all known Ethernet circuits is to be stored in the database.

Command Parameters

ALL

Copies configurator module information from the permanent database into the database at the local node.

SURVEILLANCE control

Indicates whether a list of active systems is to be kept for the Ethernet circuit or circuits specified. There are two control values:

| DISABLED | The list is not to be kept. This is the default value. |
| ENABLED | The list is to be kept. |

Example

1. NCP> SET MODULE CONFIGURATOR KNOWN CIRCUITS -
   _ SURVEILLANCE ENABLED

   This command specifies that a list of active systems on all known Ethernet circuits is to be kept.

2. NCP> SET CONFIGURATOR CIRCUIT UNA-0 SURVEILLANCE ENABLED

This command specifies that a list of active systems on Ethernet circuit UNA–0 is to be kept.

3. `NCP> SET MODULE CONFIGURATOR ALL`

This command loads all permanent database entries for the configurator module into the volatile database at the local node.

SET/DEFINE MODULE X25-ACCESS

SET/DEFINE MODULE X25-ACCESS — The SET MODULE X25-ACCESS and DEFINE MODULE X25-ACCESS commands associate the host node with the specified multihost connector node for the purpose of accessing a specified X.25 network. Note that the word MODULE is optional.

Format

SET MODULE X25-ACCESS [qualifier parameter [...]]

DEFINE MODULE X25-ACCESS [qualifier parameter [...]]

Qualifiers

NETWORK
KNOWN NETWORKS

Qualifiers

NETWORK network-name
KNOWN NETWORKS

Identifies the name of a network or all known networks to be accessed by the host node. The network name must be the same network name specified on the multihost node to which the access node connects. After you specify the network qualifier, you must include one or more of the following access control parameters:

ACCOUNT account	Identifies the user account for use when connecting to the multihost connector node. Specify a string from 1 to 39 characters. By default, no account is used.
ALL	Copies the access parameters from the permanent database into the volatile database. Use this parameter only with the SET MODULE X25-ACCESS command.
NODE node-id	Identifies a multihost connector node connected to the PSDN that you want to access. When you specify an X25-ACCESS network for the first time, this parameter is mandatory.
PASSWORD password	Specifies the password for use by the access routines when connecting to the multihost connector node. Specify a string from 1 to 39 characters. The default is that no password is used.
USER user-id	Specifies the user identification for use by the access routines in connecting to the multihost connector node. Specify a string from 1 to 16 characters. The default is that no user identification is used.

Example

1. `NCP> SET MODULE X25-ACCESS NETWORK TELENET1 NODE FRED`
   `NCP> SET MODULE X25-ACCESS NETWORK PSS1 NODE GINGER`

   These two commands allow users on the host node to access the network called TELENET1 through node FRED, and the network called PSS1 through node GINGER.

2. `NCP> SET MODULE X25-ACCESS KNOWN NETWORKS ALL`

   This command loads all permanent database entries stored for the X25-ACCESS database into the volatile database at the local node.

SET/DEFINE MODULE X25-PROTOCOL

SET/DEFINE MODULE X25-PROTOCOL — The SET MODULE X25-PROTOCOL command creates or modifies the parameters of the protocol module component in the volatile database. The DEFINE MODULEX25-PROTOCOL command creates or modifies the parameters of the protocol module component in the permanent database. Use separate SET MODULE X25-PROTOCOL commands to specify a DTE with its associated parameters, a group with its associated parameters, and the network profile. The X25-PROTOCOL module contains records that identify the packets witching data networks (PSDNs) your DTE is connected to, associate your local DTE (and possibly a group name) with VAX P.S.I., and control the transmission of packets. Note that the word MODULE is optional.

Format

SET MODULE X25-PROTOCOL   [qualifier [...] parameter [...]]

DEFINE MODULE X25-PROTOCOL   [qualifier [...] parameter [...]]

Qualifiers

DTE
KNOWN DTES
GROUP
KNOWN GROUPS
NETWORK
KNOWN NETWORKS

Qualifiers

DTE dte-address
KNOWN DTES

Identifies a local DTE or specifies all known DTEs. See the *Public Network Information Guide* manual for the format of the address. If you specify the DTE qualifier, you must associate with it either of the following parameters.

NETWORK network-name	Selects the DTE(s) specified on a particular network.
KNOWN NETWORKS	Selects the DTE(s) on all known networks.

If you have only one network set up, that network is the default for SET commands. For DEFINE commands the NETWORK parameter is mandatory. If you have more than one network set up, the

NETWORK parameter is mandatory. In either case, you must follow the DTE qualifier with one or more of the following parameters:

ALL	Copies DTE parameters from the permanent database into the volatile database. Used only with the SET MODULE X25-PROTOCOL command.
CALL TIMER seconds	Specifies a timer that starts when a request to set up an X.25 virtual circuit is transmitted, and whose termination clears the request if no response has been received. Specify a value up to a maximum of 255. By default, the timer takes the network value specified by the PROFILE parameter. See the *Public Network Information Guide* for the network value of this parameter.
CHANNELS list	Specifies a list of logical channel numbers (LCNs) reserved for outgoing calls. The *list* value is one or more LCNs. Separate multiple channel numbers with hyphens to indicate ranges and with commas to indicate individual numbers. Specify a value up to a maximum of 4095 for each number in the list. If the DTE is to make outgoing calls, this parameter is mandatory when you specify a DTE for the first time. The LCN range at the DTE is defined by PSDN authorities at subscription time.
CLEAR TIMER seconds	Specifies a timer that starts when a request to clear an X.25 virtual circuit is transmitted, and whose termination retransmits the clear request if no response has been received. Specify a value up to a maximum of 255. By default, the timer takes the network value specified by the PROFILE parameter. See the *Public Network Information Guide* for the network value of this parameter.
COUNTER TIMER seconds	Sets a timer whose expiration causes a DTE counter logging event. Specify a decimal integer up to a maximum of 65,535.
DEFAULT DATA bytes	Specifies the default packet size for X.25 virtual circuits. This value must be at least 5 bytes less than the MAXIMUMBLOCK value you specify in the SET LINE command and must be a power of 2. Specify a value in the range 16 to 4096 bytes. By default, packet size takes the network value specified by the PROFILE parameter. See the *Public Network Information Guide* for the network value of this parameter.
DEFAULT WINDOW count	Specifies the default window size for X.25 virtual circuits, that is, the maximum number of packets for which outstanding acknowledgments are allowed. Specify a value up to a maximum of 127. By default,window size takes the network value specified by the PROFILE parameter. See the *Public Network Information Guide* for the network value of this parameter.
INTERFACE interface-code	Specifies the way in which the processor operates. This parameter can be used only with the ISO8208 profile. The parameter can take one of the following values:

	DTE	The processor operates as a DTE.
	DCE	The processor operates as a DCE.
	NEGOTIATED	The processor can operate either as a DTE or DCE. The restart protocol

	determines the appropriate method of operation.
	The default *interface-code* is DTE.
INTERRUPT TIMER seconds	This is the interrupt timer (ISO T26 timer). This parameter can be used only with the ISO8208 profile. Specify a value up to a maximum of 255. If you do not specify a value, there is no time limit on interrupt acknowledgments.
LINE line-id	Identifies the line associated with the DTE. Specify aline name in the format *dev-c[-u]*. This parameter is mandatory when you specify a DTE for the first time.
MAXIMUM CIRCUITS count	Specifies the maximum number of virtual circuits for the DTE. By default, the count takes the network value specified by the PROFILE parameter. See the *Public Network Information Guide* for the network value of this parameter.
MAXIMUM CLEARS count	Specifies the maximum number of attempts to clear an X.25 virtual circuit. Specify a value up to a maximum of 255. By default, the count takes the network value specified by the PROFILE parameter. See the *Public Network Information Guide* for the network value of this parameter.
MAXIMUM DATA bytes	Specifies the maximum packet size for X.25 virtual circuits. This value must be at least 5 bytes less than the MAXIMUMBLOCK value you specify in the SET LINE command and must be a power of 2. Specify a value in the range 16 to 4096 bytes. By default, packet size takes he network value specified by the PROFILE parameter. See the *Public Network Information Guide* for the network value of this parameter.
MAXIMUM RESETS count	Specifies the maximum number of attempts to reset an X.25 virtual circuit. If this count is exceeded, the virtual circuit will be cleared. Specify a value up to a maximum of 255. By default, the count takes the network value specified by the PROFILE parameter. See the *Public Network Information Guide* for the network value of this parameter.
MAXIMUM RESTARTS count	Specifies the maximum number of attempts to restart an X.25 virtual circuit. Specify a value up to a maximum of 255. By default, the count takes the network value specified by the PROFILE parameter. See the *Public Network Information Guide* for the network value of this parameter.
MAXIMUM WINDOW count	Specifies the window size allowed for X.25 virtual circuits, that is, the maximum number of packets for which outstanding acknowledgments are allowed. Specify a value up to a maximum of 127. By default,the window size takes the network value specified by the PROFILE parameter. See the *Public Network Information Guide* for the network value of this parameter.
RESET TIMER seconds	Specifies a timer that starts when a reset is transmitted, and whose termination retransmits the reset if no response has been received. Specify a value up to a maximum of 255. By default, the timer takes the network value specified by the PROFILE parameter. See the *Public Network Information Guide* for the network value of this parameter.

RESTART TIMER seconds	Specifies a timer that starts when a restart is transmitted, and whose termination retransmits the restart if no response has been received. Specify a value up to a maximum of 255. By default, the timer takes the network value specified by the PROFILE parameter. See the *Public Network Information Guide* for the network value of this parameter.	
STATE dte-state	Specifies the operational state of the DTE. This parameter takes one of the following values:	
	ON	The DTE is available for normal use.
	OFF	The DTE is not in use. This is the default state.
	SHUT	The DTE is to be closed down but only when all present activity has ceased. Used only with the SET MODULE X25-PROTOCOL command.

GROUP group-name
KNOWN GROUPS

Identifies a closed user group (CUG) or bilateral closed user group(BCUG) or specifies all known groups. Each group specified should have a unique *group-name*, from 2 to 16 characters long. If you specify the GROUP qualifier, you must follow it with one or more of the following parameters:

ALL	Copies the group parameters from the permanent database into the volatile database. Used only with the SET MODULEX25-PROTOCOL command.
DTE dte-address	Identifies the local DTE associated with the group name. This parameter is mandatory when you are creating a group.
NETWORK network-name	Identifies a network to which the DTE is connected. This parameter is mandatory when you are creating a group.
NUMBER group-number	Specifies the CUG or BCUG number. You can omit leading zeros. This parameter is mandatory when you are associating a DTE with a group.
TYPE BILATERAL	Specifies the user group as a BCUG. This parameter is mandatory when you are associating a DTE with a BCUG. It is omitted if the user group is a CUG.

NETWORK network-name
KNOWN NETWORKS

Identifies a network or specifies all known networks. If you specify the NETWORK qualifier, you must follow it with one of the following parameters.

ALL	Copies the profile name from the permanent database to the volatile database. Use only with the SET MODULE X25-PROTOCOL command.
PROFILE profile-name	Specifies a profile to be used on the network. The *profile-name* is a character string of up to 32 characters. This parameter is mandatory. Refer to the *Public Network Information Guide* for the valid *profile-name*.

Example

1. `NCP> SET MODULE X25-PROTOCOL DTE 123789456 NETWORK TELENET -`
   `_ CHANNELS 20-10,3,9 LINE DUP-0 MAXIMUM CIRCUITS 400 STATE ON`

   This command creates a record for your local DTE (DTE address 123789456) and associates a line DUP–0 and a set of logical channels with this DTE.

2. `NCP> SET MODULE X25-PROTOCOL NETWORK NET1 PROFILE PSDNC`

   This command specifies that you are going to use PSDNC, and that you are going to connect to this PSDN through the network NET1.

3. `NCP> SET MODULE X25-PROTOCOL GROUP ESECUG DTE 123789456 -`
   `_ NETWORK NET1 NUMBER 12`

   This command specifies that your DTE is a member of closed user group ESECUG with group number 12.

SET/DEFINE MODULE X25-SERVER/X29-SERVER

SET/DEFINE MODULE X25-SERVER/X29-SERVER — The SET MODULE X25-SERVER and SET MODULE X29-SERVER commands create or modify the parameters of the X.25 or X.29 call handler in the volatile database. The DEFINE MODULE X25-SERVER and DEFINEMODULE X29-SERVER commands create or modify the parameters of the X.25 or X.29 call handler in the permanent database. Use separate SET MODULE X25-SERVER or DEFINE MODULE X29-SERVER commands to specify the module parameters and the destinations. The server modules contain records that identify and specify parameters for a destination, specify the maximum number of circuits that each module (that is, all destinations for a particular module) may have, and specify the state of the module. Note that the word MODULE is optional.

Format

SET MODULE X25-SERVER [[qualifier] parameter [...]]

SET MODULE X29-SERVER [[qualifier] parameter [...]]

DEFINE MODULE X25-SERVER [[qualifier] parameter [...]]

DEFINE MODULE X29-SERVER [[qualifier] parameter [...]]

Qualifiers

DESTINATION
KNOWN DESTINATIONS

Qualifiers

DESTINATION dest-name
KNOWN DESTINATIONS

Identifies a destination or specifies all known destinations. If you specify the destination qualifier, you must follow it with one or more of the following parameters:

ACCOUNT account	Identifies the user account used in connecting to a host node for incoming calls to a destination. Specify a string from 1 to 39 characters. By default, no account is used.
ALL	Copies the destination parameters from the permanent database into the volatile database. Use only with the SET MODULE X25-SERVER or the SET MODULE X29-SERVER command.
CALL MASK hex-value	Specifies the mask applied to the incoming call data before it is tested. By default, no value is used.
CALL VALUE hex-value	Specifies the string used to test the incoming call data. By default, no value is used.
CALLED ADDRESS dte-address	For redirected calls, identifies the DTE that was originally called. The parameter value is used to select a destination for a redirected call. By default, no called DTE is used.
EXTENSION MASK hex-string	Provides a mask for the called address extension facility. Using logical AND, the mask is merged with the called address extension in the incoming call and the result is compared with the extension value. By default, no extension mask is used.
EXTENSION VALUE hex-string	Provides a value for the called address extension facility. The value is compared with the address extension in the incoming call (after merging as described for EXTENSION MASK). By default, no extension mask is used.
GROUP group-name	Identifies a closed user group (CUG) or bilateral closed user group (BCUG). The *group-name* is an ID string. By default, no group name is used.
INCOMING ADDRESS	Identifies the address of the DTE receiving the call (as specified in the call packet).
NETWORK network-name	Specifies the network for which calls are accepted. By default, no network name is used.
NODE node-id	Identifies the P.S.I. Access node that is to be the destination of the incoming X.25 call. Specify this parameter if the executor node is a VMS node with VAX P.S.I. software in multihost mode installed, serving as a connector node. This parameter is mandatory when you specify this destination for the first time. Used only with the SETMODULE X25-SERVER command or the DEFINE MODULE X25-SERVER command.
OBJECT object-id	Identifies the object used in connecting to a host node for incoming calls to a destination. The *object-id* is an ID string. If the object name looks like a number, enclose the string in quotation marks. This parameter is mandatory when you specify a destination for the first time.
PASSWORD password	Specifies the password used in connecting to a host node for incoming calls to a destination. Specify a string from 1 to 39 characters. The default is that no password is used.
PRIORITY priority	Specifies the priority of the destination record and is used to select one of a set of destinations for which the incoming call may be valid. Specify a value up to a maximum of 255, where 255 is the highest priority. By default, *priority* takes a value of 0.

RECEIVING DTE dte-address	Identifies the local DTE. It is used to select a destination for calls received on the local DTE. By default, no receiving DTE is used.	
REDIRECT REASON reason-code	Identifies a reason for redirecting an incoming call. The *reason-code* can take one of the following values:	
	BUSY	The original DTE was busy and could not accept any more calls.
	OUT OF ORDER	The original DTE was out of order.
	SYSTEMATIC	All calls to the original DTE are automatically rerouted.
	The *reason-code* is used to select a destination for are directed call. By default, no *reason-code* is used.	
SENDING ADDRESS dte-address	Identifies the address of the remote DTE that originated a call. By default, no sending DTE is used.	
SUBADDRESSES range	Specifies the range of local DTE subaddresses that identify the destination for the incoming call. The *range* consists of one or two subaddresses. The subaddress is a decimal integer up to a maximum of 9999. See the *Public Network Information Guide* for the maximum length of the subaddress on your network. Separate two subaddresses with a single hyphen to indicate a range. The second subaddress must be greater than the first. By default, no subaddress range is used.	
USER user-id	Specifies the user identification for use by incoming calls to the destination. Specify a string from 1 to 16 characters. The default is that no user identification is used.	

Command Parameters

ALL

Copies the call handler parameters from the permanent database into the volatile database. Used only for the SET MODULE X25-SERVER command or the SET MODULE X29-SERVER command.

COUNTER TIMER seconds

Sets a timer whose expiration causes a server module counter logging event. Specify a decimal integer up to a maximum of 65,535.

MAXIMUM CIRCUITS count

Specifies the maximum number of circuits that the module (that is, all destinations) may have. By default, the maximum is 512.

STATE module-state

Specifies the operational state of the server module. This parameter takes one of the following values:

OFF	The module is not in use. For the X.29 server module, OFF is the default state.
ON	The module is available for normal use. For the X.25 server module, ON is the default state.

SHUT	The module is to be closed down but only when all present activity has ceased. The SHUT value applies only to the SET MODULEX25-SERVER or X29-SERVER command.

Example

1. `NCP> SET MODULE X25-SERVER DESTINATION JOE SUBADDRESSES 12-24 -`
   `_ SENDING ADDRESS 987321654 PRIORITY 3 OBJECT OBJONE`

   This command creates a destination called JOE that will handle incoming X.25 calls from a DTE with an address of 987321654 and a subaddress in the range 12 to 24 specified in the DECnet object database.

2. `NCP> SET MODULE X25-SERVER DESTINATION DEFDES -`
   `_ OBJECT LAST PRIORITY 0`

   This command modifies the destination called DEFDES that will handle any incoming calls. The destination has the lowest priority and will only handle calls that fail to find another destination.

3. `NCP> SET MODULE X25-SERVER DESTINATION THRUSH SUBADDRESSES 11-20 -`
   `_ OBJECT 36 NODE THRUSH`

   This command creates a destination called THRUSH for node THRUSH that will match incoming X.25 calls with a subaddress in the range 11 to 20. This command is used if the executor node is a VMS node with VAXP.S.I. software in multihost mode installed, serving as a connector node. Object 36 is the object number for VAX P.S.I. Access on the remote node.

SET NODE ALL

SET NODE ALL — The SET NODE ALL command updates the volatile database on the executor node with all the parameters stored for a particular node in the permanent database on the executor node.

Format

SET   [node-component ALL]

Node Components

NODE
KNOWN NODES

Node Components

[NODE node-id]

Identifies the node whose parameters are to be updated.

KNOWN NODES

Indicates that all parameters for all known nodes are to be updated.

Example

1. `NCP> SET KNOWN NODES ALL`

This command loads all permanent database entries stored for all known nodes into the volatile database.

SET/DEFINE NODE

SET/DEFINE NODE — The SET NODE command creates or modifies node parameters in the volatile database on the local node. The DEFINE NODE command creates or modifies node parameters in the permanent database on the local node.

Format

SET   [node-component parameter [...]]

DEFINE   [node-component parameter [...]]

Node Components

NODE
KNOWN NODES

Node Components

NODE node-id

Identifies the node (local or remote) for which specified parameters are to be created or modified in the database.

KNOWN NODES

Indicates that the specified parameters for all known nodes are to be created or modified in the database.

Command Parameters

ACCESS option

Specifies the allowed logical link connections for the node. There are four options:

BOTH	Allows incoming and outgoing logical link connections. This is the default.
INCOMING	Allows logical link connections from the remote node.
NONE	Does not allow incoming or outgoing logical link connections to this node.
OUTGOING	Allows the local node to initiate connections to the remote node, but does not allow connections from the remote node.

If you have OPER privilege, you can override the access restriction specified in this parameter.

ADDRESS node-address

Specifies the address of the node to which you want the database entry to refer.

COUNTER TIMER seconds

Specifies a timer whose expiration causes a node counter logging event.

CPU cpu-type

Identifies the node's CPU type. There are four possibilities:

DECSYSTEM1020
PDP11
PDP8
VAX

[DIAGNOSTIC FILE filespec]

Applies to nodes on Ethernet circuits. Identifies the file to be read when the adjacent node has been downline loaded and has requested diagnostics. The *filespec* is interpreted according to the file system of the executor node.

DUMP ADDRESS number

Identifies the address in memory to begin an upline dump of the adjacent node.

DUMP COUNT number

Specifies the default number of memory units to upline dump from the adjacent node.

DUMP FILE filespec

Identifies which file to write to when the adjacent node is dumped up line. The *filespec* is interpreted according to the file system of the executor node.

HARDWARE ADDRESS E-address

Identifies the Ethernet address originally assigned to the Ethernet controller for the system on the adjacent node. Used during operations such as downline loading to communicate with the system before the system has set up its physical address.

HOST node-id

Identifies the host node. For an adjacent node, the host address is a parameter that the adjacent node receives when it is downline loaded. If no host is specified, the default is the executor node.

INBOUND node-type

Required for nodes when the VERIFICATION INBOUND parameter is specified for the circuit over which the connection is to be made. Specifies the type of the node. The *node-type* is checked by the executor node if the specified node attempts to form a dynamic connection with the executor node. If VERIFICATION INBOUND is not specified for the circuit, the INBOUND parameter for the node is ignored. There are two possible node types:

ENDNODE	Allows the remote node to be connected only if it is configured as an end node.
ROUTER	Allows the remote node to be connected whether it is configured as an end node or a router.

LOAD ASSIST AGENT filespec

Specifies the VMS image that will define the system software to be downline loaded to an adjacent node. The load assist agent can be used to tailor the system software to be loaded. The LOAD ASSIST AGENT parameter applies to downline load operations to Local Area VAXclusters only.

LOAD ASSIST PARAMETER item

Specifies a parameter to be passed to a load assist agent. The LOADASSIST PARAMETER value applies to downline load operations to Local Area VAXclusters only.

LOAD FILE filespec

Specifies a file containing the system software to be downline loaded to an adjacent node.

MANAGEMENT FILE filespec

Specifies a file containing the management information to be downline loaded to an adjacent node.

NAME node-name

Specifies the node name to be associated with the node identification. You can assign only one name to a node address or line identification.

NONPRIVILEGED item

Specifies nonprivileged inbound access control information for the node. Associate any of the following parameters with the NONPRIVILEGED parameter:

ACCOUNT account	Identifies the account for the default nonprivileged DECnet account on the designated node.
PASSWORD password	Identifies the password for the default nonprivileged DECnet account on the designated node.
USER user-id	Identifies the user name for the default nonprivileged DECnet account on the designated node.

PRIVILEGED item

Specifies privileged inbound access control information for the node. Associate any of the following parameters with the PRIVILEGED parameter:

ACCOUNT account	Identifies the account for the default privileged DECnet account on the designated node.
PASSWORD password	Identifies the password for the default privileged DECnet account on the designated node.
USER user-id	Identifies the user name for the default privileged DECnet account on the designated node.

RECEIVE PASSWORD password

Does not apply to nodes on an Ethernet circuit. Defines the password (1 to 8 characters) that is expected from the remote node during a routing initialization sequence. You use this parameter only if verification is enabled or set to INBOUND for the circuit.

[SECONDARY LOADER filespec]

Specifies a file containing the secondary boot loader to be downline loaded to an adjacent node.

[SERVICE CIRCUIT circuit-id]

Establishes the circuit to be used for downline loading. This circuit is the default value for the VIA parameter of the LOAD command. The node identification must be that of the target node.

[SERVICE DEVICE device-type]

Identifies the target node's line controller for the service line over which the operation is to take place. The possibilities are as follows:

BNA	DA	DL	DLV	DMB	DMC
DMF	DMP	DMR	DMV	DP	DQ
DTE	DU	DUP	DV	DZ	KDP
KDZ	KL	PCL	QNA	SVA	UNA

Only the synchronous port on the DMF service device may be used.

SERVICE NODE VERSION version

Specifies the DECnet–VAX software version of the node that downline loads its software to a target node. The two possibilities are PHASE III and PHASE IV. The default is PHASE IV.

SERVICE PASSWORD hex-password

Defines the password required to trigger the bootstrap mechanism on the target node. The password is a 16-digit hexadecimal number. For DDCMP circuits, the password is in the range 0 to FFFFFFFF; for Ethernet circuits, it is in the range 0 to FFFFFFFFFFFFFFFF. The default is 0.

SOFTWARE IDENTIFICATION software-id

Specifies the ID of the software to be downline loaded. You include this parameter only if the SOFTWARE TYPE parameter is specified.

SOFTWARE TYPE software-type

Identifies a particular file type to be downline loaded. There are four possibilities:

MANAGEMENT FILE
SECONDARY LOADER
SYSTEM
TERTIARY LOADER

The default is SECONDARY LOADER.

You include this parameter only if the SOFTWARE IDENTIFICATION parameter is specified.

TERTIARY LOADER filespec

Specifies a file containing a tertiary boot loader to be downline loaded to an adjacent node.

TRANSMIT PASSWORD password

Does not apply to nodes on an Ethernet circuit. Specifies a password (1 to 8 characters) sent to the remote node during a routing initialization sequence. This parameter is used only if the VERIFICATION parameter has been set to ENABLED or INBOUND for the circuit.

Description

You can use this command to create or modify parameters for a remote node. You can also use this command, specifying the local node address, to create or modify certain executor parameters.

Refer to the *VSI OpenVMS DECnet Networking Manual* for a table of the parameters used with the SET NODE command according to their functions.

Example

1. `NCP> SET NODE 5.14 NAME DENVER`

   This command sets the node name of node 5.14 to DENVER.

2. ```
   NCP> SET NODE 2.11 −
   _ NONPRIVILEGED −
   _ USER NETNONPRIV −
   _ PASSWORD NONPRIV −
   _ PRIVILEGED −
   _ USER NETPRIV −
   _ PASSWORD PRIV
   ```

   This command establishes default privileged and nonprivileged access control information for node 2.11.

3. `NCP> SET NODE 14 ADDRESS 2`

   This command associates the information for node 1.14 with a new node whose address is 1.2. The executor is assumed to be in area 1.

4. `NCP> SET NODE LARK HARDWARE ADDRESS AA−00−03−00−00−C0`

   This command associates with the node LARK the Ethernet hardware address originally assigned to the Ethernet controller at node LARK. This information in the volatile database can be used during down line loading of target node LARK over an Ethernet circuit.

5. `NCP> SET NODE 8.* ACCESS NONE`

   This command prevents incoming or outgoing logical link connections to all nodes in area 8.

SET/DEFINE NODE CIRCUIT

SET/DEFINE NODE CIRCUIT — The SET NODE command with the CIRCUIT parameter creates or modifies loop node parameters in the volatile database on the local node. The DEFINENODE command with the CIRCUIT parameter creates or modifies loop node parameters in the permanent database on the local node. The CIRCUIT parameter is the only valid parameter for loop nodes. You can assign only one loop node name to a particular circuit.

Format

SET   [node-component parameter]

DEFINE   [node-component parameter]

Node Component

NODE

Node Component

NODE node-id

Identifies the loop node name or address for which the CIRCUIT parameter is to be created or modified in the database.

Command Parameter

CIRCUIT circuit-id

Identifies which circuit to use for all traffic to the node.

Example

1. `NCP> SET NODE TESTER CIRCUIT DMP-0.4`

   This command sets the loop node name (TESTER) and identifies the circuit over which loop information is to be transmitted.

SET OBJECT ALL

SET OBJECT ALL — The SET OBJECT ALL command updates the volatile database on the local node with all the object parameters stored for a particular object in the permanent database on the local node.

Format

SET [object-component ALL]

Object Components

OBJECT
KNOWN OBJECTS

Object Components

OBJECT object-name

Identifies the object whose parameters are to be updated.

KNOWN OBJECTS

Indicates that all parameters for all known objects are to be updated.

Example

1. `NCP> SET KNOWN OBJECTS ALL`

This command loads all permanent database parameter entries stored for all known objects into the volatile database.

SET/DEFINE OBJECT

SET/DEFINE OBJECT — The SET OBJECT command creates or modifies object parameters in the volatile database on the local node. The DEFINE OBJECT command creates or modifies object parameters in the permanent database on the local node. For VAX P.S.I. operations, the SET OBJECT command specifies the parameters in the volatile database for a process that is activated by module X25-SERVER or X29-SERVER when an incoming X.25 or X.29 call arrives. For VAX P.S.I., the DEFINEOBJECT command specifies these parameters in the permanent database.

Format

SET   [object-component parameter [...]]

DEFINE   [object-component parameter [...]]

Object Components

OBJECT
KNOWN OBJECTS

Object Components

OBJECT object-name

Identifies the object for which specified parameters are to be created or modified in the database.

KNOWN OBJECTS

Indicates that the specified parameters are applicable to all known objects.

Command Parameters

ACCOUNT account

Identifies the default user's account for access control on inbound connects to the object when no access control is specified by the remote node. For VAX P.S.I., identifies the user account for use by incoming X.25 calls to the object; if not specified, no account is used.

ALIAS INCOMING option

Does not apply to VAX P.S.I.

Specifies how a particular object responds to incoming connect requests directed to the alias node address. You establish the alias node address using the SET EXECUTOR command. There are two options for ALIAS INCOMING.

DISABLED	Does not allow a specified object to receive incoming connect requests that have been directed to the alias node address. An object whose

	resources are not accessible clusterwide should have ALIAS INCOMING disabled. If an attempt is made to connect to an object that does not have ALIAS INCOMING enabled, the status message NO SUCH OBJECT is returned.
ENABLED	Allows a specified object to receive incoming connect requests that have been directed to the alias node address. An object such as PHONE, which uses a protocol that depends on multiple links, should not be enabled for ALIAS INCOMING. By default, if an alias node identifier has been specified, ALIAS INCOMING is enabled for all objects except for PHONE.

[ALIAS OUTGOING option]

Does not apply to VAX P.S.I.

Specifies whether a particular object uses the alias node identifier specified in the SET EXECUTOR command in its outgoing connect requests and other protocols. Specify either of the following two options:

DISABLED	Does not allow a specified object to use the alias node address in its outgoing connect requests.
ENABLED	Allows a specified object to use the alias node address in its outgoing connect requests. An object such as PHONE, which uses a protocol that depends on multiple links, should not have the ALIAS OUTGOING parameter enabled. By default, only the object MAIL has ALIAS OUTGOING enabled.

FILE filespec

Specifies the command file containing the command procedure used to start the indicated object. If not specified, the default is SYS$SYSTEM:object-name.COM. When you specify an object for the first time, this parameter is mandatory.

NUMBER number

Specifies the object number. Use a number up to a maximum of 255,except for those reserved. See Table 3.1 for a list of reserved object numbers. When you specify an object for the first time,this parameter is mandatory. For VAX P.S.I., specify this parameter as 0.

OUTGOING CONNECT PRIVILEGES privilege-list

Does not apply to VAX P.S.I.

Specifies those privileges that a user must have in order to make an outbound connection to the object.

PASSWORD password

Identifies the default user's password for access control on inbound connects to the object when no access control is specified by the remote node. This password must match the password established for the account. For VAX P.S.I., this parameter identifies the user's password for use by incoming X.25 or X.29 calls to the object.

PRIVILEGES privilege-list

Does not apply to VAX P.S.I.

Specifies those privileges normally required by the object. A user with those privileges may be supplied with default outbound privileged access control information when connecting to the object.

PROXY option

Assigns the proxy login access defaults to individual objects. Specify one of the following four options:

BOTH	Allow both incoming and outgoing proxy login access. This is the default option.
INCOMING	Allows proxy login to the object.
NONE	Does not allow incoming or outgoing proxy login access.
OUTGOING	Allows the object to initiate proxy login.

USER user-id

Identifies the default user's identification for access control on inbound connects to the object when no access control is specified by the remote node. For VAX P.S.I., this parameter specifies the user identification for use by incoming X.25 calls to the object. When you specify an object for the first time, this parameter is mandatory.

Description

A DECnet object is identified by object name and object type. (The type is specified in the NUMBER parameter.) For VAX P.S.I. operations, an object is identified by *object-name*.

The privilege list in the SET/DEFINE OBJECT command is used to validate the user privileges for outbound connections to that object. The access control information is used as the default access control for inbound connections.

Refer to the *VSI OpenVMS DECnet Networking Manual* for a table of object parameters and their functions. Table 3.1 lists the object type codes used with the SETOBJECT and DEFINE OBJECT commands. All values in Table 3.1 are expressed in decimal.

Table 3.1. Object Type Codes

Code	Object Type Mnemonic	Description
0	TASK	User program
1–16		Reserved for VSI use
17	FAL	File Access Listener for remote file and record access
18	HLD	Host loader for RSX–11S down line task loading requests
19	NML	Network Management Listener object
20		RSTS/E media transfer program (NETCPY)
21–22		Reserved for VSI use

Code	Object Type Mnemonic	Description
23	REMACP	Network terminal handler (host side)
24		Network terminal handler (terminal side)
25	MIRROR	Loop back mirror
26	EVL	Event receiver
27	MAIL	VMS Mail Utility
28		Reserved for VSI use
29	PHONE	VMS Phone Utility and RSX–11M/M–PLUS Phone Utility
30–41		Reserved for VSI use
42	CTERM	Network terminal handler
43–62		Reserved for VSI use
63	DTR	DECnet Test Receiver object
64–127		Reserved for VSI use
128–255		Reserved for customer use

Example

1. ```
   NCP> SET OBJECT NML NUMBER 19
   ```

   This command sets an object number, 19, for the NML object.

2. ```
   NCP> SET OBJECT NML –
   _ PRIVILEGES OPER DIAGNOSE –
   _ USER NET_NONPRIV –
   _ PASSWORD NET_NONPRIV
   ```

   This command establishes default access control information for the NML object and sets those privileges required to connect to this object.

3. ```
   NCP> SET OBJECT OBJONE FILE STARTUP.COM NUMBER 0 –
   _ USER NET PASSWORD NET
   ```

   This command creates an object called OBJONE with a command procedure called STARTUP.COM. The incoming connection uses a password and user identification of NET.

4. ```
   NCP> SET EXECUTOR ALIAS NODE 2.13...
   NCP> SET OBJECT FOX ALIAS OUTGOING ENABLED
   ```

   The SET EXECUTOR ALIAS NODE command establishes address 2.13 as the alias node identifier for the local node. The SET OBJECT command associates the object FOX with the alias node identifier for all outgoing connect requests.

5. ```
   NCP> SET OBJECT T* PROXY NONE
   ```

   This command prevents incoming or outgoing proxy login access for all objects with names beginning with *T*.

6. ```
   NCP> SET OBJECT TEST OUTGOING CONNECT PRIVILEGES READALL OPER
   ```

This command specifies that a user process must have privileges READALL and OPER in order to make an outbound connection to object TEST.

SHOW AREA

SHOW AREA — The SHOW AREA command displays area information from the volatile database available to the local node.

Format

SHOW [area-component parameter [qualifier]]

Area Components

ACTIVE AREAS
KNOWN AREAS
AREA

Area Components

ACTIVE AREAS

Indicates that information for all active areas is to be displayed.

AREA area-id

Identifies a particular area for which information is to be displayed.

KNOWN AREAS

Indicates that information for all known areas is to be displayed.

Command Parameters

CHARACTERISTICS

Indicates that static area information is to be displayed.

STATUS

Indicates that dynamic area information is to be displayed.

SUMMARY

Indicates that only the most useful area information is to be displayed. This is the default display type.

Qualifier

TO filespec

Specifies the output file. If none is specified, SYS$OUTPUT is the default.

Interpreting the Display

CIRCUIT circuit-id

This read-only parameter identifies the circuit used to get to a remote area. The *circuit-id* is an ID string.

COST cost

This read-only parameter represents the total cost over the current path to the destination area. The *cost* is a positive integer value associated with using a circuit. The Routing layer routes messages (data) along the path between two areas with the least cost.

HOPS hops

This read-only parameter represents the number of hops over to a destination area. A hop is the routing value representing the logical distance between two areas in a network.

NEXT NODE node-id

This read-only value indicates the next node on the circuit used to get to the destination area.

STATE state

This read-only value indicates the state of the area. The two possible states are REACHABLE and UNREACHABLE.

Example

```
1. NCP> SHOW KNOWN AREAS STATUS
   Known Area Volatile Status as of 30-DEC-1989 09:50:34
     Area      State      Cost    Hops    Circuit         Next node to area
      2       reachable    0       0                        2.11 (BOSTON)
      3       reachable    4       1      DMC-0             3.5 (TRNTO)
      5       reachable    7       2      DMC-0             3.5 (TRNTO)
     11       reachable    4       1      DMC-3            11.9 (DALLAS)
     44       reachable   11       1      X25-INC          44.2 (LONDON)
```

This command displays status information for all known areas in the network.

```
2. NCP> SHOW KNOWN AREA CHARACTERISTICS
   Known Area Volatile Characteristics as of 30-DEC-1989 11:16:27Area = 2
   State                   = reachable
   Next node to area       =  2.11 (BOSTON)Area = 3
   State                   = reachable
   Circuit                 = DMC-0
   Next node to area       =  3.5 (TRNTO)Area = 5
   State                   = reachable
   Circuit                 = DMC-0
   Next node to area       =  3.5 (TRNTO)Area = 11
   State                   = reachable
   Circuit                 = DMC-3
   Next node to area       =  11.9 (DALLAS)Area = 44
   State                   = reachable
   Circuit                 = X25-INC
```

```
   Next node to area        =  44.2 (LONDON)
```

This command displays characteristics for all known areas in the network.

3. ```
   NCP> SHOW AREA 11 SUMMARY
   Known Area Volatile Summary as of 30-DEC-1989 11:16:44
     Area    State         Circuit           Next node to area
     11      reachable      DMC-3             11.9 (DALLAS)
   ```

This command displays only the most useful information for area 11 in the network.

SHOW/LIST CIRCUIT

SHOW/LIST CIRCUIT — The SHOW CIRCUIT command displays circuit information from the volatile database available to the local node or DTE. The LIST CIRCUIT command displays circuit information from the permanent database available to the local node or DTE.

Format

SHOW   [circuit-component parameter [qualifier] [...]]

LIST   [circuit-component parameter [qualifier] [...]]

Circuit Components

ACTIVE CIRCUITS
KNOWN CIRCUITS
CIRCUIT

Circuit Components

ACTIVE CIRCUITS

Indicates that information for all active circuits is to be displayed.

CIRCUIT circuit-id

Identifies a particular circuit for which information is to be displayed.

KNOWN CIRCUITS

Indicates that information for all known circuits is to be displayed.

Command Parameters

CHARACTERISTICS

Indicates that static circuit information is to be displayed.

COUNTERS

Indicates that circuit error and performance statistics are to be displayed.

STATUS

Indicates that dynamic circuit information is to be displayed, including end node adjacencies and routing node adjacencies.

SUMMARY

Indicates that dynamic circuit information is to be displayed, including the routing adjacencies available to this circuit. SUMMARY is the default display type.

Qualifiers

ADJACENT NODE node-id

Indicates that the display of a list of circuits is to be restricted to those circuits leading to the specified adjacent node.

TO filespec

Specifies the output file. If none is specified, SYS$OUTPUT is the default.

Interpreting the Display

Adjacent node node-id

This read-only parameter indicates an adjacent node on the circuit. There can be many adjacent nodes on an Ethernet circuit.

Block size number

This read-only parameter is the block size in bytes for the adjacent node, as negotiated with the adjacent Routing layer during routing initialization over the circuit.

Designated router node-id

This read-only value is the Routing layer identification of the node that is to be used for routing to nonrouting nodes (end nodes) on this circuit.

Listen timer seconds

This read-only parameter determines the maximum time allowed to elapse before a message (a Routing Hello message or a user message) is received from an adjacent node on the circuit. The value can be up to a maximum of 65,535. Note that the LISTEN TIMER value is three times that of the HELLO TIMER circuit parameter.

Loopback name

This read-only parameter is the node name associated with a circuit for loopback testing. It identifies the circuit to be used for all traffic to the loop node.

Polling substate

Applies only to DDCMP CONTROL circuits. This read-only value represents the state of the tributary as determined by the polling algorithm when the multipoint polling state is AUTOMATIC.

The polling substate is displayed as a tag on the polling state (for example, AUTOMATIC-INACTIVE). Possible values of the polling substate are as follows:

Active
Inactive
Dying
Dead

Substate

This read-only value is the operational substate of the circuit. The substate is displayed as a tag on the STATE parameter (for example, ON-SYNCHRONIZING). See Table C.1 for a complete list of circuit/line substates. Possible substate values are as follows:

Synchronizing
Starting
Reflecting
Looping
Loading
Dumping
Triggering
Autoservice
Autoloading
Autodumping
Autotriggering
Failed

Example

1.
```
NCP> SHOW KNOWN CIRCUITS STATUS
Known Circuit Volatile Status as of 30-DEC-1989 15:39:04
   Circuit            State                    Loopback     Adjacent
 Block
                                               Name           Node
 Size
  DMC-0              on                                    3.5  (TRNTO)
576
  DMC-1              on        -starting
  UNA-0              on                                    2.22 (LARK)
576
                                                           2.23 (DOVE)
576
                                                           2.20 (ROBIN)
576
                                                           2.21 (THRUSH)
576
  X25-INC            on                        INC         44.2 (LONDON)
576
  X25-INC2           on        -synchronizing
  X25-RY1            off                       RY1
  X25-RY2            off                       RY2
  X25-ZK             off                       ZK
```

This command displays status information for all known circuits connected to the local node. This information includes the current state of the circuit; a loop node name (if any) associated with the

circuit; and the address, name, and block size of the adjacent node on that circuit. For Ethernet circuits, all nodes attached to the circuit are displayed.

2. ```
NCP> SHOW ACTIVE CIRCUITS CHARACTERISTICS
Active Circuit Volatile Characteristics as of 30-DEC-1989 15:39:21
Circuit = DMC-0
State   = on
Service   = enabled
Cost   = 12
Hello timer   = 15
Listen timer   = 30
Maximum buffers   = 255
Verification   = disabled
Adjacent node   = 3.5 (TRNTO)
Listen timer   = 30
Circuit = UNA-0
State   = on
Designated router   = 2.20 (ROBIN)
Cost   = 1
Maximum routers allowed = 33
Router priority   = 64
Hello timer   = 15
Verification   = disabled
Adjacent node   = 2.22 (LARK)
Listen timer   = 45
Circuit = UNA-0
Adjacent node   = 2.23 (DOVE)
Listen timer   = 45
Circuit = UNA-0
Adjacent node   = 2.20 (ROBIN)
Listen timer   = 45
Circuit = UNA-0
```

This command displays circuit characteristics for all circuits whose states are ON.

3. ```
NCP> SHOW CIRCUIT UNA-0 STATUS
Circuit Volatile Status as of 30-DEC-1989 15:45:04
   Circuit   State   Loopback   Adjacent   Block
                                  Name   Node   Size
   UNA-0   on   2.22 (LARK)   576
   UNA-0   2.23 (DOVE)   576
   UNA-0   2.20 (ROBIN)   576
   UNA-0   2.21 (THRUSH)   576
```

This command displays status information for circuit UNA–0.

4. ```
NCP> TELL DOVE SHOW CIRCUIT UNA-0 CHARACTERISTICS
Circuit Volatile Characteristics as of 30-DEC-1989 15:46:20
Circuit = UNA-0
State                   = on
Designated router       = 2.20 (ROBIN)
Cost                    = 1
Maximum routers allowed = 33
Router priority         = 64
Hello timer             = 15
Verification            = disabled
Adjacent node           = 2.20 (ROBIN)
Listen timer            = 45
```

```
Circuit = UNA-0
Adjacent node          = 2.11 (BOSTON)
Listen timer           = 45
Circuit = UNA-0
Adjacent node          = 2.22 (LARK)
Listen timer           = 45
Circuit = UNA-0
Adjacent node          = 2.21 (THRUSH)
Listen timer           = 45
```

This command displays circuit characteristics for circuit UNA–0 on node DOVE.

5. NCP> TELL MYNODE SHOW CIRCUIT TX-0-5 CHARACTERISTICS
```
Circuit Volatile Characteristics as of 30-DEC-1989 15:35:12
Circuit = TX-0-5
State                  = on
Substate               = -synchronizing
Service                = enabled
Cost                   = 10
Hello timer            = 15
Verification           = inbound
```

This command displays circuit characteristics for circuit TX-0-5 on node MYNODE.

6. NCP> SHOW KNOWN CIRCUIT ADJACENT NODE BOSTON
```
Known Circuit Volatile Summary as of 30-DEC-1989 15:50:19
   Circuit           State               Loopback      Adjacent
                                         Name          Node
   DMC-0             on                          2.11 (BOSTON)
   UNA-0             on                          2.11 (BOSTON)
```

This command displays summary information for all circuits connected to adjacent node BOSTON.

7. NCP> SHOW ACTIVE CIRCUIT COUNTERS
```
Active Circuit Counters as of 30-DEC-1989 16:10:31
Circuit = DMC-0   >65534
  Seconds since last zeroed      734
  Terminating packets received     101
  Originating packets sent         0
  Terminating congestion loss        0
  Transit packets received       0
  Transit packets sent       0
  Transit congestion loss      17
  Circuit down      0
  Initialization failure  3165010
  Bytes received  3066061
  Bytes sent     45098
  Data blocks received   43538
  Data blocks sent      105
  Data errors outbound, including:
        NAKs received, header block check error
      0  Data errors inbound
      0  Local buffer errors
    228  Remote buffer errors, including:
        NAKs received, buffer unavailable       0
  Local reply timeouts        0
  Remote reply timeouts
Circuit = UNA-0   >65534
```

```
      Seconds since last zeroed   184949
      Terminating packets received   214036
      Originating packets sent        0
      Terminating congestion loss   555679
      Transit packets received   555334
      Transit packets sent         2
      Transit congestion loss        1
      Circuit down         0
      Initialization failure  1142708
      Data blocks sent 104475601
      Bytes sent   2389413
      Data blocks received 277120757
      Bytes received         0
      Unrecognized frame destination      9
      User buffer unavailable
   Circuit = X25-INC   >65534
      Seconds since last zeroed      650
      Terminating packets received      959
      Originating packets sent        0
      Terminating congestion loss    11540
      Transit packets received    19461
      Transit packets sent      243
      Transit congestion loss       18
      Circuit down        0
      Initialization failure
```

This command displays counter information for all active circuits. Refer to Appendix C for a description of each circuit counter.

8. `NCP> SHOW CIRCUIT SVA-* SUMMARY`

This command displays dynamic circuit information for all SVA circuits, including the routing adjacencies available to those circuits.

SHOW/LIST EXECUTOR

SHOW/LIST EXECUTOR — The SHOW EXECUTOR command displays local node information from the volatile database. The LIST EXECUTOR command displays local node information from the permanent database.

Format

SHOW EXECUTOR [parameter [qualifier]]

LIST EXECUTOR [parameter [qualifier]]

Qualifier

TO

Command Parameters

CHARACTERISTICS

Indicates that static local node information is to be displayed.

COUNTERS

Indicates that local node error and performance statistics are to be displayed.

STATUS

Indicates that dynamic local node information is to be displayed.

SUMMARY

Indicates that only the most useful local node information is to be displayed. This is the default display type.

Qualifier

TO filespec

Specifies the output file. If none is specified, SYS$OUTPUT is the default.

Interpreting the Display

Active links number

This read-only parameter represents the number of active logical links from the executor to the destination node.

Delay seconds

This read-only parameter is the average round-trip delay in seconds from the executor to the destination node.

Management version n.n.n

This read-only parameter identifies the version number of the Network Management layer. The format of the number consists of the version number, the Engineering Change Order (ECO) number, and the user ECO number (for example, V3.0.0).

NSP version n.n.n

This read-only parameter identifies the version number of the End Communication layer. The format for the number is the same as for the management version number.

Physical address E-address

This read-only parameter is the Ethernet address that identifies the executor node.

Routing version n.n.n

This read-only parameter identifies the version number of the Routing layer. The format for the number is the same as for the management version number.

Example

1. ```
NCP> SHOW EXECUTOR CHARACTERISTICS
   Node Volatile Characteristics as of 30-DEC-1989 15:37:32
   Executor node   = 2.11 (BOSTON)
   Identification   = DECnet-VAX V5.0,   VMS V5.0
   ```

```
Management version      = V4.0.0
Incoming timer          = 45
Outgoing timer          = 45
Incoming Proxy          = Enabled
Outgoing Proxy          = Enabled
NSP version             = V4.0.0
Maximum links           = 128
Delay factor            = 80
Delay weight            = 5
Inactivity timer        = 60
Retransmit factor       = 10
Routing version         = V2.0.0
Type                    = routing IV
Routing timer           = 600
Broadcast routing timer = 40
Maximum address         = 1023
Maximum circuits        = 16
Maximum cost            = 1022
Maximum hops            = 15
Maximum visits          = 63
Maximum area            = 63
Max broadcast nonrouters = 64
Max broadcast routers   = 32
Maximum path splits     = 1
Area maximum cost       = 1022
Area maximum hops       = 30
Maximum buffers         = 100
Buffer size             = 576
Default access          = incoming and outgoing
Pipeline quota          = 1500
Alias incoming          = Enabled
Alias maximum links     = 32
Alias node              = 2.10 (CLUSTR)
Path split policy       = Normal
```

This command displays local node characteristics. This display shows values that you have set for the local node. In addition, it provides supplemental information about the software versions of NML, NSP, and Routing.

2. ```
   NCP> SHOW EXECUTOR STATUS
   Node Volatile Status as of 30-DEC-1989 15:37:53
   Executor node           = 2.11 (BOSTON)
   State                   = on
   Physical address        = AA-00-04-00-AB-04
   Active links            = 2
   Delay                   = 1
   ```

   This command displays status information for the local node. This format includes the operational state of the local node. The physical address is the Ethernet address of the executor node.

3. ```
   NCP> SHOW EXECUTOR SUMMARY
   Node Volatile Summary as of 30-DEC-1989 15:38:04
   Executor node           = 2.11 (BOSTON)
   State                   = on
   Identification          = VMS HOST SYSTEM
   Active links            = 4
   ```

   This command displays summary information for the executor node.

```
4.  NCP> SHOW EXECUTOR COUNTERS
    Node Counters as of 30-DEC-1989 15:38:17
    Executor node = 2.11 (BOSTON)   >65534
      Seconds since last zeroed  1073585
      Bytes received  1069826
      Bytes sent    96120
      Messages received    96120
      Messages sent     1267
      Connects received     1267
      Connects sent       11
      Response timeouts       0
      Received connect resource errors      15
      Maximum logical links active       34
      Aged packet loss       70
      Node unreachable packet loss        0
      Node out-of-range packet loss        0
      Oversized packet loss       0
      Packet format error       0
      Partial routing update loss        0
      Verification reject
```

This command displays counter information for the executor node. Refer to Appendix C for a
description of each node counter.

SHOW/LIST LINE

SHOW/LIST LINE — The SHOW LINE command displays line information from the volatile database
available to the local node. The LIST LINE command displays line information from the permanent
database available to the local node.

Format

SHOW   [line-component parameter [qualifier]]

LIST   [line-component parameter [qualifier]]

Line Components

ACTIVE LINES
LINE
KNOWN LINES

Line Components

ACTIVE LINES

Indicates that information for all active lines is to be displayed.

KNOWN LINES

Indicates that information for all known lines is to be displayed.

LINE line-id

Identifies a particular line for which information is to be displayed.

Command Parameters

CHARACTERISTICS

Indicates that static line information is to be displayed.

COUNTERS

Indicates that line error and performance statistics are to be displayed.

STATUS

Indicates that dynamic line information is to be displayed.

SUMMARY

Indicates that only the most useful line information is to be displayed. This is the default display type.

Qualifier

TO filespec

Specifies the output file. If none is specified, SYS$OUTPUT is the default.

Interpreting the Display

Hardware address E-address

This read-only parameter is the Ethernet address associated with the line device hardware.

Substate

This read-only value is the operational substate of the line. The substate is displayed as a tag on the STATE parameter (for example, ON-SYNCHRONIZING). See Appendix C for a complete list of circuit/line substates. Possible substate values are as follows:

Synchronizing
Starting
Reflecting
Looping
Loading
Dumping
Triggering
Autoservice
Autoloading
Autodumping
Autotriggering
Failed

Example

1. ```
NCP> SHOW ACTIVE LINES CHARACTERISTICS
Active Line Volatile Characteristics as of 30-DEC-1989 17:54:19
```

```
   Line = DMF-0Receive buffers        = 4
   Controller                = normal
   Duplex                    = full
   Protocol                  = DDCMP point
   Service timer             = 4000
   Retransmit timer          = 3000
   Line = QNA-0
   Receive buffers           = 6
   Controller                = normal
   Protocol                  = Ethernet
   Service timer             = 4000
   Hardware address          = AA-00-03-01-25-AB
   Device buffer size        = 1498
```

This command displays line characteristics for all active lines; that is,those lines whose state is ON. This display shows values for line parameters that you have set for individual lines.

2.
```
NCP> SHOW KNOWN LINES STATUS
Known Line Volatile Status as of  30-DEC-1989 10:21:27
   Line              State
   DMC-0               on
   DMC-1               on
   DUP-0               on
   UNA-0               on
```

This command displays status information for all known lines connected to the local node. This display shows the current state of the line.

3.
```
NCP> SHOW LINE UNA-0 SUMMARY
Line Volatile Summary as of  30-DEC-1989 10:22:11
   Line              State
   UNA-0               on
```

This command displays summary information for line UNA–0.

4.
```
NCP> TELL LARK SHOW LINE UNA-0 CHARACTERISTICS
Line Volatile Characteristics as of  30-DEC-1989 10:23:41
Line = UNA-0
Receive buffers           = 9
Controller                = normal
Protocol                  = Ethernet
Hardware address          = AA-00-03-00-00-C0
Buffer size               = 1498
```

This command displays line characteristics for line UNA–0 on node LARK.

5.
```
NCP> SHOW LINE UNA-0 COUNTERS
Line Counters as of  30-DEC-1989 10:25:43
Line = UNA-0   >65534
   Seconds since last zeroed  3098176
   Data blocks received  1928504
   Multicast blocks received        0
   Receive failure 302792074
   Bytes received 200175224
   Multicast bytes received        0
   Data overrun         0
   Local buffer errors  1652342
   Data blocks sent   452703
```

```
   Multicast blocks sent       226
   Blocks sent, multiple collision      351
   Blocks sent, single collision    12169
   Blocks sent, initially deferred 139320099
   Bytes sent 46438927
   Multicast bytes sent        0
   Send failure        0
   Collision detect check failure    17270
   Unrecognized frame destination       0
   System buffer unavailable     5401
   User buffer unavailable
```

This command shows the line counters for Ethernet circuit UNA–0. Refer to Appendix C for a complete description of each line counter.

6. `NCP> SHOW LINE DMC-* STATUS`

This command displays status information for all DMC lines connected to the local node.

SHOW LINKS

SHOW LINKS — The SHOW LINKS command displays link information from the volatile database available to the local node.

Format

SHOW   [link-component parameter [qualifier]]

Link Components

KNOWN LINKS
LINK
KNOWN LINKS WITH NODE

Link Components

KNOWN LINKS

Indicates that information for all known links is to be displayed.

KNOWN LINKS WITH NODE node-id

Indicates that information for all known links to the designated node is to be displayed.

LINK link-number

Displays information about the specific link.

Command Parameters

CHARACTERISTICS

Displays static link information.

STATUS

Displays dynamic link information.

SUMMARY

Displays only the most useful link information. This is the default display type.

Qualifier

TO filespec

 Specifies the output file. If none is specified, SYS$OUTPUT is the default.

Interpreting the Display

State

 This read-only value is the operational state of the link. See Appendix C for a complete list of link states. Possible state values are as follows:

 Closed
 CI sending
 CI ACKed
 CI receiving
 CC sending
 Run
 DI received
 DI sending

Example

1. ```
NCP> SHOW KNOWN LINKS
Known Link Volatile Summary as of 30-DEC-1989 15:53:42
   Link   Node   PID   Process   Remote link Remote
 user
   4104   11.9 (DALLAS)   20600090 GRAY   3116   MAIL
   8229   3.5 (TRNTO)   2060014C DAVIS   7217   NML
   3125   2.17 (NYC)   2060023D REMACP   5175   THOMPSON_1
```

   This command displays link information for all links known to the local node. This format is the same for all three display types. The display shows the local link number, node name and address to which the links apply, the process identification number (PID), the process associated with the link, the remote link, and the remote user's PID or process name.

2. ```
NCP> SHOW KNOWN LINKS WITH NODE DALLAS
Known Link Volatile Summary as of 30-DEC-1989 15:53:42
   Link        Node           PID       Process      Remote link  Remote
 user
   4104   11.9 (DALLAS)     20600090  GRAY               3116   MAIL
```

   This command displays link information for all active links with remote node DALLAS.

3. ```
NCP> SHOW KNOWN LINKS CHARACTERISTICS
Known Link Volatile Characteristics as of 30-DEC-1989 15:00:42
```

```
Link = 8245
State                     = run
PID                       = 264006AB
Remote node               = 2.58 (THRUSH)
Delay time                = 1
Remote link               = 9270
Remote user               = TERM
Username                  = MARTIN
Process name              = MARTIN
Link = 9270
State                     = CI received
PID                       = 26400091
Remote node               = 2.58 (THRUSH)
Delay time                = 1
Remote link               = 8245
Remote user               = BARNETT
Username                  = SYSTEM
Process name              = REMACP
```

This command displays link characteristics for all links known to the local node. Refer to Appendix C for a complete description of each link state.

SHOW/LIST LOGGING

SHOW/LIST LOGGING — The SHOW LOGGING command displays logging information from the volatile database that is available to the local node. The LIST LOGGING command displays logging information from the permanent database that is available to the local node.

Format

SHOW   [logging-component parameter [qualifier] [...]]

LIST   [logging-component parameter [qualifier] [...]]

Logging Components

ACTIVE LOGGING
LOGGING FILE
KNOWN LOGGING
LOGGING MONITOR
LOGGING CONSOLE

Logging Components

ACTIVE LOGGING

Indicates that information for all active logging is to be displayed.

KNOWN LOGGING

Indicates that information for all known logging is to be displayed.

LOGGING CONSOLE

Indicates that information for the logging console is to be displayed.

LOGGING FILE

Indicates that information for the logging file is to be displayed.

LOGGING MONITOR

Indicates that information for the logging monitor is to be displayed.

Command Parameters

EVENTS

Indicates that all static logging information is to be displayed.

STATUS

Indicates that dynamic logging information is to be displayed.

SUMMARY

Indicates that only the most useful logging information is to be displayed. This is the default display type.

Qualifiers

KNOWN SINKS

Indicates that information for all known sinks is to be displayed.

SINK NODE node-id

Identifies a particular sink node for which information is to be displayed. If you do not include this parameter, NCP displays only local node logging.

TO filespec

Specifies the output file. If none is specified, SYS$OUTPUT is the default.

Example

1. ```
NCP> SHOW LOGGING FILE EVENTS KNOWN SINKS
Logging Volatile Events as of 30-DEC-1989 13:40:54
Logging sink type = file
Sink node   = 3.5 (TRNTO)
Events   = 4.0-5
Logging sink type = file
Sink node   = 11.9 (DALLAS)
Events   = 0.0-7
Events   = 2.0-1
Events   = 5.0-4
```

   This command displays events being logged to the logging file component as specified for all known sinks. This display lists the sink node, the source for events, and the actual events that are logged to the logging file at that node.

2. ```
NCP> SHOW ACTIVE LOGGING STATUS KNOWN SINKS
Active Logging Status as of 30-DEC-1989 14:02:15
```

```
      Logging sink type = console
        Sink Node      Source        Events                    State     Name
      3.5 (TRNTO)      DMC-0         4.0-5,8-10                on
                                     5.0-4
      Logging sink type = file
      Sink Node        Source        Events                    State     Name
      3.5 (TRNTO)      DMC-0         4.0-5,8-10                on        NET.LOG
                                     5.0-411.9 (DALLAS)  DMC-3         5.0-4
      11.9 (DALLAS)                  0.0-7                               2.0-1
```

This command displays status information for all active logging at all known sinks. This display
identifies each logging component and information particular to that component. The status format
is similar to the events format, except that the name of the component and its operational state are
included in this display. Note that the executor node in this example is node TRNTO.

3.
```
   NCP> SHOW KNOWN LOGGING SUMMARY SINK NODE TRNTO
   Known Logging Volatile Summary as of 30-DEC-1989 15:30:20
   Logging sink type = console
   Sink Node        Source        Events                    State     Name
   5 (TRNTO)        DMC-0         4.0-5,8-10                on
                                  5.0-4
```

This command displays summary information for all known logging to node TRNTO. The summary
format is the same as the events format.

SHOW/LIST MODULE CONFIGURATOR

SHOW/LIST MODULE CONFIGURATOR — The SHOW MODULE CONFIGURATOR command
displays information about the Ethernet configurator module from the volatile database at the local node.
The LISTMODULE CONFIGURATOR command displays information about the configurator module
from the permanent database at the local node. Note that the word MODULE is optional.

Format

SHOW MODULE CONFIGURATOR [circuit-qualifier parameter [qualifier]]

LIST MODULE CONFIGURATOR [circuit-qualifier parameter [qualifier]]

Circuit Qualifiers

CIRCUIT
KNOWN CIRCUITS

Circuit Qualifiers

CIRCUIT circuit-id

Applies only to Ethernet circuits. Indicates that configurator information should be displayed only for
the Ethernet circuit identified by *circuit-id*.

KNOWN CIRCUITS

Applies only to Ethernet circuits. Indicates that configurator information should be displayed for all
known Ethernet circuits.

Command Parameters

CHARACTERISTICS

Indicates that dynamic information about all active systems on the specified Ethernet circuit or circuits is to be displayed. (Specifying CHARACTERISTICS or STATUS results in a similar display.)

STATUS

Indicates that dynamic information about all active systems on the specified Ethernet circuit or circuits is to be displayed. (Specifying CHARACTERISTICS or STATUS results in a similar display.)

SUMMARY

Indicates that the only information to be displayed is the list of circuits being monitored and the length of time surveillance has been maintained. This is the default display type.

Qualifier

TO filespec

Specifies the output file. If none is specified, SYS$OUTPUT is the default.

Interpreting the Display

Circuit name

This read-only parameter identifies the circuit for which surveillance is being maintained.

Device type device-type

This read-only value identifies the type of device over which the remote system is communicating on the circuit. The *device-type* value represents one of the standard line devices (for example, UNA).

Elapsed time hours:minutes:seconds

This read-only value indicates the amount of time that surveillance has been enabled on the circuit. The *hours* value is a decimal integer up to a maximum of 65,535; the *minutes* and *seconds* values are decimal integers in the range 0 to 59 (for example, 19:48:24).

Function list

This read-only parameter lists the maintenance functions that the remote system supports. The list consists of one or more of the following items:

Item	Meaning
Boot	Remote controlled boot
Counters	Data link counter can be read
Dump	Upline dump
Load	Multiblock downline load
Loop	Loopback

Item	Meaning
Primary	Primary loader

Hardware address E-address

This read-only value is the Ethernet hardware address originally assigned to the DEUNA controller hardware at the remote system.

Maintenance version n.n.n

This read-only value indicates the maintenance protocol version of the remote system, consisting of the version number, the Engineering Change order (ECO) number, and the user ECO number (for example, V3.0.0).

Physical address E-address

This read-only value represents the Ethernet physical address of a remote system on the Ethernet. If the system has not set its own physical address, the value will be the same as the Ethernet hardware address originally assigned to the system.

Surveillance flag

This read-only value indicates whether surveillance is enabled or disabled for the system indicated.

Surveillance control

This read-only value indicates whether a list of active systems is to be kept for the indicated circuit. A control value of disabled means that the list is not being kept (this is the default value). A control value of enabled means that the list is being kept.

Time of last report day-month hour:minute:second

This read-only value is the date and time the remote system last reported in on a circuit under surveillance. The *day* value is a decimal integer in the range 1 to 31, *month* is the name of the month, *hour* is a decimal integer in the range 0 to 23, and *minute* and *second* are decimal integers in the range 0 to 59 (for example, 16-JUL 14:30:03).

Example

```
1. NCP> SHOW CONFIGURATOR KNOWN CIRCUITS SUMMARY TO PRINTFILE
   Module Configurator Volatile Summary as of 30-DEC-1989 08:55:21
      Circuit            Surveillance      Elapsed Time
      UNA-0              enabled           00:12:31
```

This command causes summary information about surveillance on all known Ethernet circuits to be directed to the file PRINTFILE.

```
2. NCP> SHOW MODULE CONFIGURATOR KNOWN CIRCUITS STATUS
   Module Configurator Volatile Status as of 30-DEC-1989 09:15:25
   Circuit name            = UNA-0
   Surveillance flag       = enabled
   Elapsed time            = 00:32:43
   Physical address        = AA-00-04-00-A3-4
   Time of last report     = 29-Dec 09:14:08
   Maintenance version     = V3.0.0
```

```
Function list           = Loop, Primary loader
Hardware address        = AA-00-03-00-00-07
Device type             = UNA
Circuit name            = UNA-0
Surveillance flag       = enabled
Elapsed time            = 00:32:43
Physical address        = AA-00-03-00-0A-04
Time of last report     = 29-Dec 09:11:29
Maintenance version     = V3.0.0
Function list           = Loop, Primary loader
Hardware address        = AA-00-03-00-00-0A
Device type             = UNA
Circuit name            = UNA-0
Surveillance flag       = enabled
Elapsed time            = 00:32:43
Physical address        = AA-00-04-00-D8-04
Time of last report     = 29-Dec 09:11:44
Maintenance version     = V3.0.0
Function list           = Loop, Primary loader
Hardware address        = AA-00-03-00-00-1C
Device type             = UNA
```

This command displays information about active systems on all Ethernet circuits known to the executor node.

SHOW/LIST MODULE X25-ACCESS

SHOW/LIST MODULE X25-ACCESS — The SHOW MODULE X25-ACCESS command displays network names and parameters of the X.25 Access module from the volatile database. The LIST MODULEX25-ACCESS command displays networks and parameters of the X.25 Access module from the permanent database. Note that the word MODULE is optional.

Format

SHOW MODULE X25-ACCESS   [network-qualifier parameter [qualifier]]

LIST MODULE X25-ACCESS [network-qualifier parameter [qualifier]]

Network Qualifiers

NETWORK
KNOWN NETWORKS

Network Qualifiers

KNOWN NETWORKS
NETWORK network-name

Displays information about a specific X.25 network or about all of the X.25 networks accessed through any connector nodes.

Command Parameters

CHARACTERISTICS

Indicates that static X25-ACCESS information is to be displayed.

STATUS

Indicates that dynamic X25-ACCESS information is to be displayed.

SUMMARY

Indicates that only the most useful X25-ACCESS information is to be displayed. This is the default display type.

Qualifier

TO filespec

Specifies the output file. If none is specified, SYS$OUTPUT is the default.

Example

1. `NCP> SHOW MODULE X25-ACCESS KNOWN NETWORKS STATUS TO MYFILE.NET`

   This command writes status information for all known X.25 networks to the file named MYFILE.NET.

2. ```
   NCP> SHOW MODULE X25-ACCESS NETWORK PSS1 SUMMARY
   Module X25-Access Volatile Summary as of  30-DEC-1989 12:00:56
   Network              = PSS1
   Node                 = 13.4 (EGRET)
   ```

   This command displays static network information about the X.25 network named PSS1.

SHOW/LIST MODULE X25-PROTOCOL

SHOW/LIST MODULE X25-PROTOCOL — The SHOW MODULE X25-PROTOCOL command displays the parameters of the protocol module from the volatile database. The LIST MODULE X25-PROTOCOL command displays the protocol module parameters stored in the permanent database. Note that the word MODULE and the component qualifier are optional.

Format

SHOW MODULE X25-PROTOCOL [[component-qualifier]parameter [qualifier]]

LIST MODULE X25-PROTOCOL [[component-qualifier]parameter [qualifier]]

Component Qualifiers

DTE
KNOWN DTES
GROUP
KNOWN GROUPS
NETWORK
KNOWN NETWORKS

Component Qualifiers

DTE dte-address
KNOWN DTES

> Displays information about the specified DTE or about all known DTEs.

> If you specify the DTE qualifier, you must associate with it either of the following parameters:

| NETWORK network-name | Selects the DTE(s) specified on a particular network. |
| KNOWN NETWORKS | Selects the DTE(s) on all known networks. |

GROUP group-name
KNOWN GROUPS

> Displays information about the specified group or about all known groups.

NETWORK network-name
KNOWN NETWORKS

> Displays information about the specified network or about all known networks.

Command Parameters

CHARACTERISTICS

Displays static module information.

COUNTERS

Displays DTE error and performance statistics. Does not apply to the LIST MODULE X25-PROTOCOL command.

STATUS

Displays dynamic module information. Does not apply to the LISTMODULE X25-PROTOCOL command.

SUMMARY

Displays only the most useful module information. This is the default display type.

Qualifier

TO filespec

> Specifies the output file. If you omit this parameter, the default file is SYS$OUTPUT.

Interpreting the Display

Active channels count

> This read-only parameter is the count of known switched virtual circuit (SVC) logical channel numbers currently in use. This counter applies only to those channels defined by the CHANNELS

parameter of the SET MODULE X25-PROTOCOL command. The channels can be in use for either outgoing or incoming SVCs.

Active switched count

This read-only parameter is the count of SVCs currently in use.

Maximum channels number

This read-only parameter indicates the number of logical channels defined.

Examples

1.
```
NCP> SHOW MODULE X25-PROTOCOL NETWORK FRED CHARACTERISTICS
Module X25-Protocol Volatile Characteristics as of 30-DEC-1989 17:55:24
Network                 = FRED
Profile                 = PSS
```

This command displays module characteristics for the X25-PROTOCOL module.

2.
```
NCP> SHOW MODULE X25-PROTOCOL KNOWN DTES CHARACTERISTICS
Known Module X25-Protocol Volatile Characteristics as of 30-DEC-1989
 18:07:49
DTE                     = 12345
Network                 = TALOS
Line                    = KMV-0-0
Channels                = 31-1
Maximum channels        = 31
Maximum circuits        = 255
Default data            = 128
Default window          = 2
Maximum data            = 1024
Maximum window          = 7
Maximum clears          = 6
Maximum resets          = 6
Maximum restarts        = 6
Call timer              = 200
Clear timer             = 180
Reset timer             = 180
Restart timer           = 180
Interface mode          = DTE
DTE                     = 54321
Network                 = FRED
Line                    = DPV-0-0
Channels                = 64-1
Maximum channels        = 64
Maximum circuits        = 255
Default data            = 128
Default window          = 2
Maximum data            = 1024
Maximum window          = 7
Maximum clears          = 6
Maximum resets          = 6
Maximum restarts        = 6
Call timer              = 200
Clear timer             = 180
Reset timer             = 180
```

```
Restart timer          = 180
Interface mode         = DTE
```

This command displays characteristics for all DTEs known to the X25-PROTOCOL module.

3. NCP> SHOW MODULE X25-PROTOCOL DTE 123456789 NETWORK –
   _ PSS-COMBINATION STATUS
   Module X25-Protocol Volatile Status as of 30-DEC-1989 11:50:12
```
    DTE              Network         State              Active
   Active

                                                       Channels
    Switched
    123456789      PSS_COMBINATION  on   -running        0           0
```

This command displays the status of DTE 123456789 in the network PSS-COMBINATION.

4. NCP> SHOW MODULE X25-PROTOCOL DTE 123456789 NETWORK FRED COUNTERS
   Module X25-Protocol Counters as of 30-DEC-1989 18:08:26
```
   DTE                   = 54321
   Network               = FRED
         7144  Seconds since last zeroed
        11705  Bytes received
        11705  Bytes sent
          501  Data blocks received
          501  Data blocks sent
            2  Calls received
            2  Calls sent
            0  Fast selects received
            0  Fast selects sent
            2  Maximum switched circuits active
            1  Maximum channels active
            0  Received call resource errors
            0  Locally initiated resets
            0  Remotely initiated resets
            0  Network initiated resets
            0  Restarts
```

This command displays information about each DTE counter. Refer to Appendix C for a description of each DTE counter.

5. NCP> SHOW MODULE X25-PROTOCOL KNOWN GROUPS CHARACTERISTICS
   Known Module X25-Protocol Volatile Characteristics as of 30-DEC-1989
    18:13:50
```
   Group           DTE             Network         Number    Type
    SEALS          54321           FRED                54    bilateral
    WALES          54321           FRED                 6
```

This command displays information about each closed user group.

SHOW/LIST MODULE X25-SERVER/X29-SERVER

SHOW/LIST MODULE X25-SERVER/X29-SERVER — The SHOW MODULE X25-SERVER and SHOW MODULE X29-SERVER commands display the parameters of the X.25 and X.29 call handlers stored in the volatile database. The LIST MODULE X25-SERVER and LIST MODULE X29-SERVER

commands display the parameters of the X.25 and X.29 call handlers stored in the permanent database. Note that the word MODULE is optional.

Format

SHOW MODULE X25-SERVER [[component-qualifier] parameter [qualifier]]

SHOW MODULE X29-SERVER [[component-qualifier] parameter [qualifier]]

LIST MODULE X25-SERVER [[component-qualifier] parameter [qualifier]]

LIST MODULE X29-SERVER [[component-qualifier] parameter [qualifier]]

Component Qualifiers

DESTINATION
KNOWN DESTINATIONS

Component Qualifiers

DESTINATION name
KNOWN DESTINATIONS

Displays information about the specified destination or about all known destinations.

Command Parameters

CHARACTERISTICS

Displays static module information.

COUNTERS

Displays module error and performance statistics. Does not apply to the LIST MODULE X25-SERVER or LIST MODULE X29-SERVER command.

STATUS

Displays dynamic module information. Does not apply to the LISTMODULE X25-SERVER or LIST MODULE X29-SERVER command.

SUMMARY

Displays only the most useful module information. This is the default display type.

Qualifier

TO filespec

Specifies the output file. If you omit this parameter, the default file is SYS$OUTPUT.

Interpreting the Display

Active circuits count

This read-only parameter indicates the number of circuits the module currently has open.

Example

1. ```
   NCP> SHOW MODULE X25-SERVER CHARACTERISTICS
   Module X25-Server Volatile Characteristics as of  30-DEC-1989 12:04:58
   Maximum circuits        = 255
   ```

   This command displays characteristics for the X25-SERVER module.

2. ```
   NCP> LIST MODULE X25-SERVER DESTINATION XTR CHARACTERISTICS
   Module X25-Server Permanent Characteristics as of  30-DEC-1989 12:04:27
   Destination             = XTR
   Object                  = PSIXTR
   Priority                = 30
   Sending DTE             = 234273412345
   Subaddresses            = 33-35
   ```

   This command displays server module characteristics from the permanent database for destination XTR.

3. ```
   NCP> SHOW MODULE X25-SERVER STATUS
   Module X25-Server Volatile Status as of  30-DEC-1989 12:05:13
   Active circuits         = 1
   State                   = on
   ```

   This command displays status information for the X25-SERVER module.

4. ```
   NCP> SHOW MODULE X25-SERVER COUNTERS
   Module X25-Server Counters as of  30-DEC-1989 12:05:21
           4704  Seconds since last zeroed
              7  Maximum circuits active
              2  Incoming calls rejected, no resources
   ```

   This command displays counter information for each server module. See Appendix C for a description of each server module counter.

SHOW/LIST NODE

SHOW/LIST NODE — The SHOW NODE command displays node information from the volatile database available to the local node. The LIST NODE command displays node information from the permanent database available to the local node.

Format

SHOW   [node-component parameter [qualifier]]

LIST   [node-component parameter [qualifier]]

Node Components

ACTIVE NODES
LOOP NODES
ADJACENT NODES
NODE
KNOWN NODES

Node Components

ACTIVE NODES

For a routing node, indicates that information about all reachable nodes is to be displayed. For a nonrouting node (end node), indicates that information about the executor is to be displayed. Optionally, you can associate the following CIRCUIT parameter with this parameter:

CIRCUIT circuit-id	Specifies that the display of a list of nodes is to be restricted to those nodes adjacent to the specified circuit.

ADJACENT NODES

Indicates that information about all adjacent nodes is to be displayed. Adjacent nodes are those the executor perceives Routing can reach that are separated from the executor by a single circuit. Each occurrence of anode on a different circuit appears as a separate adjacent node. Optionally, you can associate the following CIRCUIT parameter with this parameter:

CIRCUIT circuit-id	Specifies that the display of a list of nodes is to be restricted to those nodes adjacent to the specified circuit.

KNOWN NODES

Indicates that information about all known nodes is to be displayed. Optionally, you can associate the following CIRCUIT parameter with this parameter:

CIRCUIT circuit-id	Specifies that the display of a list of nodes is to be restricted to those nodes adjacent to the specified circuit.

LOOP NODES

Indicates that information about all loop nodes is to be displayed.

NODE node-id

Identifies a particular node about which information is to be displayed.

Command Parameters

CHARACTERISTICS

Indicates that static node information is to be displayed.

COUNTERS

Indicates that node error and performance statistics are to be displayed.

STATUS

Indicates that dynamic node information is to be displayed.

SUMMARY

Indicates that only the most useful node information is to be displayed. This is the default display type.

Qualifier

TO filespec

Specifies the output file. If none is specified, SYS$OUTPUT is the default.

Interpreting the Display

Active links number

This read-only parameter represents the number of active logical links from the executor to the destination node.

Circuit circuit-id

This read-only parameter identifies the circuit used to get to a remote node.

Cost number

This read-only parameter represents the total cost over the current path to the destination node. The DECnet Routing layer routes messages (data) along the path between two nodes with the smallest cost. Cost is a positive integer value.

Delay seconds

This read-only parameter is the average round-trip delay in seconds from the executor to the destination node.

Hops number

This read-only parameter indicates the number of hops from the executor node to a destination node. A hop is a value assigned by the Routing layer that represents the logical distance between two nodes on a network.

Management version n.n.n

This read-only parameter identifies the version number of the Network Management layer. The format of the number consists of the version number, the Engineering Change Order (ECO) number, and the user ECO number (for example, V3.0.0).

Next node node-id

This read-only parameter indicates the address and name of the next node on the circuit used to get to the node whose status is being displayed. Knowing which node is the partner on the next hop of the path to the destination node aids in tracing the path to that destination over a large number of hops.

NSP version n.n.n

This read-only parameter identifies the version number of the End Communication layer. The format for the number is the same as for the management version number.

Physical address E-address

This read-only parameter is the Ethernet address that identifies the executor node.

Routing version n.n.n

This read-only parameter identifies the version number of the Routing layer. The format for the number is the same as for the Management version number.

Type node-type

This read-only parameter indicates the type of the specified node. The values of *node-type* are as follows:

Phase II
Routing III
Nonrouting III
Routing IV
Nonrouting IV
Area

If the specified node is not adjacent to the local node, the *node-type* will be blank.

Example

1.  ```
    NCP> SHOW ACTIVE NODES CHARACTERISTICS
    Active Node Volatile Characteristics as of 30-DEC-1989  13:38:34
    Executor node = 2.11 (BOSTON)
    Identification         = DECnet-VAX V5.0,  VMS V5.0
    Management version     = V4.0.0
    Incoming timer         = 45
    Outgoing timer         = 45
    Incoming Proxy         = Enabled
    Outgoing Proxy         = Enabled
    NSP version            = V3.2.0
    Maximum links          = 128
    Delay factor           = 80
    Delay weight           = 5
    Inactivity timer       = 60
    Retransmit factor      = 10
    Routing version        = V2.0.0
    Type                   = routing IV
    Routing timer          = 600
    Maximum address        = 1023
    Maximum circuits       = 16
    Maximum cost           = 1022
    Maximum hops           = 15
    Maximum visits         = 63
    Maximum area           = 63
    Max broadcast nonrouters = 64
    Max broadcast routers  = 32
    Maximum path splits    = 1
    Area maximum cost      = 1022
    Area maximum hops      = 30
    Maximum buffers        = 100
    Buffer size            = 576
    Default access         = incoming and outgoing
    Pipeline quota         = 1500
    Alias incoming         = Enabled
    Alias maximum links    = 32
    Alias node             = 2.10 (CLUSTR)
    ```

```
Path split policy        = Normal
Remote node =    3.5 (TRNTO)
Nonprivileged user id    = NETNONPRIV
Remote node =   11.9 (DALLAS)
Nonprivileged user id    = NETNONPRIV
Remote node =   12.34 (MYNODE)
Inbound                  = router
Remote node =    2.13 (KANSAS)
Nonprivileged user id    = NETNONPRIV
Remote node =    2.17 (NYC)
Nonprivileged user id    = NETNONPRIV
Loop node =     0 (TESTER)
```

This command displays characteristics for all active nodes. This display shows values that you have set for both the local node and remote nodes.

2. NCP> SHOW NODE TRNTO STATUS

```
Node Volatile Status as of  30-DEC-1989 13:39:05
  Node        State      Active  Delay    Type          Cost  Hops   Line
                         Links
3.5 (TRNTO) reachable     1        2      routing IV     1     1   DMC-0
```

This command displays status information about remote node TRNTO. This format includes the following information:

- Node name and address.

- Routing state (reachable or unreachable).

- Active links associated with the node.

- Delay time required to set the retransmission (estimated round trip delay).

- Node type (phase II, routing III, routing IV, nonrouting III, nonrouting IV, or area).

- Minimum total path cost to the node. This value represents the cost of the actual path a packet will use to get to this node.

- Minimum total hops to the node. Note that the minimum cost path may be different from the minimum hops path. This value does not necessarily prevent the actual path from being used.

- Line associated with the node, that is, the line over which traffic to the node is currently being routed.

Full status information is displayed only for reachable nodes.

1. NCP> SHOW KNOWN NODES SUMMARY

```
Known Node Volatile Summary as of  30-DEC-1989 13:39:47
Executor node = 2.11 (BOSTON)
State                  = on
Identification         = HOST SYSTEM
Active links           = 2
Node              State      Active  Delay      Line
                             Links
3.5  (TRNTO)      reachable                     DMC-0
11.9 (DALLAS)     reachable    1       2        DMC-3
2.13 (KANSAS)     reachable    1       4        DMC-4
5.14 (DENVER)     unreachable
```

```
   2.15 (BANGOR)     unreachable
   2.17 (NYC)        reachable                    DMC-2
    0   (LOCAL)                                   DMC-0
```

This command displays summary information for all known nodes. The summary format contains a subset of the information displayed in the status format. Note that the display also shows loop nodes and the line associated with the loop node name.

2.  ```
    NCP> SHOW NODE TRNTO COUNTERS
    Node Counters as of  30-DEC-1989 13:40:08
    Remote node =    3.5 (TRNTO)
          18748  Seconds since last zeroed
         557304  Bytes received
         231849  Bytes sent
          27703  Messages received
          27556  Messages sent
             56  Connects received
              5  Connects sent
             12  Response timeouts
              0  Received connect resource errors
    ```

This command displays counter information about remote node TRNTO. Note that remote node counters are a subset of those maintained for the local node. Refer to Appendix C for a description of each node counter.

3.  ```
    NCP> SET NODE LOCAL LINE DMC-0
    NCP> SHOW LOOP NODES STATUS
    Loop Node Volatile Status as of  30-DEC-1989 13:43:19
      Node    State       Active Delay    Type   Cost  Hops  Line
                          Links
    0 (LOCAL)                                                 DMC-0
    ```

This command displays status information about all loop nodes.

4.  ```
    NCP> SHOW NODE M* STATUS
    ```

This command displays status information about all nodes whose names begin with *M*.

5.  ```
    NCP> SHOW NODE %SU STATUS
    ```

This command requests a display of status information of all nodes that have three-letter node names and *SU* as the second and third characters of their names.

SHOW/LIST OBJECT

SHOW/LIST OBJECT — The SHOW OBJECT command displays object information from the volatile database available to the local node. The LIST OBJECT command displays object information from the permanent database available to the local node. For VAX P.S.I., the SHOW OBJECT or LIST OBJECT command displays the parameters of a process that is activated by module X25-SERVER or X29-SERVER when an incoming X.25 call arrives.

Format

SHOW [object-component parameter [qualifier]]

LIST [object-component parameter [qualifier]]

Object Components

OBJECT
KNOWN OBJECTS

Object Components

KNOWN OBJECTS

Indicates that information about all known objects is to be displayed.

OBJECT object-name

Identifies a particular object about which information is to be displayed.

Command Parameters

CHARACTERISTICS

Indicates that static object information is to be displayed. The SHOW OBJECT CHARACTERISTICS command displays only those parameters that you have defined.

STATUS

Indicates that dynamic object information is to be displayed.

SUMMARY

Indicates that only the most useful object information is to be displayed. This is the default display type.

Qualifier

TO filespec

Specifies the output file. If none is specified, SYS$OUTPUT is the default.

Description

This command is an implementation-specific network management command because objects can have different characteristics on different DECnet implementations. An error occurs if you execute this command at a node other than a DECnet–VAX node.

Example

```
1.  NCP> SHOW OBJECT MAIL CHARACTERISTICS
    Object Volatile Characteristics as of  30-DEC-1989 13:46:22
    Object = MAIL
    Number                 = 27
    File id                = MAIL.EXE
    User id                = NETNONPRIV
    Proxy access           = outgoing
    Alias outgoing         = Enabled
    Alias incoming         = Enabled
```

This command displays object characteristics for the MAIL object. This display shows values that you have set for the object.

2. `NCP> SHOW OBJECT NML CHARACTERISTICS TO NMLCHARS.NET`

This command copies NML object characteristics from the volatile database to a file named NMLCHARS.NET.

3. ```
NCP> SHOW OBJECT PACKETSINK SUMMARY
Object Volatile Summary as of 30-DEC-1989 12:07:44
   Object   Number File/PID   User Id   Password
  PACKETSINK   0   DISK$USER01:[X25TEST]PR   X25TEST   TESTER
```

This command displays object characteristics for the PACKETSINK object.

4. `NCP> SHOW OBJECT I* SUMMARY`

This command displays object characteristics for all objects whose names begin with *I*.

TELL

TELL — The TELL command prefix specifies the executor node for a single NCP command.

Format

TELL   [node-component parameter [...] command]

Node Component

node-spec

Node Component

node-spec

Specifies a node name or address optionally followed by access control information as specified for VMS operating systems. Use one of the following formats:

```
node-idnode-id"user password account"
```

The *node-id* can be a logical name. You can override access control in a logical name by specifying explicit access control information in the command.

Command Parameters

ACCOUNT account

Identifies the user's account for access control verification at the designated executor node.

PASSWORD password

Identifies the user's password for access control verification at the designated executor node.

USER user-id

Identifies the user's ID for access control verification at the designated executor node.

Description

The TELL command sets the executor for only one command and must prefix the command for which it is intended. You cannot use the TELL prefix with the CLEAR EXECUTOR NODE, COPY KNOWN NODES, SET EXECUTOR NODE, or TELL commands.

This command allows you optionally to specify access control information in one of two ways, either as part of the *node-spec* or as distinct parameters.

Example

1. `NCP> TELL 3.5 SET KNOWN LINES ALL`

   This command sets the executor to node 3.5 where the SET KNOWN LINES ALL command executes.

2. `NCP> TELL TRNTO"GRAY MARY" SHOW KNOWN LINES...`

   This command uses access control information to set the executor to node TRNTO where the SHOW KNOWN LINES command executes. The information is displayed locally.

3. `NCP> TELL TRNTO USER SYSTEM PASSWORD MANAGER ZERO EXECUTOR`

   This command uses an alternate access control format to set the executor to node TRNTO where the ZERO EXECUTOR command executes.

TRIGGER NODE

TRIGGER NODE — The TRIGGER NODE command starts the bootstrap mechanism of a target node to allow the node to load itself. Use this command to initiate the loading sequence for an unattended system.

Format

TRIGGER   [node-component parameter [...]]

Node Component

NODE

Node Component

NODE node-id

Identifies the node whose bootstrap is to be triggered.

Command Parameters

PHYSICAL ADDRESS E-address

Applies only to nodes on Ethernet circuits. Specifies the Ethernet address that the target node currently uses to identify itself. The value is the Ethernet physical address the target node has set for itself, or, if the target node has not set an Ethernet address, the HARDWARE ADDRESS parameter if associated with the target node in the executor node's volatile database.

SERVICE PASSWORD password

Identifies the password required to trigger the bootstrap mechanism on the target node. The password is a 16-digit hexadecimal number. For DDCMP circuits, it is in the range 0 to FFFFFFFF; for Ethernet circuits, it is in the range 0 to FFFFFFFFFFFFFFFF.

VIA circuit-id

Identifies the circuit over which the operation is to take place.

Example

1. `NCP> TRIGGER NODE NYC`

   This command triggers the bootstrap mechanism on node NYC to initiate a downline load operation.

2. `NCP> TRIGGER NODE NYC SERVICE PASSWORD FEFEFEFEFEFEFEFE VIA DMC-2`

   This command provides a service password in order to trigger the bootstrap mechanism on node NYC and to initiate a downline load operation over circuit DMC–2.

3. `NCP> TRIGGER NODE SWIFT PHYSICAL ADDRESS AA-00-04-00-07-04 -`
   `_ VIA UNA-0`

   This command triggers the bootstrap mechanism on Ethernet node SWIFT to initiate a downline load operation. The executor node uses the Ethernet physical address specified in the command to address the node SWIFT on Ethernet circuit UNA–0.

4. `NCP> TRIGGER NODE SWIFT VIA UNA-0`

   This command triggers the bootstrap mechanism on node SWIFT on Ethernet circuit UNA–0 to initiate a downline load operation. DECnet–VAX at the executor node attempts to address node SWIFT using an Ethernet physical address derived from the node number. If node SWIFT does not recognize this address, the executor node uses the Ethernet hardware address for node SWIFT if specified in the volatile database at the executor node.

TRIGGER VIA

TRIGGER VIA — The TRIGGER VIA command starts the bootstrap mechanism of a target node using the specified circuit to allow the node to load itself. Use this command to initiate the loading sequence for an unattended system.

Format

`TRIGGER VIA` [circuit-id parameter [...]]

Command Parameters

PHYSICAL ADDRESS E-address

Applies only to target nodes on Ethernet circuits. Specifies the Ethernet address that the target node currently uses to identify itself. The value is the Ethernet physical address the target node has set for itself, or,if the target node has not set an Ethernet address, the HARDWARE ADDRESS parameter if it is associated with the target node in the executor node's volatile database. This parameter must be included if an Ethernet circuit is specified in VIA *circuit-id*.

SERVICE PASSWORD password

Identifies the password required to trigger the bootstrap mechanism on the target node. The password is a hexadecimal number. For DDCMP circuits, it is in the range 0 to FFFFFFFF; for Ethernet circuits, it is in the n range 0 to FFFFFFFFFFFFFFFF. The word SERVICE is optional.

Description

This command triggers the target node through the specified circuit. The target node identification is obtained from the volatile database on the executor node.

Example

1. `NCP> TRIGGER VIA DMC-0`

   This command triggers the bootstrap mechanism on the node connected to circuit DMC–0.

2. `NCP> TRIGGER VIA UNA-0 PHYSICAL ADDRESS AA-00-04-00-07-04`

   This command triggers the bootstrap mechanism at the target node whose Ethernet physical address on circuit UNA–0 is specified in the command.

ZERO CIRCUITS

ZERO CIRCUITS — The ZERO CIRCUITS command resets circuit counters on the local node. Note that the word COUNTERS is optional.

Format

`ZERO` [circuit-component [COUNTERS]]

Circuit Components

CIRCUIT circuit-id

Identifies a particular circuit for which counters are to be reset. For VAX P.S.I., specify the X.25 circuit name using an ID string.

KNOWN CIRCUITS

Indicates that counters for all known circuits are to be reset.

Example

1. `NCP> ZERO KNOWN CIRCUITS COUNTERS`

This command resets all circuit counters for all known circuits.

ZERO EXECUTOR

ZERO EXECUTOR — The ZERO EXECUTOR command resets all counters for the local node. Note that the word COUNTERS is optional.

Format

`ZERO EXECUTOR` [COUNTERS]

Example

1. `NCP> ZERO EXECUTOR COUNTERS`

   This command resets all counters for the local node.

ZERO LINE

ZERO LINE — The ZERO LINE command resets line counters on the local node. Note that the word COUNTERS is optional.

Format

`ZERO` [line-component [COUNTERS]]

Line Components

LINE line-id

Identifies a particular line for which counters are to be reset.

KNOWN LINES

Indicates that counters for all known lines are to be reset.

Examples

1. `NCP> ZERO KNOWN LINES COUNTERS`

   This command resets all line counters for all known lines.

2. `NCP> ZERO LINE DUP–0`

   This command resets line counters for the line DUP–0.

ZERO MODULE X25-PROTOCOL

ZERO MODULE X25-PROTOCOL — The ZERO MODULE X25-PROTOCOL command zeroes the counters for a particular DTE or for all known DTEs. Note that the words MODULE and COUNTERS are optional.

Format

ZERO MODULE X25-PROTOCOL [parameter [qualifier] [COUNTERS]]

Qualifiers

NETWORK
KNOWN NETWORKS

Command Parameters

[DTE dte-address]

Zeroes the counters for the specified DTE only.

[KNOWN DTES]

Zeroes the counters for all known DTEs.

You must associate the DTE parameter with either of the following qualifiers:

NETWORK network-name	Selects the DTE(s) specified on a particular network.
KNOWN NETWORKS	Selects the DTE(s) on all known networks.

Examples

1. NCP> ZERO MODULE X25-PROTOCOL KNOWN DTES COUNTERS

   This command zeroes all DTE counters for all known DTEs on the default network.

2. NCP> ZERO MODULE X25-PROTOCOL KNOWN DTES NETWORK TELENET COUNTERS

   This command zeroes counters for all DTEs on the network Telenet.

ZERO MODULE X25-SERVER/X29-SERVER

ZERO MODULE X25-SERVER/X29-SERVER — The ZERO MODULE X25-SERVER and ZERO MODULE X29-SERVER commands zero the counters for the X.25 and X.29 call handlers. Note that the words MODULE and COUNTERS are optional.

Format

ZERO MODULE X25-SERVER [COUNTERS]

ZERO MODULE X29-SERVER [COUNTERS]

Example

1. NCP> ZERO MODULE X25-SERVER COUNTERS

   This command zeroes all module counters associated with the X.25 call handler.

2. `NCP> ZERO NODE TRNTO COUNTERS`

   This command resets all node counters maintained on the local node for remote node TRNTO.

ZERO NODE

ZERO NODE — The ZERO NODE command resets node counters on the local node. Note that the word COUNTERS is optional.

Format

ZERO   [node-component [COUNTERS]]

Node Components

NODE
KNOWN NODES

Node Components

NODE node-id

Identifies a particular node for which counters are to be reset.

KNOWN NODES

Indicates that counters for all known nodes are to be reset.

Example

1. `NCP>ZERO NODE TRNTO COUNTERS`

   This command resets all node counters maintained on the local node for remote node TRNTO.

Chapter 4. DECnet Test Sender/ DECnet Test Receiver (DTS/DTR) Commands

DTS and DTR are cooperating tasks that perform various functions to exercise the DECnet software and test communication lines. DTS initiates each test by issuing a connection request to DTR. DTS passes parameters pertinent to the type of test request to DTR. The user interface to DTS consists of a group of commands. These commands enable you to select the type of test and subtest and to specify options relative to the operation of the test. These options include parameters to regulate the duration of the test, the type of information to be used for the test, and buffer-level control.

Note

The OpenVMS DECnet Test Receiver image (DTRECV) cannot perform multiple buffering. In addition, it does not support any negative acknowledgment (NAK) or BACK options nor flow control other than segment. The OpenVMS DECnet Test Sender image (DTSEND) does not support multiple buffering either.

4.1. DTS Command Syntax

This section describes the DTS command syntax for all tests. Each command consists of a command keyword, optional general command qualifiers, each preceded by a slash (/) character, and optional command-specific qualifiers, each preceded by a slash character. You may abbreviate DTS commands and qualifiers, as long as the command is unambiguous.

Note that, when using DTS commands in a batch job, you must use uppercase characters in the command.

The general DTS command syntax and the definition of each command component is as follows.

```
command-keyword/general-qualifier,.../command-specific-qualifier,...
```

command-keyword	Specifies the general test to be performed.
CONNECT	Indicates a connect test
DISCONNECT	Indicates a disconnect test
DATA	Indicates a data test
INTERRUPT	Indicates an interrupt test
general-qualifier	Specifies one or more general test characteristics regardless of the test to be performed. These qualifiers and their default values are described in each command (CONNECT, DATA, DISCONNECT, and INTERRUPT) where they apply.
command-specific-qualifier	Specifies one or more test-specific qualifiers that regulate the operation of the test.

The general command syntax adheres to the DCL rules for comments and continuation lines, as shown in the following example:

```
$ RUN SYS$SYSTEM:DTSEND
_TEST: DATA/PRINT/TYPE=SEQ !Perform Data Test
.
.
.
_TEST: INTERRUPT/NODENAME=DALLAS -
_/PRINT/TYPE=PAT !Perform Interrupt Test
.
.
.
_TEST: EXIT
$
```

4.2. DTS/DTR Usage Summary

DECnet Test Sender (DTS) and DECnet Test Receiver (DTR) are the DECnet transmitter and receiver test programs that exercise network task-to-task capabilities. DTSEND and DTRECV are the DECnet for OpenVMS implementations of these programs. The images and command files to use DTSEND and DTRECV are included with the DECnet for OpenVMS software.

There are four basic network tests provided by DTS and DTR:

- Connection tests

- Data tests

- Disconnection tests

- Interrupt tests

Each test is composed of a set of subtests. See the following sections for a description of each test and its subtests.

Format

The following shows of the DCL command that invokes DTSEND:

```
RUN SYS$SYSTEM:DTSEND
```

Command Parameters

This command has no command parameters.

Before invoking DTS, be sure that the necessary line (if any) is in the ON state.

To invoke DTS, type RUN SYS$SYSTEM:DTSEND in response to the DCL prompt.

DTS returns with the following information and prompt:

```
DTS Version xxx initiated on dd-mmm-yy hh:mm:ss
_Test:
```

DTS accepts input entered interactively at the terminal. DTS also accepts a command procedure as input.

To exit DTS and thereby terminate the test, enter EXIT at the _Test: prompt. For example:

```
_Test: EXIT
```

```
DTS terminated on dd-mmm-yy hh:mm:ss
$
```

Use the /PRINT qualifier to print test results. Also, you can use the /STATISTICS and /DISPLAY qualifiers to print statistics and messages transmitted to DTR, respectively. The /STATISTICS and /DISPLAY qualifiers apply only to the data and interrupt tests. See the individual commands for more information.

4.3. DTS Commands

This section explains how to use the four DTS tests. Examples are provided for each test.

CONNECT
DATA
DISCONNECT
HELP
INTERRUPT

CONNECT

CONNECT — Invokes a connection test.

Format

CONNECT   [/general-qualifier[,...]/command-specific-qualifier [,...]]

General Qualifiers

/NODENAME=node-id

Identifies the node (by name or address) on which DTR is to run. The default is the local node. Because you cannot use access control information as part of the node name, DTR runs on the default nonprivileged account on the remote node.

To run DTR on an account other than the default, you associate the account with the DTR object.

/PRINT

Instructs DTS to print (or log) test results.

/NOPRINT

Disables printing of test results. The default is /NOPRINT.

/SPEED=number

Indicates the line speed (bits per second) of the test line. DTS uses this qualifier for reporting statistics (DTS does not control line speeds). The default is /SPEED=1000000.

Command-Specific Qualifiers

/TYPE=test-type

Indicates the subtest for the connection test. There are two subtests:

ACCEPT	Indicates a connect accept test. The default is /TYPE=ACCEPT.
REJECT	Indicates a connect reject test.

/RETURN=return-option

Indicates the type of optional user data DTR returns. There are two possibilities:

STANDARD	Indicates standard user data.
RECEIVED	Indicates received user data.

/NORETURN

Disables return of user data. The default is /NORETURN.

Description

Connection tests evaluate the ability of the DECnet software to process connect, connect accept, and connect reject requests with and without optional user data.

Connection tests that you may perform include the following:

- Connect reject without user data

- Connect accept without user data

- Connect reject with 16 bytes of standard user data

- Connect accept with 16 bytes of standard user data

- Connect reject with received user data used as reject user data

- Connect accept with received user data used as accept user data

Examples

1. `_Test: CONNECT/TYPE=ACCEPT/RETURN=RECEIVED`

   This command invokes the connection test with the connect accept subtest.

   DTS attempts to connect to DTR on the local node (by default). Connect user data is to be returned as part of the test.

2. `_Test: CONNECT/NODENAME=TRNTO`

   This command invokes the connection test with the connect accept subtest.

   DTS attempts to connect to DTR on remote node TRNTO. DTR returns no optional user data.

DATA

DATA — Invokes a data test.

Format

DATA [/general-qualifier[,...]/command-specific-qualifier [,...]]

General Qualifiers

/NODENAME=node-id

Identifies the node (by name or address) on which DTR is to run. The default is the local node. Because you cannot use access control information as part of the node name, DTR runs on the default nonprivileged account on the remote node.

To run DTR on an account other than the default, you must associate the account with the DTR object.

/PRINT

Instructs DTS to print (or log) test results.

/NOPRINT

Disables printing of test results. The default is /NOPRINT.

/STATISTICS

Instructs DTS to print statistics on data and interrupt tests. The default is /STATISTICS.

/NOSTATISTICS

Disables the output of statistics.

/DISPLAY=number

Instructs DTS to print the specified number of bytes (in hexadecimal) of data and interrupt messages transmitted to DTR.

/NODISPLAY

Disables the display of data and interrupt messages that are transmitted to DTR.

The default is /NODISPLAY.

/SPEED=number

Indicates the line speed (in Mbps) of the test line. DTS uses this qualifier for reporting statistics (DTS does not control line speeds). The default number is 100 (100 Mbps).

The allowable values are 10, 100, 1000, and 10000 for 10 Mbps, 100 Mbps, 1 Gbps and 10 Gbps respectively.

Command- Specific Qualifiers

/TYPE=test-type

Indicates the subtest for the data test. There are four subtests:

SINK	Indicates a sink test. The default is /TYPE=SINK.

SEQ	Indicates a sequence test.
PAT	Indicates a pattern test.
ECHO	Indicates an echo test.

/SIZE=number

Indicates the data message length in bytes. The lower bounds are 0 for a sink or echo test; 4 for a sequence test; and 5 for a pattern test. The upper bound is 4096 bytes for all tests. The default is /SIZE=128.

/[test-duration]

Indicates the duration of the test. There are three units of time for this parameter:

/SECONDS=number
/MINUTES=number
/HOURS=number

The number specifies the duration of the test. The default unit is /SECONDS=30.

The maximum test duration is 3,600,000 seconds or 1000 hours.

Description

Data tests provide a full spectrum of test capabilities, ranging from data sink operations to data integrity checking. Data tests that you may perform include the following:

- Sink Test. DTR ignores all data received during this test. DTR performs neither sequence nor content validation.

- Sequence Test. Data messages transmitted by DTS to DTR include a 4-byte sequence number. During the sequence test, if a message is received out of sequence, DTR aborts the logical link and the test.

- Pattern Test. Data messages transmitted to DTR have both a sequence number and a standard data pattern. During the pattern test, if either the sequence number or the received data does not match the expected data, DTR aborts the logical link and the test.

- Echo Test. DTR transmits all data messages received back to DTS during this test. DTS performs data validity tests.

Examples

```
_Test: DATA/PRINT/TYPE=SEQ/SIZE=128/SECONDS=10
_DTS-S-NORMAL, normal successful completion
Test parameters:
Test duration (sec) 10
Target nodename ""
Line speed (baud) 1000000
Message size (bytes) 128
Summary statistics:
Total messages XMIT 788 RECV 0
Total bytes XMIT 100864
Messages per second 78.8
Bytes per second 10086
```

```
Line throughput (baud) 80691
_ Line utilization 8.0
```

This command invokes the data test with the sequence subtest. DTS sends messages to DTR on the local node where test information is to be printed. The message size is 128 bytes, and the duration of the test is 10 seconds.

DISCONNECT

DISCONNECT — Invokes a disconnection test.

Format

DISCONNECT [/general-qualifier[,...]/command-specific-qualifier [,...]]

General Qualifiers

/NODENAME=node-id

Identifies the node (by name or address) on which DTR is to run. The default is the local node. Because you cannot use access control information as part of the node name, DTR runs on the default nonprivileged account on the remote node.

To run DTR on an account other than the default, you must associate the account with the DTR object.

/PRINT

Instructs DTS to print (or log) test results.

/NOPRINT

Disables printing of test results. The default is /NOPRINT.

Command- Specific Qualifiers

/TYPE=test-type

Indicates the subtest for the disconnection test. There are two subtests:

SYNCHRONOUS	Indicates a synchronous disconnect test.
ABORT	Indicates a disconnect abort test. The default is /TYPE=ABORT.

/RETURN=return-option

Indicates the type of optional user data DTR returns. There are two possibilities:

STANDARD	Indicates standard user data.
RECEIVED	Indicates received user data.

/NORETURN

Disables return of user data. The default is /NORETURN.

Description

Disconnection tests evaluate whether DTS detects the difference between disconnection and abort sequences generated by DTR as well as the ability of DTS to receive the proper optional user data. Disconnection tests that you can perform include the following:

- Disconnect without user data

- Abort without user data

- Disconnect with 16 bytes of standard user data

- Abort with 16 bytes of standard user data

- Disconnect with received connect user data used as disconnect user data

- Abort with received connect user data used as abort user data

Examples

```
_Test: DISCONNECT/NODENAME=DENVER/TYPE=ABORT
```

This command invokes the disconnection test with the disconnect abort subtest.

DTS performs this test with DTR on remote node DENVER. No optional user data is to be returned.

HELP

HELP — The HELP command displays general information about DTS commands and parameters.

Format

HELP   [topic]

General Qualifiers

topic

Specifies DTS and DTR keyword that refers to a topic or subtopic on which you can obtain information.

Examples

```
_Test: HELP
Information available:
CONNECT DATA DISCONNECT DTS INTERRUPT
Topic? dts
DTS
DTS and DTR are cooperationg tasks that perform various functions to
exercise network software. DTR functions as a slave to DTS. DTS
initiates each test by issuing a connection request to DTR. DTS
passes parameters pertinent to the type of test request to DTR in
the optional data of the connection request. The user interface to
DTS consists of a group of commands:
CONNECT
```

```
DATA
DISCONNECT
INTERRUPT
You may abbreviate command keywords and qualifiers to four characters
or less (if unique).
Note that, when using DTS commands in a batch job, you must use uppercase
characters in the command.
Topic?
```

This command invokes HELP and displays information about DTS.

INTERRUPT

INTERRUPT — Invokes an interrupt test.

Format

INTERRUPT   [/general-qualifier[,...]/command-specific-qualifier [,...]]

General Qualifiers

/NODENAME=node-id

Identifies the node (by name or address) on which DTR is to run. The default is the local node. Because you cannot use access control information as part of the node name, DTR runs on the default nonprivileged account on the remote node.

To run DTR on an account other than the default, you must associate the account with the DTR object.

/PRINT

Instructs DTS to print (or log) test results.

/NOPRINT

Disables printing of test results. The default is /NOPRINT.

/STATISTICS

Instructs DTS to print statistics on data and interrupt tests. The default is /STATISTICS.

/NOSTATISTICS

Disables the output of statistics.

/DISPLAY=number

Instructs DTS to print data and interrupt messages transmitted to DTR.

/NODISPLAY

Disables the display of data and interrupt messages that are transmitted to DTR.

The default is /NODISPLAY.

/SPEED=number

> Indicates the line speed (bits per second) of the test line. DTS uses this qualifier for reporting statistics. (DTS does not control line speeds.) The default is /SPEED=1000000.

Command- Specific Qualifiers

/TYPE=test-type

> Indicates the subtest for the interrupt test. There are four subtests:

SINK	Indicates a sink test. The default is /TYPE=SINK.
SEQ	Indicates a sequence test.
PAT	Indicates a pattern test.
ECHO	Indicates an echo test.

/SIZE=number

> Indicates the data message length in bytes. The lower bounds are 0 for a sink or echo test; 4 for a sequence test; and 5 for a pattern test. The upper bound is 16 bytes for all tests. The default is /SIZE=16.

/[test-duration]

> Indicates the duration of the test. There are three units of time for this parameter:

> /SECONDS=number
> /MINUTES=number
> /HOURS=number

> The number specifies the duration of the test. The default unit is /SECONDS=30.

> The maximum test duration is 3,600,000 seconds or 1000 hours.

Description

As with data tests, interrupt tests provide a full range of test capabilities from data sink operations to data integrity checking. Interrupt tests that you can perform include the following:

- Sink Test. DTR ignores all interrupt data received during this test. DTR performs neither sequence nor content validation.

- Sequence Test. Interrupt messages transmitted by DTS to DTR include a 4-byte sequence number. During the sequence test, if a message is received out of sequence, DTR aborts the logical link and the test.

- Pattern Test. Interrupt messages transmitted to DTR have both a sequence number and a standard data pattern. During the pattern test, if either the sequence number or the received data does not match the expected data, DTR aborts the logical link and the test.

- Echo Test. DTR transmits all interrupt messages received back to DTS during this test. DTS performs data validity tests.

Examples

```
_Test: INTERRUPT/NODENAME=DALLAS/PRINT/TYPE=PAT/SIZE=16
_DTS-S-NORMAL, normal successful completion
Test parameters:
Test duration (sec) 30
Target nodename "DALLAS"
Line speed (baud) 1000000
Message size (bytes) 16
Summary statistics:
Total messages XMIT 2734 RECV 0
Total bytes XMIT 43744
Messages per second 91.1
Bytes per second 1458
Line throughput (baud) 11665
__ Line utilization 1.1
```

This command invokes the interrupt test with the pattern subtest. DTS sends interrupt messages to DTR on node DALLAS where test information is to be printed. The interrupt message size is 16 bytes, and the duration of the test is 30 seconds (by default).

Appendix A. DECnet Circuit and Line Devices

This appendix lists DECnet for OpenVMS circuit and line devices. NCP allows for management of the local node and for remote management of any DECnet node that you specify as the executor. The display that you see and the devices that you manage depend on the supported features of the particular DECnet system that you are managing.

Note that some of the devices shown may not be supported on the your system.

Check the Software Product Description for the latest device information.

Table A.1 shows the devices for Ethernet circuits and lines. Devices on this table conform to the Ethernet and/or IEEE 802.3 protocol.

Table A.1. Ethernet Circuit and Line Devices

Mnemonic	Driver	Description
BNA	ET	DEBNA, DEBNI communications link
ISA	EZ	SGEC communications link
KFE	EF	VAXft 3000 communications link
MNA	EX	DEMNA communications link
MXE	EC	PMAD communications link
QNA	XQ	DEQNA, DELQA, DESQA, DEQTA communications link
SVA	ES	DESVA communications link
UNA	XE	DEUNA, DELUA communications link

Table A.2 shows the devices for DDCMP and CI circuits and lines. DDCMP devices in this table conform to the DDCMP protocol. CI devices communicate using protocols defined by the System Communications Architecture (SCA).

Table A.2. DDCMP and CI Circuit and Line Devices

Mnemonic	Driver	Description
DIV	SD	DIV32, ISDN controller
DMB	SI	DMB32 synchronous line unit
DMC	XM	DMC11, DMR11 synchronous line units
DMF	XG	DMF32 synchronous line unit
DMP	XD	DMP11, DMV11 synchronous point-to-point or multipoint line device
DSB	SL	DSB32 synchronous line unit
DSF	SF	VAXft 3000 synchronous communications controller
DST	ZS	DST32, DSH32 synchronous device
DSV	SJ	DSV11 synchronous line interface

Mnemonic	Driver	Description
DSW	ZT	DSW-21, DSW-41, DSW-42 synchronous communication device
TT	NO	DZ11, DZ32, DZQ11, DZV11 asynchronous device
TX	NO	DMB32, DHB32, DHU11, DHV11, DHT32, DMF32 or DMZ32, DHQ11, DSH32,CXY08 asynchronous device
CI	CN	CI–780, CI–750, CIBCA, or CIBCI computer interconnect

Table A.3 shows the devices for X.25 circuits and lines. Devices on this table conform to the LAPB protocol.

Note

Although some systems do not support VAX P.S.I., you can manage remote VAX P.S.I. systems from the local node.

Table A.3. X.25 Circuit and Line Devices

Mnemonic	Driver	Description
DMB	SI	DMB32 synchronous line unit
DMF	XG	MF32 synchronous line unit
DPV	UE	DPV11–DA synchronous line interface
DSB	SL	DSB32 synchronous line unit
DSV	SJ	DSV11 synchronous line interface
DSW	ZT	DSW-21, DSW-41, DSW-42 synchronous communication devices
DUP	XW	DUP11–DA synchronous line interface
KMV	UV	KMV1A synchronous line interface combined with X.25 level 2 microcode
KMX	XS	KMS11–BD or –BE synchronous line interface combined with X.25 level 2 microcode
KMY	XN	KMS1P synchronous line interface combined with X.25 level 2 microcode

Table A.4 shows the devices for FDDI circuits and lines. Devices on this table conform to the FDDI protocol.

Table A.4. FDDI Circuit and Line Devices

Mnemonic	Driver	Description
FZA	FC	DEFZA FDDI communications link
MFA	FX	DEMFA FDDI communications link

Appendix B. Network Counter Summary

Following are descriptions of circuit, line, node, X.25 protocol module, and X.25 and X.29 server module counters. Where possible, the description of each counter includes the probable causes for the particular type of occurrence. In some cases, the counters correspond to network events. The events and event descriptions provide additional information relative to the specific occurrence. The individual counter descriptions state which counters correspond to events.

NCP allows for management of the local node and for remote management of any DECnet node that you specify as the executor. The displays, counters, and messages that you receive by issuing NCP commands originate at the executor node. This appendix shows all the information that DECnet for OpenVMS can display. The actual display you receive depends on the supported features of the particular DECnet system that you are managing.

To see what management features are supported on DECnet for OpenVMS, see Chapter 3.

NCP reports the contents of all circuit, line, node, and module counters in decimal notation. Counter content displays with an angle bracket (>) indicate that the counter has overflowed. In some cases where the counter has more than one error associated with it, the standard counter is expanded to include a reason.

The following examples illustrate the display of circuit and line counters.

```
NCP> SHOW KNOWN CIRCUITS COUNTERS
Known Circuit Counters as of 30-DEC-1989 20:06:19
  Circuit = DMC-0        >65534
  Seconds since last zeroed            11
  Terminating packets received          5
  Originating packets sent         0
  Terminating congestion loss        1769
  Transit packets received        1884
  Transit packets sent         0
  Transit congestion loss         26
  Circuit down         2
  Initialization failure         625
  Adjacency down         326
  Peak adjacencies        782495
  Bytes received         757263
  Bytes sent        16972
  Data blocks received        17740
  Data blocks sent         0
  Data errors outbound         54
  Data errors inbound, including:
              NAKs sent, header block check error
              NAKs sent, data field block check error          0
  Local buffer errors         0
  Remote buffer errors         16
  Local reply timeouts          0
  Remote reply timeouts
Circuit = UNA-0        >65534
  Seconds since last zeroed        241392
  Terminating packets received        248861
  Originating packets sent          0
  Terminating congestion loss        351707
```

```
   Transit packets received        351510
   Transit packets sent            12
   Transit congestion loss            3
   Circuit down            0
   Initialization failure          625
   Adjacency down          326
   Peak adjacencies        628570
   Data blocks sent     51239061
   Bytes sent      1199899
   Data blocks received    148469990
   Bytes received          0
   Unrecognized frame destination       1095
   User buffer unavailable


NCP> SHOW KNOWN LINE COUNTERS
Known Line Counters as of 30-DEC-1989 20:06:31
Line = DUP-0       >65534
   Seconds since last zeroed      1879643
   Bytes received      5349550
   Bytes sent      137572
   Data blocks received       117752
   Data blocks sent        19
   Data errors inbound          0
   Data errors outbound       >254
   Remote reply timeouts          0
   Local reply timeouts           0
   Remote buffer errors           7
   Remote process errors, including:
               FRMR sent, header format error         4
   Local process errors, including:
               FRMR received, header format error
Line = UNA-0       >65534
   Seconds since last zeroed      1240675
   Data blocks received       608195
   Multicast blocks received          0
   Receive failure    146028085
   Bytes received     78201969
   Multicast bytes received           0
   Data overrun        686344
   Data blocks sent        49646
   Multicast blocks sent          363
   Blocks sent, multiple collisions       384
   Blocks sent, single collision       9520
   Blocks sent, initially deferred    53774695
   Bytes sent      7160674
   Multicast bytes sent           6
   Send failure, including:
               Carrier check failed           6
   Collision detect check failure       7014
   Unrecognized frame destination        5
   System buffer unavailable      1148
   User buffer unavailable
```

B.1. Circuit Counters

This section contains a description of each circuit counter.

Adjacency down

This counter indicates a loss of connection to an adjacent node that may result from any of the following:

• Node listener timeout

• Invalid data received at a node listener

• Unexpected control (initialization or verification) message received

• Routing message received with a checksum error

• Node identification from a routing message or a Hello message that is not the one expected

• Hello message received indicating that connectivity became one-way

• Adjacency idled

Bytes received

This counter indicates the number of bytes of data received by the local node over the circuit. You can use this information together with the data blocks received counter to determine the inbound traffic load.

Bytes sent

This counter indicates the number of bytes of data sent by the local node over the circuit. You can use this information together with the data blocks sent counter to determine the outbound traffic load.

Circuit down

This counter indicates the number of failures-operator or software induced-for the circuit. These failures may include any number of hardware-, software-, or operator-caused problems. This counter corresponds to events 4.7-9 (circuit down).

Data blocks received

This counter indicates the number of data blocks received by the local node. You can use this information as a statistical base when evaluating the number of inbound data errors, remote reply timeouts, and local buffer errors.

Data blocks sent

This counter indicates the number of data blocks sent by the local node. You can use this information as a statistical base when evaluating the number of outbound data errors, local reply timeouts, and remote buffer errors.

Data errors inbound

This counter indicates the number of data errors that normally result from errors on the inbound communications channel to the local node. These errors are caused usually by a noisy communications line or a poorly functioning modem. This counter may include either or both of the following qualifiers if they contribute any of the following errors:

NAKs sent, header block check error
NAKs sent, data field block check error

Data errors outbound

This counter indicates the number of data errors that normally result from errors on the outbound communications channel from the local node. These errors are caused usually by a noisy communications line or a poorly functioning modem. This counter represents the sum of all NAKs received.

Initialization failure

This counter indicates the number of times the circuit failed to initialize with remote Routing software. These failures may include any number of hardware-, software-, or operator-caused problems. This counter corresponds to events 4.11-13 (initialization failure).

Local buffer errors

This counter indicates the number of local buffer errors that normally result from a local receive buffer that is too small. These errors are caused when a remote node sends a block that is too big for the local node's receive buffer. This counter may include the following qualifier if it contributes to the error:

NAKs sent, buffer unavailable

This counter corresponds to event 5.9 (local buffer too small) and event 4.3 (oversized packet loss).

Local reply timeouts

This counter indicates the number of timeouts that normally result from the following:

- The loss of communication between nodes while the local node has data to transmit

- An inappropriate value set for the reply timer on the local node

Locally initiated resets

This counter indicates the number of resets sent over the circuit.

Network initiated resets

This counter indicates the number of resets originated by the PSDN received over the circuit.

Originating packets sent

This counter indicates the number of data packets sent by the Routing layer on the local node.

Peak adjacencies

This counter indicates the maximum number of nodes that are up on a circuit.

Remote buffer errors

This counter indicates the number of remote buffer errors that normally result from a remote receive buffer that is too small.

Remote reply timeouts

This counter indicates the number of timeouts that normally result from the following:

- The loss of communication between nodes while the remote node has data to transmit

- An inappropriate value set for the reply timer on the remote node

Remotely initiated resets

This counter indicates the number of resets received over the circuit.

Seconds since last zeroed

This counter indicates the number of seconds that have elapsed since the circuit counters were zeroed. This counter provides a time frame for other counter values. The software increments this counter every second and clears it when other counters are cleared.

Selection intervals elapsed

This counter indicates the number of selection intervals for half-duplex and multipoint control stations. This is the number of times a station selects another station. You can use this information as a statistical base when evaluating selection timeout errors.

Selection timeouts

This counter indicates the number of selection timeouts that normally result from the following:

- Loss of communication with a remote station.

- Data errors on the communications channel to or from that station.

- An inappropriate value for the station's select timer. This counter corresponds to event 5.5 (select error threshold) and may include either the *No reply to select* or the *Incomplete reply to select* qualifiers if they contribute to the errors.

Terminating congestion loss

This counter indicates the number of packets intended for the node that were discarded because Routing could not buffer them.

Terminating packets received

This counter indicates the number of data packets received by the Routing layer on the local node.

Transit congestion loss

This counter indicates the number of transit data packets discarded for congestion reasons. This counter is maintained only on full-routing nodes. If congestion loss increases, increase the MAXIMUM BUFFERS parameter for the local node.

Transit packets received

This counter indicates the number of data packets received over the circuit and to be routed through the local node to another node. It is maintained only on full-routing nodes.

Transit packets sent

This counter indicates the number of data packets sent over the circuit and being routed through the local node to another node. It is maintained only on full-routing nodes.

Unrecognized frame destination

This counter indicates the number of times a frame was discarded because the frame was addressed to a multicast address not enabled on the circuit.

User buffer unavailable

This counter indicates the total number of times that no user buffer was available for an incoming frame that passed all filtering. User buffers are supplied by users on receive requests.

B.2. Line Counters

This section contains a description of each line counter. The counters in this section may be displayed for devices other than the DMC11 on remote nodes other than DECnet-VAX.

Blocks sent, initially deferred

This counter indicates the total number of times a frame transmission was deferred on its first transmission attempt. Used in measuring Ethernet contention with no collisions.

Blocks sent, multiple collisions

This counter indicates the total number of times a frame was successfully transmitted on the third or later attempt after normal collisions on previous attempts.

Blocks sent, single collision

This counter indicates the total number of times a frame was successfully transmitted on the second attempt after a normal collision on the first attempt.

Bytes received

This counter indicates the number of bytes of data received over the line. Applies to Ethernet and VAX P.S.I. operations.

Bytes sent

This counter indicates the number of bytes of data sent over the line. Applies to Ethernet and VAX P.S.I. operations.

Collision detect check failure

This counter indicates the approximate number of times a collision detect was not sensed after a transmission. This counter is also reported each time a packet is transmitted using either a DEUNA or DELUA on a broadband Ethernet.

Data blocks received

This counter indicates the number of data blocks received over the line. Applies to Ethernet and VAX P.S.I. operations.

Data blocks sent

This counter indicates the number of data blocks sent over the line. Applies to Ethernet and VAX P.S.I. operations.

Data errors inbound

This counter indicates the number of incoming data errors that result from faults on the channel between the local DTE and DCE. The counter can include any of the following qualifiers:

Block too long
Block check error
Reject sent

Data errors outbound

This counter indicates the number of outgoing data errors that result from faults on the channel between the local DTE and DCE. The counter can include the following qualifier:

Reject received

Data overrun

This counter indicates the total number of times the hardware lost an incoming frame because it was unable to keep up with the data rate.

Local buffer errors

This counter indicates the number of receive-not-ready (RNR) frames sent. The counter can include the following qualifier:

RNR sent, buffer unavailable

Local process errors

This counter indicates errors for both VAX For VAX P.S.I. operations, this counter indicates that a frame reject (FRMR) has been received over the line or that your system is being overloaded. The counter can include any of the following qualifiers:

Transmit underrun
Receive overrun
FRMR received, header format error

The first two qualifiers usually indicate that the system is overloaded and the third usually indicates that the VAX P.S.I. software may be incorrectly configured.

For DDCMP operations, this counter indicates the number of local DDCMP processing errors. DECnet-VAX maintains this counter only for DMP and DMF lines. For a multipoint circuit, this counter reflects the total errors for the control station, and not for each tributary. These errors may be caused by a programming error at the local node, or by a circuit error that was not detected by the Data Link protocol. This counter may include any of the following qualifiers if they contribute to the errors:

NAKs sent, receive overrun
Receive overrun, NAK not sent
Transmit overruns
NAKs received, header format error

For DDCMP lines, this counter corresponds to events 5.3-4 and 5.6.

Local reply timeouts

This counter indicates the number of times a frame with a poll bit set has been received over the line, that is, the number of errors that result from faults on the line.

Multicast blocks received

This counter indicates the total number of multicast blocks successfully received.

Multicast bytes received

This counter indicates the total number of multicast data bytes successfully received (includes bytes in Ethernet data field but not the Ethernet data link headers).

Multicast blocks sent

This counter indicates the total number of multicast blocks successfully sent.

Multicast bytes sent

This counter indicates the total number of multicast data bytes successfully sent (including bytes in the Ethernet data field, but not in the Ethernet data link headers).

Receive failure

This counter indicates the total number of blocks received with some data error (the blocks are data frames) that passed either physical or multicast address comparison. For each increment of the counter, a type of failure is recorded, as follows:

Block check error
Framing error
Frame too long

Remote buffer errors

This counter indicates the number of receive-not-ready (RNR) frames received. The counter can include the following qualifier:

RNR received, buffer unavailable

Remote process errors

This counter indicates errors for both VAX P.S.I. and DDCMP lines. For VAX P.S.I. operations, this counter indicates an invalid n (R) and a frame reject (FRMR) sent over the line. The counter can include the following qualifiers:

Invalid n(R) received
FRMR sent, header format error

These errors usually indicate that the DCE is functioning incorrectly.

For DDCMP operations, this counter indicates the number of remote DDCMP processing failures. DECnet-VAX maintains this counter only for DMP and DMF lines. For a multipoint circuit, this counter reflects the total errors for the control station, and not for each tributary. These errors may be caused by a programming error or a hardware malfunction at the adjacent node. They could also be caused by a circuit error that was not detected by the Data Link protocol. This counter may include any of the following qualifiers if they contribute to these errors:

NAKs received, receive overrun
NAKs sent, header format error
Selection address errors
Streaming tributaries

For DDCMP lines, this counter corresponds to events 5.4 and 5.6-8.

Remote reply timeouts

This counter indicates the number of times the retransmit timer for that line has expired. Either the line is not connected to a modem, the PSDN is not responding fast enough, or the retransmit timer is set too low.

Seconds since last zeroed

This counter indicates the number of seconds that elapsed since the line counters were zeroed. This counter provides a time frame for other counter values. The software increments this counter every second and clears it when other counters are cleared.

Send failure

This counter indicates the total number of times a transmit attempt failed. For each increment of the counter, a type of failure is recorded, as follows:

Excessive collisions
Carrier check failed
Short circuit
Open circuit
Frame too long
Remote failure to defer

The "short circuit" failure is also reported each time a packet is transmitted using a DEQNA with a DECOM transceiver.

System buffer unavailable

This counter indicates the total number of times no system buffer was available for an incoming frame. This can be any buffer between the hardware and the user buffers (those supplied on receive requests).

Unrecognized frame destination

This counter indicates the number of times a frame was discarded because there was no portal with the protocol enabled. The count includes frames received for the physical address only. It does not include frames received for the multicast or broadcast address.

User buffer unavailable

This counter indicates the total number of times no user buffer was available for an incoming frame that passed all filtering. The user buffer is one supplied by the user on a receive request.

B.3. Node Counters

This section contains a description of each node counter.

Aged packet loss

This counter indicates the number of data packets discarded for visiting too many nodes. This usually occurs while the databases throughout the network are recovering from a disruption (for example, when a circuit or line goes down) in the former path to a destination. This counter is maintained only on full-routing nodes and corresponds to event 4.0 (aged packet loss).

Buffer unavailable

This counter indicates the number of data segments transferred from the remote node that were discarded because of insufficient cache buffering.

Bytes received

This counter indicates the number of user data bytes received from a remote node. This includes interrupt messages, but excludes Connect, Accept, Reject, and Disconnect messages.

Bytes sent

This counter indicates the number of user data bytes sent to a remote node.

Connects received

This counter indicates the number of logical link connection requests received by the local node.

Connects sent

This counter indicates the number of logical link connection requests sent by the local node.

Maximum logical links active

This counter indicates the largest number of logical links that have been active since DECnet-VAX software was started or since executor counters were zeroed.

Messages received

This counter indicates the number of NSP messages received from a remote node.

Messages sent

This counter indicates the number of NSP messages sent to a remote node.

Node out-of-range packet loss

This counter indicates the number of data packets discarded because the destination node's address is greater than the maximum address defined for the local node. This counter corresponds to event 4.2 (node out-of-range packet loss).

Node unreachable packet loss

This counter indicates the number of data packets lost because the destination node could not be accessed. This counter is maintained only on full-routing nodes. This counter corresponds to event 4.1 (node unreachable packet loss).

Oversized packet loss

This counter indicates the number of received data packets that were too large to forward because of the block size of the data link that would be used. This counter is maintained only on full-routing nodes, and corresponds to event 4.3 (oversized packet loss).

Packet format error

This counter indicates the number of packet format errors that occur because of invalid packet control information. This counter corresponds to event 4.4 (packet format error).

Partial routing update loss

This counter indicates the number of received routing messages that were too long to process. Part of a routing update may be lost if it contains a reachable node with an address greater than the maximum address defined for the local node. This counter is maintained only on full-routing nodes, and corresponds to event 4.5 (partial routing update loss).

Received connect resource errors

This counter indicates the number of inbound connect messages for which the local node did not have sufficient resources. These errors may result from dynamic memory problems or too few logical link slots; that is, the MAXIMUM LINKS parameter value is too small.

Response timeouts

This counter indicates the number of times there was no response to an NSP segment within the allotted timeout period. This implies that the local node is having to retransmit messages. Such retransmission can be caused either by messages being discarded in the network or by a wide variance in the round-trip delay to the node. Normally, it indicates an overload condition in the network.

Seconds since last zeroed

This counter indicates the number of seconds that have elapsed since the node counters were zeroed. It provides a time frame for other counter values. The software increments this counter every second and clears it when other counters are cleared.

Verification reject

This counter indicates the number of received verification messages that were invalid. It corresponds to event 4.6 (verification reject).

B.4. X.25 Protocol Module DTE Counters

This section contains a description of each X.25 protocol module DTE counter. These counters apply only to VAX P.S.I. operations.

Bytes received

This counter indicates the number of bytes of data received by the local DTE. You can use this information together with the data blocks received counter to determine the total traffic load.

Bytes sent

This counter indicates the number of bytes of data sent by the local DTE. You can use this information together with the data blocks sent counter to determine the total traffic load.

Calls received

This counter indicates the number of incoming calls received.

Calls sent

This counter indicates the number of outgoing calls made.

Data blocks received

This counter indicates the number of data blocks received by the local DTE.

Data blocks sent

This counter indicates the number of data blocks sent by the local DTE.

Fast selects received

This counter indicates the number of calls received with the fast select facility specified.

Fast selects sent

This counter indicates the number of calls sent with the fast select facility specified.

Locally initiated resets

This counter indicates the number of resets sent by the local DTE.

Maximum channels active

This counter indicates the maximum number of channels from the logical channels list that were active at any one time since the counters were last zeroed.

This counter applies only to those channels defined by the CHANNELS parameter of the SET MODULE X25-PROTOCOL command.

Maximum switched circuits active

This counter indicates the number of switched virtual circuits that were active at any one time since the counters were last zeroed.

Network initiated resets

This counter indicates the number of resets (originated by the PSDN) received by the local DTE.

Received call resource errors

This counter indicates the number of times an incoming call has been rejected because of insufficient resources or an incorrect configuration (for example, no destination or object specified).

Remotely initiated resets

This counter indicates the number of resets (originated by a remote DTE) received by the local DTE.

Restarts

This counter indicates the number of times the restart protocol procedure was used on the DTE.

Seconds since last zeroed

This counter indicates the number of seconds that have elapsed since the module counters were zeroed. This counter provides a time frame for other counter values. The VAX P.S.I. software increments this counter every second and clears it when the counters are zeroed.

B.5. X.25 and X.29 Server Module Counters

This section contains a description of each X.25 and X.29 server module counter. These counters apply only to VAX P.S.I. operations.

Incoming calls rejected, no resources

This counter indicates the number of times the incoming call handler rejected a request to set up a virtual circuit because of insufficient resources. It includes calls rejected by security for the X.25 server module.

Maximum circuits active

This counter indicates the number of switched virtual circuits that have been set up since the counters were last zeroed.

Seconds since last zeroed

This counter indicates the number of seconds that have elapsed since the module counters were zeroed. This counter provides a time frame for other counter values. The VAX P.S.I. software increments this counter every second and clears it when the counters are zeroed.

Appendix C. DECnet Circuit and Line Devices

Table C.1 lists all recognized DECnet circuit and line devices. Note that the DECnet-VAX system does not support all of the devices in this table.

Table C.1. DECnet Circuit and Line Devices

Mnemonic	Multi- plexed	Multi- access	DECnet-VAX Support	VAX P.S.I. Support	Description
BNA	No	Yes	Yes	No	DEBNA multiaccess communications link
CI	No	Yes	Yes	No	CI-780, CI-750, CIBCA, or CIBCI computer interconnect
DA	No	No	No	No	DA11-B or DA11-AL UNIBUS link
DL	No	No	No	No	DL11-C, DL11-WA, and DL11-E asynchronous line interface
DLV	No	No	No	No	DLV11-E asynchronous line interface
DMB	No	No	Yes	Yes	DMB32 synchronous line unit
DMC	No	No	Yes	No	DMC11-DA/AR, DMC11-MA/AL, DMC11-MD/AL, DMC11-FA/AR interprocessor links; DMR11-AA, DMR11-AB, DMR11-AC, DMR11-AE interprocessor links

Mnemonic	Multi- plexed	Multi- access	DECnet-VAX Support	VAX P.S.I. Support	Description
DMF	No	No	Yes	Yes	DMF32 synchronous line unit
DMP	No	Yes	Yes	No	DMP11 multipoint synchronous line device, DMV11 multipoint synchronous line device
DP	No	No	No	No	DP11-DA synchronous line interface
DPV	No	No	No	Yes	DPV11-DA synchronous line interface
DQ	No	No	No	No	DQ11-DA synchronous serial line interface
DTE	No	No	No	No	DTE20 interprocessor link
DU	No	No	No	No	DU11-DA synchronous line interface (includes DUV11)
DUP	No	No	No	Yes	DUP11-DA synchronous line interface
DV	Yes	No	No	No	DV11-AA/ BA NPR synchronous line multiplexer
KDP	Yes	No	No	No	KMC11/ DUP11- DA NPR synchronous line multiplexer
KDZ	Yes	No	No	No	KMC11/ DZ11-A, -B, -C, -D NPR asynchronous line multiplexer

Mnemonic	Multi- plexed	Multi- access	DECnet-VAX Support	VAX P.S.I. Support	Description
KL	No	No	No	No	KL8-J serial line interface
KMV	No	No	No	Yes	KMV1A synchronous line interface combined with X.25 level 2 microcode
KMX	Yes	No	No	Yes	KMS11-BD or -BE synchronous line interface combined with X.25 level 2 microcode
KMY	No	No	No	Yes	KMS1P synchronous line interface combined with X.25 level 2 microcode
QNA	No	Yes	Yes	No	DEQNA or DELQA multiaccess communications link
SVA	No	Yes	Yes	No	DESVA multiaccess communications link
TT	Yes	No	Yes	No	DZ11-F, DZ32-F, DZV11-D, or DZQ-11 asynchronous serial line multiplexer
TX	No	No	Yes	No	DHU11, DHV11, DMF32, or DMZ32 asynchronous line unit
UNA	No	Yes	Yes	No	DEUNA or DELUA multiaccess communications link

Appendix D. Event Class and Type Summary

Following is a summary of NCP events in terms of their class and type. In general, event classes relate to specific layers of the DECnet architecture. The event logging components support the event classes summarized.

Event Class	Description
0	Network Management layer
1	Applications layer
2	Session Control layer
3	End Communications layer
4	Routing layer
5	Data Link layer
6	Physical Link layer
7	X.25 packet level events
8-31	Reserved for other common classes
32-63	RSTS system specific
64-95	RSX system specific
96-127	TOPS 10/20 system specific
128-159	VMS system specific
160-191	RT system specific
192-223	CT system specific
224-255	Communication server specific
256-352	Reserved for future use
353	Distributed Name Service (DNS) clerk events
354-479	Reserved for future use
480-511	Customer specific

DECnet-VAX logs events only for event classes 0, 3, 4, and 128-159. If you attempt to turn on logging for any other event, it does not get logged. All other classes are supported for logging of processed events from other remote nodes. Events in class 7 are logged only on nodes that have VAX P.S.I. installed. Events in class 353 are logged only in networks where the Distributed Name Service (DNS) software is running.

If the logging sink is the monitor and no name is specified, DECnet-VAX uses the OPCOM facility to display event messages on all terminals enabled as NETWORK (by means of REPLY/ ENABLE=NETWORK), including the console. Event messages have the following format:

```
 event type class.type, event-text from node address [(node-name)] dd-mmm-
yy hh:mm:ss.ms component-type, event-qualifiers,...
```

The event text is a standard text message for each event class and type. The message format also includes the source node (address and node name, if available) and time stamp for when the event occurred. For

most events, the message format includes the component type and name for which the event applies. Finally, the message format may include one or more event qualifiers that provide additional information as to the cause of the event. The following example is a typical message that may be displayed on the console:

```
    Opcom, 22:45:51.19, SYSTEM        Accnt=
    Opcom, DECnet event 4.7, line down, line fault
From node 5 (TRNTO), 30-DEC-1989 22:45:51.13
Line DMC-0,0
Line synchronization lost
```

The information accompanying each event description in Section D.1 includes a brief description of event qualifiers that may be displayed for the event.

Note that OPCOM does not display line or node counters on the terminal device. However, they are written to the console sink.

D.1. Network Management Layer Events

The following specific event classes and types are supported for each layer. Only those events and entire event classes marked with an asterisk (*) are logged by DECnet-VAX components. Only those events and entire event classes marked with an asterisk (*) are logged by DECnet-VAX components.

0.0 Event records lost*

Events occurred too rapidly for the event logger to buffer them.

0.1 Automatic node counters

A node counter timer expired, thus prompting this event. This message displays the name of the node to which the event applies, along with the node counters for that node.

0.2 Automatic line counters

A line counter timer expired, thereby producing this event. This message displays the name of the line to which the event applies, along with the line counters for that line.

0.3 Automatic circuit service

An adjacent node requested an automatic circuit service operation. This message displays the name of the circuit to which the event applies, along with the following event qualifiers: the service function performed (load or dump), the status of the operation (requested, successful, or failed), the node address, the file specification, and the software type. If the operation fails, this status includes an NML error message and details.

0.4 Line counters zeroed

Line counters were zeroed. This message displays the name of the line to which the event applies. The event logger logs these counters prior to the execution of a request to zero them.

0.5 Node counters zeroed

Node counters were zeroed. This message displays the name of the node to which the event applies. The event logger logs these counters prior to the execution of a request to zero them.

0.6 Passive loopback*

The software initiated or terminated a passive loopback test on behalf of an adjacent node. This message displays the name of the line to which the event applies, along with one event qualifier that specifies the state of the operation (initiated or terminated).

0.7 Aborted service request*

An adjacent node requested a service over a line connected to the local node. However, a problem prevented it from being processed at the local node. This message displays the name of the line to which the event applies, along with one event qualifier that specifies the reason for the failure. The reason may be one of the following:

- Line open error

    NML received an MOP message and was unable to acquire control of the line. Either NML did not have the privilege to perform the operation or it could not set the substate of the line; or the line had another owner.

- Line state change by higher level

    The line was preempted by a higher priority function. For example, you used NCP to turn off the line.

- Receive error

    A line error occurred while the Network Management layer was trying to receive the request.

- Receive timeout

    The line message receive timer expired before the request could be received from the adjacent node. Either the timer was too short, the line error level was too great for any message to get through, or the adjacent node stopped requesting.

- Unrecognized request

    A message was received but was not recognizable as a request for upline dumping, downline loading, or passive loopback testing. The adjacent node may be running an incompatible version of the line service protocol.

0.8 Automatic counters*

A counter timer for a node, circuit, or line has expired, thereby producing this event. This message displays the name of the component to which the event applies, along with the counters for that line.

0.9 Counters zeroed*

Counters were zeroed for the node, circuit, or line. This message displays the name of the component to which the event applies. The event logger logs these events prior to the execution of a request to zero them.

D.2. Session Control Layer Events

The following specific event classes and types are supported for the session control layer.

2.0 Local node state change

The operational state of the local node changed because of an operator command. Note, however, that the transition from SHUT to OFF also happens automatically when the last logical link is disconnected (under normal operation).

This message displays three event qualifiers: the reason for the state change (operator command or normal operation), the old state (ON, OFF, SHUT, or RESTRICTED), and the new state.

2.1 Access control reject

The local node rejected a connection request because of invalid access control information.

This message displays four event qualifiers: the name and address of the source node, the object type number and process ID of the source process requesting the connection, the object type number and process ID of the destination process to receive the connection request, and the invalid access control information.

D.3. End Communications Layer Events

The following specific event classes and types are supported for the End Communications layer. The Network Services Protocol (NSP) is the protocol that operates within the End Communications layer. Only those events and entire event classes marked with an asterisk (*) are logged by DECnet-VAX components. Entire event classes marked with an asterisk (*) are logged by DECnet-VAX components.

3.0 Invalid message*

NSP received a message that could not be interpreted. This may indicate a software malfunction in either the local or remote NSP. This message displays one event qualifier - the NSP message that was invalid. Refer to the *Network Services Protocol Functional Specification* for a description of NSP messages.

3.1 Invalid flow control*

The remote NSP attempted to modify the local flow control value in an invalid manner. This may indicate a software malfunction in either the local or remote NSP. This message displays one event qualifier - the current flow control value. Refer to the *Network Services Protocol Functional Specification* for a description of flow control.

3.2 Database reused

The local node received a connection request from a node for which there is no counter block. All counter blocks have been previously used, and one of the previously used blocks is available for this new node. This results in the loss of node counters for the node that formerly occupied the database entry.

This message displays the name of the node for which the database entry was formerly used, along with the node counters for that node.

D.4. Routing Layer Events

The following specific event classes and types are supported for the Routing layer. Only those events and entire event classes marked with an asterisk (*) are logged by DECnet-VAX components. layer. Only those events and entire event classes marked with an asterisk (*) are logged by DECnet-VAX components.

4.0 Aged packet loss*

Routing discarded a packet because it had visited too many nodes. This can be a normal occurrence when the network is reconfiguring its routing databases. It can be a failure when the MAXIMUM HOPS value is set too small. This can cause the MAXIMUM VISITS value to be too small for a path that should be usable.

This message displays the name of the line to which the event applies, along with one event qualifier - the packet header. This is information from the beginning of the packet. For non-Ethernet packets, it consists of a hexadecimal byte of flags, the decimal destination and source node addresses, and a hexadecimal byte of forwarding data. For Ethernet packets, it also includes the Ethernet address of the destination and source, the service type, and the protocol type. Refer to the *Routing Layer Functional Specification* for additional information.

4.1 Node unreachable packet loss*

Routing discarded a packet because the local node found that the destination node was unreachable. This event provides a trace of what has happened to packets that are not reaching their destination.

This message displays the name of the line to which the event applies, along with one event qualifier - the packet header (as described for event 4.0).

4.2 Node out-of-range packet loss*

Routing discarded a packet because the destination node number was greater than the maximum node number known to the local node. Typically, this results from the addition of a new node to the network without increasing the MAXIMUM ADDRESS value on the local node, yet expecting the local node to route packets to that node.

This message displays the name of the line to which the event applies, along with one event qualifier - the packet header (as described for event 4.0).

4.3 Oversized packet loss*

Routing discarded a packet because it was too large to forward to the appropriate adjacent node. Typically, this occurs when the adjacent node's buffer size is too small or when the source node sends a packet that is too large.

This message displays the name of the line over which the packet was to be forwarded, along with one event qualifier - the packet header (as described for event 4.0).

4.4 Packet format error*

Routing discarded a packet because of a format error in the packet header. This usually results from a programming error in the packet formatting by the adjacent node, though it could result from a line error that was not detected by the line protocol.

This message displays the name of the line to which the event applies, along with one event qualifier - the packet beginning. This consists of the first 6 bytes of the packet, displayed as hexadecimal.

4.5 Partial routing update loss*

Routing received a routing message that contained node addresses greater than the maximum address known to the local node. Subsequently, information on these nodes was lost. This occurs when the

MAXIMUM ADDRESS value on an adjacent node has been increased to accommodate more nodes, but the local node's has not.

This message displays the name of the line over which this message was received, along with two event qualifiers - the packet header (as described for event 4.0) and the highest node address in the routing update that was lost.

4.6 Verification reject*

An attempt to initialize with another node failed. The local node received an invalid password in the verification requested of the adjacent node during routing initialization over the line. Either the local node expected the wrong receive password, or the adjacent node sent the wrong transmit password.

This message displays the name of the line to which the event applies, along with one event qualifier - the address of the adjacent node that failed to initialize.

4.7 Circuit down, circuit fault*

An error has occurred for the circuit. This message displays the name of the circuit to which the event applies, along with one event qualifier - the reason for the event. The reason could be one of the following:

- Adjacent node address change

  The adjacent node changed addresses without going through the normal initialization sequence. This is also logged when an adjacent node attempts to initialize with the local node, but the address of the adjacent node is not in the database.

- Adjacent node address out of range

  The adjacent node's address is greater than the maximum address defined for the local node. This may be caused by an incorrectly defined node address or by a failure to update the local node's database when a new node was added.

- Adjacent node block size too small

  The line block size provided by the adjacent node is too small for normal network operation. The block size may be set incorrectly at the adjacent node.

- Adjacent node listener receive timeout

  The node has received no message over the data link within the last 30 seconds. This usually means that the remote node is not running.

- Adjacent node listener received invalid data

  A test message sent by the adjacent node contained invalid or corrupted data. This is most likely caused by a hardware problem.

- Call failed

  An outgoing SVC call failed. This is an X.25 event.

- Data errors

  The line was declared down by the line protocol of the local node handler when the line exceeded an error threshold.

- Dropped by adjacent node

  The adjacent node was responsible for breaking the circuit connection.

- Invalid verification seed value

  A Routing initialization message sent by an adjacent node is not formatted properly. This is most likely caused by a remote network software problem.

- Line synchronization lost

  The normal line protocol was restarted or terminated by the adjacent node. Either a line exceeded an error threshold, or network management initiated a line state change. DMR/DMC failures that cause a line synchronization error are as follows:

  - Threshold errors, including more than eight attempts to transmit a message, or eight NAKs received in a row.

  - Start message received in the ON state (that is, the remote system detected an error and restarted the line).

  - Maintenance requested while in the ON state (that is, the remote system tried to perform a maintenance operation such as LOOP CIRCUIT).

  - Message was lost because no buffer was available in CPU memory.

  - Nonexistent memory error.

  - Procedure error, because of driver failure or hardware failure.

  - Timeout on request to transmit a message in 255 seconds.

  - Power failure.

- Routing update checksum error

  A routing update packet failed its internal integrity test.

- Unexpected packet type

  A packet was received out of the normal protocol sequence. For example, the local node received a normal data packet when it expected a verification packet.

- Verification password required from Phase III node

  A required routing initialization password was not specified before an attempt was made to initialize the Phase III node in a Phase IV network.

- Verification receive timeout

  A required verification packet was not received from the adjacent node within the required response time. Either packets were lost on the line or a failure occurred at the adjacent node.

- Version skew

  The routing version of the adjacent node is unacceptable to the local node. The operator may have installed incorrect software at the adjacent node.

4.8 Circuit down*

An error has occurred for the circuit. This message displays the name of the circuit to which the event applies, along with the following event qualifiers: the packet header (as described for event 4.0), the reason (as described for event 4.7), and the address of the adjacent node.

4.9 Circuit down, operator initiated*

An operator error has occurred for the circuit. This message displays the name of the circuit to which the event applies, along with the following event qualifiers - the packet header (as described for event 4.0), the reason (as described for event 4.7), and the addresses of the expected node and the adjacent node.

4.10 Circuit up*

A remote node has initialized on one of the physical lines connected to the local node. This message displays the name of the line to which the event applies, along with one event qualifier - the address of the newly initialized node.

Be sure to note that this event does not imply that the node is reachable. Reachability is determined by the higher-level routing algorithms.

4.11 Initialization failure, line fault*

A remote node failed to initialize with the local node because of a physical line error. This message displays the name of the line to which the event applies, along with one event qualifier - the reason for the event (as described for event 4.7).

4.12 Initialization failure*

A remote node failed to initialize with the local node because of a software error. This message displays the name of the line to which the event applies, along with two event qualifiers - the packet header (as described for event 4.0) and the reason (as described for event 4.7).

4.13 Initialization failure, operator initiated*

A remote node failed to initialize with the local node because of an operator error. This message displays the name of the line to which the event applies, along with three event qualifiers - the packet header (as described for event 4.0), the reason (as described for event 4.7), and the version received from the adjacent node.

4.14 Node reachability change*

Because of Routing operation, the reachability of a remote node has changed. This message displays the name of the node to which the event applies, along with one event qualifier - the routing status of the node (reachable or unreachable).

4.15 Adjacency up*

The adjacent node on the circuit is initialized. This message displays the name of the circuit to which the event applies, and one event qualifier - the address of the adjacent node.

4.16 Adjacency rejected*

The adjacent node on the circuit is not initialized. This message displays the name of the circuit to which the event applies, and two event qualifiers - the address of the adjacent node and the reason for the event (as described for event 4.7).

4.17 Area reachability change*

Because of Routing operation, the reachability of an area has changed. This message displays the name of the area to which the event applies, along with one event qualifier - the routing status of the area (reachable or unreachable).

4.18 Adjacency down*

An error has occurred for an adjacency on the circuit. This message displays the name of the circuit to which the event applies, along with the following event qualifiers - the reason (as described for event 4.7), the packet header (as described for event 4.4), and the address of the adjacent node on the circuit.

4.19 Adjacency down, operator initiated*

An adjacency on the circuit is down because of an operator error. This message displays the name of the circuit to which the event applies, along with the following event qualifiers - the reason (as described for event 4.7), the packet header (as described for event 4.0), and the addresses of the expected node and the adjacent node on the circuit.

D.5. Data Link Layer Events

The following specific event classes and types are supported for the Data Link layer. Only those events and entire event classes marked with an asterisk (*) are logged by DECnet-VAX components. (*) are logged by DECnet-VAX components.

5.0 Locally initiated state change

The line state changed because of an operator command. This message displays the name of the line to which the event applies, along with two event qualifiers - the old DDCMP state (HALTED, ISTRT, ASTRT, RUNNING, or MAINTENANCE) and the new DDCMP state. Refer to the *Data Communications Message Protocol Functional Specification* for a description of these states.

5.1 Remotely initiated state change

A remote user changed the line state. This message displays the name of the line to which the event applies, along with two event qualifiers (see event 5.0).

5.2 Protocol restart received in maintenance mode

The remote node restarted normal operation while the local node had the line in maintenance mode. This message displays the name of the line to which the event applies.

5.3 Send error threshold

Too many data transmission errors occurred. This message displays the name of the line to which the event applies, along with the line counters for that line and one event qualifier - the address of the received station (node).

5.4 Receive error threshold

Too many data reception errors occurred. This message displays the name of the line to which the event applies, along with the line counters for that line and one event qualifier (see event 5.3).

5.5 Select error threshold

Too many selection errors occurred. This message displays the name of the line to which the event applies, along with the line counters for that line and one event qualifier (see event 5.3).

5.6 Block header format error

DDCMP received an invalid block header. This message displays the name of the line to which the event applies, along with one event qualifier - the invalid block header. Refer to the *Data Communications Message Protocol Functional Specification* for a description of the block header format.

5.7 Selection address error

The wrong tributary responded in the polling process. This event occurs only for a multipoint control station when one receives a message that does not match the address of the currently selected tributary.

This message displays the name of the line to which the event applies, along with three event qualifiers - the tributary addresses of the selected tributary, the received tributary, and the previous tributary.

5.8 Streaming tributary

A tributary on the line is impeding the use of that line. This message displays the name of the line to which the event applies, along with two event qualifiers - the tributary address of the received tributary and the status of the tributary. The status may be any of the following:

- Streaming

- Continued send after timeout

- Continued send after deselect

- Ended streaming

5.9 Local buffer too small

A local buffer is too small for receiving a block of data. This message displays the name of the line to which the event applies, along with two qualifiers - the length (in bytes) of the received block and the length (in bytes) of the buffer.

5.13 Line initialization failure

An initialization failure occurred over an Ethernet line. This message displays the name of the line to which the event applies.

5.14 Send failure on line

A data transmission attempt failed on an Ethernet line. This message displays the name of the line to which the event applies, along with two event qualifiers - the reason for the failure and the distance. Failure reasons can include excessive collisions, short or open circuits, too long a frame, a framing error, an unrecognized frame destination, a remote failure to defer, a block check error, or data overrun.

5.15 Receive failed on line

Data was not received on an Ethernet line. This message displays the name of the line to which the event applies, along with two event qualifiers - the reason for the failure (as described in event 5.14) and the Ethernet header, which includes the source and destination node addresses and the protocol type.

5.16 Collision detect check failed on line

A check for collision detection failed on an Ethernet line. This message displays the name of the line to which the event applies.

5.17 DTE up*

A local DTE has synchronized X.25 levels 2 and 3. The DTE address and network name are displayed.

5.18 DTE down*

A local DTE is resynchronizing X.25 levels 2 and 3. The DTE address and network name are displayed.

5.19 Retransmit maximum exceeded*

A frame has been retransmitted the maximum number of times for this line without having received an acknowledgment. This means that either the PSDN or the line to the PSDN has failed.

5.20 FRMR received*

A frame reject was received. This is an error that you cannot correct by retransmitting the same frame again. This is usually due to a protocol violation.

5.21 Illegal frame received*

The frame received was not correctly formatted, or was not the frame expected by the X.25 protocol.

D.6. X.25 Packet Level Events

The following specific event classes and types are supported for X.25 packet level events. Only those events and entire event classes marked with an asterisk (*) are logged by DECnet-VAX components on nodes that have VAX PSI installed. marked with an asterisk (*) are logged by DECnet-VAX components on nodes that have VAX P.S.I. installed.

7.0 DTE state change*

The operator has altered the DTE state to that displayed.

7.1 Illegal packet received*

VAX P.S.I. has received a packet from the PSDN that is invalid in some way. This is usually because of a protocol violation by the PSDN.

7.2 Invalid LCN*

This is used whenever the logical channel number (LCN) in the received packet is invalid. For example: the network has sent a restart packet with a nonzero LCN, a call request packet on an LCN already in use, or a packet on an unused LCN, or tried to use an LCN reserved for a PVC. This is usually because of a protocol violation by the PSDN, but can be caused by not correctly configuring the PVCs in the database.

7.3 Flow control invalid*

Invalid P(S) received or invalid P(R) received. This is because of a protocol error.

7.4 Restart*

Either a restart packet was received or sent (in which case another event may also be logged). This indicates either a protocol error or that the DCE or DTE has reinitialized.

7.5 Clear*

A clear packet with a nonzero cause code was either received or sent because of a protocol error. See the PSDN documentation for the cause of the message and the diagnostic codes displayed.

7.6 Reset*

A reset packet with a nonzero cause code was either received or sent because of a protocol error. See the PSDN documentation for the cause of the message and the diagnostic codes displayed.

7.7 Diagnostic*

A diagnostic packet was either received or sent because of a protocol error. See the PSDN documentation for the cause of the message and the diagnostic codes displayed.

7.8 Reject*

A reject packet was received. VAX P.S.I. does not support receiving reject packets, because PSDNs are not permitted to send them.

7.9 Retransmission maximum exceeded*

The specified control packet has been transmitted the maximum number of times without being acknowledged by the PSDN. Usually this is caused by a failure by the PSDN or the connection to the PSDN, but may be because the DTE is not correctly configured.

7.10 Call failed*

The call (incoming or outgoing) has failed because of one of the following reasons: the application process has exited, the application has rejected the call, no destination has been set up, insufficient resources are available, a call has collided, security checks have failed, or a PSDN or a remote DTE has cleared the call.

7.11 State change*

The operator has initiated a state change, such as shutting down the node. This event is displayed when you shut down P.S.I. (or a part of P.S.I., such as the X.29 server).

D.7. VMS System-Specific Events

The following system-specific event classes and types are supported for VMS operating systems. Events and event classes marked with an asterisk (*) are logged by DECnet-VAX components. VMS operating systems. Events and event classes marked with an asterisk (*) are logged by DECnet-VAX components.

128.1 DAP CRC error detected*

A CRC error was detected at the DAP level by the file access listener (FAL) or by RMS. The event displays the name of the node to which the connection was established.

128.2 Duplicate PHASE 2 address error*

A circuit connection was attempted to a PHASE 2 node that was already reachable. You are allowed only one circuit connection to any PHASE 2 node.

128.3 Process created*

A process has been created to handle an incoming X.25 call. The message displays the process name, process ID, and the status returned by the $CREPRC system service.

128.4 Process terminated*

A process created to handle an incoming X.25 call has terminated. The message displays the process ID and the exit status of the process.

D.8. Distributed Name Service (DNS) Clerk Events

The following specific event classes and types are supported for the Distributed Name Service (DNS) clerk.

353.5 DNS clerk unable to communicate with server

The DNS clerk was unable to communicate with a DNS server. This message displays the name of the clearinghouse where the communication failed, the node on which the DNS server servicing the clearinghouse exists, and the reason why the communication failed, which may be any of the following:

- Unknown clearinghouse

  The requested clearinghouse is not serviced by the DNS server that was contacted. This may happen when the cache maintained by the local DNS clerk contains outdated information for a directory.

- Clearinghouse down

  A DNS server is unable to service a request because the clearinghouse is not operational (stopped state).

- Wrong state

  A DNS server is unable to service a request because the clearinghouse is currently starting up or shutting down.

- Data corruption

  A DNS server is unable to service the request because the clearinghouse file has been corrupted.

- No communication

  A network error occurred on the local system or on the system containing the DNS server. The local VMS error is displayed as a part of this message.

353.20 Local DNS Advertiser error

This event communicates errors that are local to the DNS Advertiser. All of these errors have the prefix ADV. The DNS Advertiser (DNS$ADVER) has encountered an error that prevents proper operation

of the process. Exact errors are listed in the system error messages chapter in the *OpenVMS System Messages and Recovery Procedures Reference Manual*.