PATHWORKS for DOS

digital

DECnet Network Management Guide

PATHWORKS for DOS

DECnet Network Management Guide

Order Number: AA–PAFHC–TK

September 1991

This guide describes the DECnet Network Control Program (NCP). It explains how you can use NCP commands to set up your personal computer as a node on a DECnet network. It also explains how to use NCP commands to obtain detailed information about the network and your node.

Revision/Update Information:	This is a revised guide.
Operating System/Version:	See the PATHWORKS for DOS Software Product Description for this information.
Software Version:	PATHWORKS for DOS V4.1

Digital Equipment Corporation
Maynard, Massachusetts

September 1991

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

The following are trademarks of Digital Equipment Corporation:

ALL–IN–1, DEC, DECmate, DECnet, DECnet–DOS, DECnet–ULTRIX, DECrouter, DECsystem, DECstation, DECwrite, Ethernet, LAT, PATHWORKS, PCSA, RSX–11M, RSX–11M–PLUS, RSTS/E, Scholar, ThinWire, TOPS–10, TOPS–20, ULTRIX, VAX, VAXcluster, VMS, VT, WPS–PLUS, and the Digital logo.

The following are third-party trademarks:

3Com and EtherLink are registered trademarks of 3Com Corporation. COMPAQ is a registered trademark of COMPAQ Computer Corporation. Hayes is a registered trademark of Hayes Microcomputer Products, Inc. Hercules is a registered trademark of Hercules Computer Technology. IBM, PC/XT, and Personal Computer AT are registered trademarks of International Business Machines Corporation. Microsoft, MS, MS–DOS, and MS-NET are registered trademarks of Microsoft Corporation. Olivetti is a registered trademark of Ing. C. Olivetti. Tandy is a registered trademark of Tandy Corporation. Toshiba is a registered trademark of Kabushiki Kaisha Toshiba. UNIX is a registered trademark of UNIX System Laboratories. Zenith is a trademark of Zenith Data Systems Corporation.

This manual was produced by Telecommunications and Networks Publications.

Contents

3 Operating a DECnet Executor Node

4 Monitoring a DECnet Network

5 Testing the Network

6 Controlling Your Node

A Network Control Program Error Messages

B Maintenance Operations Protocol (MOP) Load & Dump Programs

Examples

Figures

Tables

Preface

The *DECnet Network Management Guide* provides the information necessary for you to manage a DECnet–DOS™ system within a DECnet™ network. This guide defines and explains the terms and concepts related to DECnet network management, and describes the network management command set for PATHWORKS for DOS Version 4.1.

The term "DECnet–DOS" refers to DECnet–DOS Version 4.1 running on supported industry-compatible personal computers using the MS–DOS® operating system.

For a list of all supported industry-compatible personal computers and supported operating systems, see the PATHWORKS for DOS V4.1, Software Product Description (SPD).

Purpose

This guide describes the Network Control Program (NCP). An overview section describes DECnet network management concepts and defines network management terms. The guide explains how you can use NCP commands to set up your personal computer as a node in a DECnet network. It also explains how to use NCP to obtain detailed information about your node and the network. In addition, it includes information about loopback testing, event logging, and displaying system counters and database information.

Intended Audience

This guide is intended for users who need to set up, monitor, update, and troubleshoot their DECnet node in a DECnet network. You should be familiar with the MS–DOS operating system, DECnet network concepts, and DECnet testing procedures (such as loopback).

Structure of This Guide

This guide consists of the following sections:

- Chapter 1 introduces DECnet–DOS network management and the Network Control Program (NCP) and explains how to use the NCP commands. This chapter discusses your personal computer's relationship to the DECnet network, components and characteristics of a DECnet network, and functions of the DECnet–DOS network management utility.
- Chapter 2 describes the NCP commands needed to set up the DECnet–DOS network software on your personal computer.
- Chapter 3 describes the NCP commands needed to maintain and update the network software on your personal computer.
- Chapter 4 describes the NCP commands needed to monitor the network software on your personal computer. Monitoring the software on other nodes is discussed in the description of the Network Management Listener (NML).
- Chapter 5 describes the NCP commands that you can use to test your personal computer's network hardware and software. The chapter discusses the use of the Loopback Mirror in loop testing.
- Chapter 6 describes the DECnet–DOS network management command set and its use.
- Appendix A lists the error messages you can receive while using NCP.
- Appendix B describes the programs and commands which facilitate the loading and dumping of devices such as terminal servers and routers.
- The Glossary contains a list of DECnet–DOS network management terms.

If you are familiar with networking concepts and DECnet products, you can omit reading the overview in Chapter 1. If you are familiar with network management and loop testing, you can go directly to Chapter 6 for the command descriptions.

Conventions

The following conventions are used in this guide:

Convention	Meaning
`Special Type`	Indicates examples of system output or user input. System output is in black; user input is in teal.
COMMAND	Represents acceptable abbreviations for commands. For example, **DE**LETE indicates that the acceptable abbreviation for the DELETE command is DEL.
UPPERCASE	In commands and examples indicates that you should enter the exact characters shown. However, you can enter them in either uppercase or lowercase.
italics	In commands and examples indicates a value that either the system supplies or you should supply.
{ }	Indicate that you are required to specify one (and only one) of the enclosed options. Do not type the braces when you enter the command.
[ ]	Indicate that the enclosed data is optional. (If a vertical list of options is enclosed, you can specify only one option.) Do not type the brackets when you enter the command.
( )	Enclose a set of options that must be specified together.
vertical list of options	A vertical list of options not enclosed within braces, brackets, or parentheses indicates that you may specify any number of options (or in some cases none if defaults apply).
key	Indicates that you should press the specified key. Ctrl/*x* indicates that you should hold down the CONTROL key while you press the *x* key, where *x* is a letter.
Return	Indicates that you should press the key that executes commands. This key may be labeled Enter, Return, or ←, depending on your keyboard.

Associated Documents

The following is a list of PATHWORKS for DOS documents that provide information for users:

- *Installing and Configuring (with Diskettes)*
- *User's Handbook*
- *DECnet User's Guide*
- *DECnet SETHOST Terminal Emulation Guide*
- *DECnet Programmer's Reference Manual*
- *Memory Solutions*
- *Software Product Description*

In addition, several other resources can be helpful to some users:

- A reference manual for the appropriate operating system.
- The README.TXT file on the first floppy diskette of your installation kit.

The postage-paid Reader's Comment form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

1

Overview of Network Management

This guide describes the PATHWORKS for DOS Network Management utility. To use the network management tools effectively, you should be familiar with the concepts and terminology of networking and network management. This chapter will familiarize you with network management by discussing the following topics:

- An overview of network management concepts
- Using the Network Control Program (NCP) to manage your DECnet node
- General NCP command formats
- Getting help
- Using the Network Management Listener
- An overview of data storage

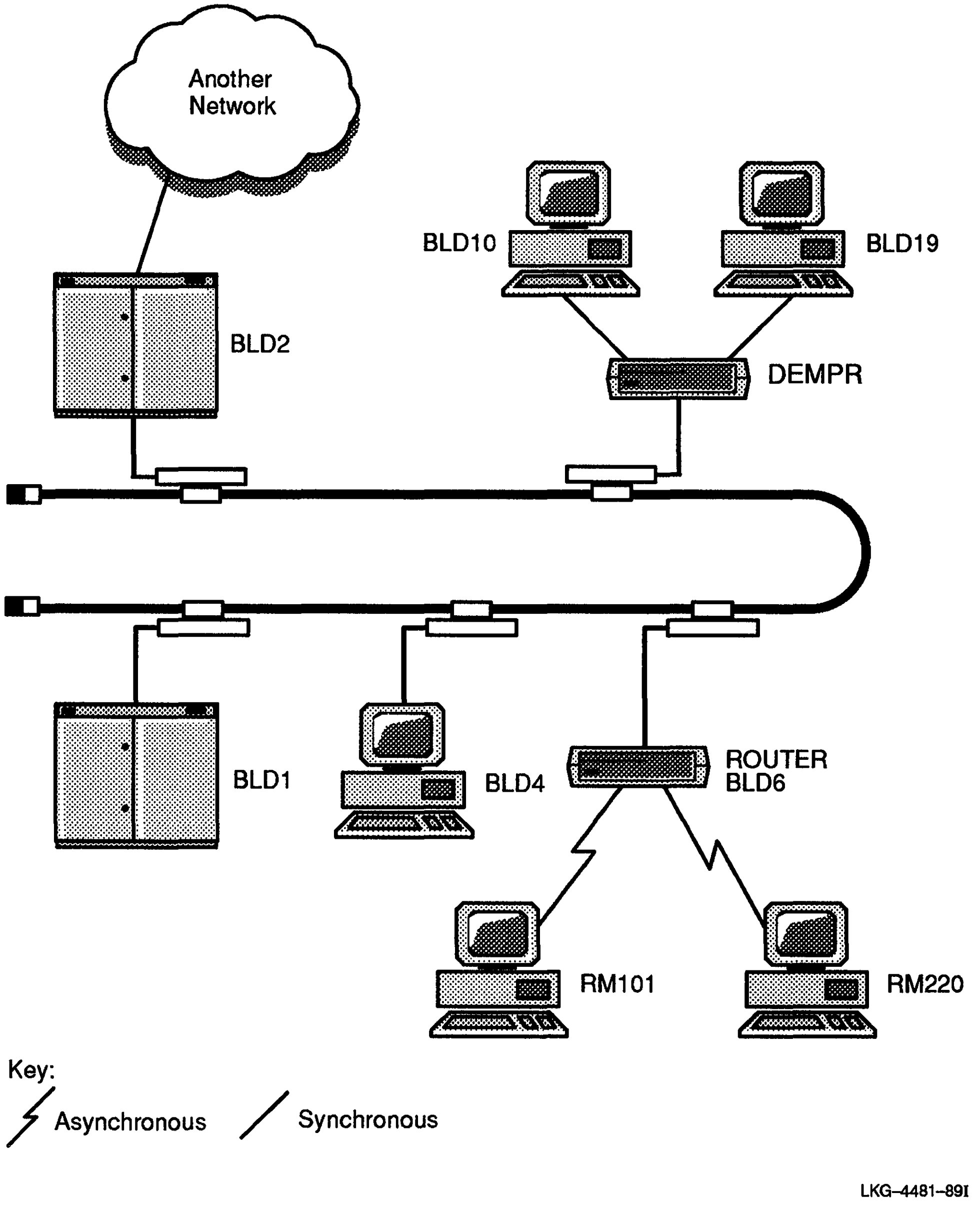
You should know who is responsible for the overall management of your network. If you have questions regarding your node or the network, this person can provide you with more information or assistance.

1.1 About DECnet Networks

A personal computer has a limited amount of space to store information. You can use removable diskettes and backup devices to increase the storage capacity of your personal computer. However, the information stored on these media is not quickly accessible, and you are still limited to information contained on your system.

A network is two or more computers connected together via communications lines (see Figure 1-1). When it is part of a network, your personal computer can access additional information including files from remote locations.

Figure 1–1 A Sample Network



DECnet software creates an extension to your MS–DOS operating system enabling it to interact in a network environment. PATHWORKS for DOS commands allow you to exchange information over the network communication lines. With PATHWORKS for DOS you can:

- Send and read mail.
- Transfer files between your system and another on the network.
- Print a file on a printer connected to another system on the network.
- Use a disk attached to another computer on the network as if it were a disk on your own personal computer.
- Connect to a remote system with your node as a terminal.

1.1.1 Terminology

As Figure 1–1 shows, personal computers are not the only components connected to the network. The DECnet software allows connections to any Digital Equipment Corporation system, many personal computers, and with the appropriate hardware and software configurations, connections to SNA networks and Packet Switched Data Networks (PSDNs). Each of these component systems on a DECnet network is called a **node**.

Throughout this guide the term *node* refers to any processor, intelligent terminal or other computer system capable of functioning independently within a DECnet network environment. Figure 1–1 depicts a typical network to illustrate terms and further define how nodes are used in the context of a DECnet network.

- An **adjacent node** is physically connected to your node by a single communications line, as BLD1 and BLD4.
- An **end node** such as BLD10, BLD19, BLD1, BLD4, RM101, or RM220 can receive information for its own use; it cannot receive messages and subsequently route them to another node.
- A **routing node**, such as the shaded nodes BLD2 and BLD6 can receive and forward information from one node in the DECnet network to another node or network.
- An **executor node** can perform network management functions.
- A **local node** is the node at which you are physically typing commands. For instance, if you are typing commands at the node RM101, RM101 would be your local node.
- A **remote node** is any node in the network other than your local node.

1.2 Managing a DECnet Node

Network Control Program (NCP) commands from your personal computer perform the following set of functions:

- Node Generation
  - Identify your personal computer to the rest of the network
  - Specify node names and default outgoing access information
  - Specify incoming access control information
- Executor Node Operation
  - Change network services parameters
  - Turn your executor node on and off in a network
  - Set the state of a line, circuit, or link
- DECnet Network Monitoring
  - Read or monitor log information
  - Display local network counters
  - Display local network parameter settings
- Node Diagnostics and Testing
  - Display information about your node and how it is interacting with the network
  - Send test messages over each element of a connection

1.3 Running the Network Control Program Utility

You can run the NCP utility using one of the following three methods:

- Single command method:

  Enter NCP followed by the specific NCP command. Press Return. After you enter each NCP command, the system returns the DOS prompt. For example:

  ```
  C:\>NCP command Return
  C:\>
  ```

- Single command method using a command or redirected input file:

  Use this method for entering commands from a command file. A command file (also called a redirected input file) contains all the NCP commands to complete a series of NCP operations. By using a command file, you need issue only one command to process all other commands in the file.

  To enter commands using a command file, type NCP followed by a left angle bracket (<) and the name of the file. Then press Return. For example:

  ```
  C:\>NCP <file-name Return
  ```

  NCP returns an error level value on completion. This error level can be used by the IF ERRORLEVEL statement in command files. If NCP has run successfully, it returns a value of 0. If the execution was not successful, it returns a non-zero value.

  By using the SET ECHO command, you can specify whether NCP displays a redirected input file on your screen as the file is executed. See Chapter 6 for specific information on how to turn echo on and off.

- Multiple command method:

  Supply the name of the utility (NCP) and press Return. The NCP utility responds with its own prompt. (You cannot use this method for submitting NCP commands in redirected mode.)

  ```
  C:\>NCP Return
  NCP>
  ```

  Enter the NCP command after the NCP prompt and press Return. Continue this procedure until you have entered all your commands.

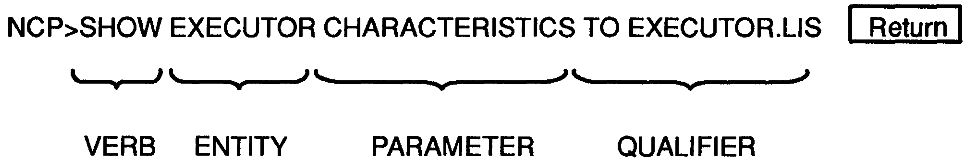
  ```
  NCP>command Return
  NCP>command Return
  NCP>command Return
  NCP>
  ```

1.3.1 Command Format

Most NCP commands consist of three parts: the command verb, an entity on which the command operates, and one or more parameters that further qualify the action to be taken. For each command, you must supply the verb, one entity and (usually) one or more parameters or qualifiers.

Figure 1–2 shows the parts of a NCP command:

Figure 1–2 A Network Control Program Command



```
NCP>SHOW EXECUTOR CHARACTERISTICS TO EXECUTOR.LIS Return
```

You can abbreviate commands and keywords to three or more unique characters. (An exception to this is the use of the keywords LINK and LINE. The first three characters for each keyword are the same; so you must enter all four characters to properly identify the keyword.) For example, you can type **SHO** for **SHOW**. However, for clarity and consistency, all examples in this manual show the full command format.

Table 1–1 displays the NCP command verbs.

Table 1–1 Network Control Program Command Verbs

To Do the Following...	Use This Command Verb...
Remove or clear temporary parameters.	CLEAR
Copy the node database to another file (without the access information).	COPY TO
Copy node information from another node into your database.	COPY FROM
Establish specific parameters that take effect when you perform a network restart.	DEFINE

(continued on next page)

Table 1–1 (Cont.) Network Control Program Command Verbs

To Do the Following...	Use This Command Verb...
Exit from NCP.	EXIT
Provide on-line information about the NCP utility and all of its commands.	HELP
Display information set up using the DEFINE command. You can also redirect this information to an output file.	DEFINE
List information contained in the permanent databases.	LIST
Run loopback tests that let you check the following: • The operation of your local node • The connection to the network • The connection to a remote node on the network • The communication hardware (including modems) that connects your system to the network	LOOP
Set up a software loopback mirror to echo messages from a remote node back to that remote node. Before a remote node can run a LOOP test to your node, you must first start the mirror on your local node.	MIRROR
Continuously display event logging information as it occurs.	MONITOR
Delete information from the DECnet databases.	PURGE
Display event logging information. You can also redirect log information to a specified output file.	READ
Establish specific parameters for the network. These parameters (and their values) are not saved across network restarts.	SET
Display information about the currently running system. This information includes counter displays, parameters established with the SET command, and parameters that cannot be set. You can redirect information displayed by the SHOW command to a specific output file.	SHOW

(continued on next page)

Table 1–1 (Cont.) Network Control Program Command Verbs

To Do the Following...	Use This Command Verb...
Instruct a remote node to display information about its executor, lines, or circuits.	TELL
Reset the counters associated with a specific network entity.	ZERO

1.3.2 Getting HELP

If you need help to select NCP commands, type HELP at the NCP> prompt; then press Return. For example:

```
NCP>HELP Return
```

The system responds with the following display:

```
The HELP command displays information about NCP commands on your screen.
HELP is available for the following commands:

;       CLEAR   COPY   DEFINE   EXIT   LIST   LOOP   MIRROR
MONITOR PURGE   READ   SET      SHOW   TELL   ZERO
```

You can type the name of a command for which you want information. For example:

```
NCP>HELP CLEAR Return
```

The system responds with the following display:

```
HELP is available for the following commands:

CLEAR EXECUTOR               CLEAR KNOWN
CLEAR LOCAL-ADAPTER-NAME     CLEAR REMOTE-ADAPTER-NAME
```

You can get additional help on a particular NCP command. For example, if you want more information on the CLEAR EXECUTOR command, type the following command:

```
NCP>HELP CLEAR EXECUTOR Return
```

The system responds with the following display:

```
Removes specified executor node parameters from running system
(DDCMP only).

RECEIVE PASSWORD   Password the executor node must receive from the
                   adjacent node in order to exchange messages.

TRANSMIT PASSWORD  Password the executor node must transmit to the
                   adjacent node in order to exchange messages.
```

1.3.3 Exiting from the Network Control Program Utility

To exit from NCP, you can use one of two methods. You can use the NCP EXIT command, or you can enter Ctrl/Z and then press Return. The current running command completes, then NCP exits. For example:

```
NCP>EXIT Return
```

or

```
NCP> Ctrl/Z Return
```

You can also use Ctrl/C to leave NCP. If you use Ctrl/C, however, the current running command is interrupted and may not complete.

1.4 Network Management Servers

The NCP utility uses various servers to handle requests from remote systems. These servers enable other nodes to request information from your local node. Likewise, the same servers, running on remote nodes, allow you to gather status from these remote nodes using NCP.

1.4.1 Using the Loopback Mirror

You can use the MIRROR command, detailed in Chapter 6, to echo loopback messages to the remote nodes that issue them. The loopback mirror runs in the foreground of DECnet–DOS nodes, so no other activity can take place while this server is running.

When you issue loopback messages to a remote node, the MIRROR returns them. If the remote node MIRROR does not return a message is, a problem must exist between your local node and the remote node to which you sent the loopback message.

1.4.2 Using the Network Management Listener

The Network Management Listener (NML) is a background process that responds to network management queries from remote DECnet systems. These systems must be running NCP or other network management tools that use the NICE V4.0 protocol.

If you want to zero the circuit counters on another personal computer on the DECnet network, you can issue the following command:

```
NCP>TELL BLD10 ZERO CIRCUIT COUNTERS Return
```

This NCP command tells the node BLD10 to zero all of its circuit counters. If NML is running on BLD10, it zeroes all of its circuit counters. If NML is not running on BLD10, NCP displays an error message on the requesting node.

The NML utility runs in the background so that it does not interfere with normal use of your personal computer. As a background task, it is limited to examining the currently running system (the volatile databases, whose information is displayed by using the SHOW commands). NML cannot access any information residing in the file system (the permanent database whose information is displayed by using the LIST commands).

1.5 Restarting a DECnet–DOS Node

PATHWORKS for DOS V4.1 allows you to stop and restart the network software. The effect of stopping and restarting network software is much the same as the effect of rebooting your DECnet node. Your node refers to permanent databases for all default networking information, and any network states are reset to the values in the permanent databases. To stop the network process, simply unload the network software (see PATHWORKS for DOS, *Memory Solutions*). To restart the network, simply reload the network software.

The term **network restart** encompasses both rebooting a running DECnet node and restarting stopped network software.

1.6 DECnet Network Management Databases for DOS

All DECnet nodes use a similar system of data organization. That is, all DECnet nodes use a system of **permanent** and **volatile** databases to configure component characteristics.

Permanent databases store on disk the characteristics you want your system to use when DECnet is started. Permanent databases are retained across network restarts.

Volatile databases are stored only in the running DECnet process in your computer's memory. They contain the values that you have used since the time DECnet was started. They are not stored in the permanent databases because you might not want to use them again. You can choose whether to store characteristics permanently. Volatile databases serve as a temporary memory for your computer.

2 Configuring a DECnet–DOS Node

This chapter presents an overview of the NCP commands and parameters that you can use to set up your node as a DECnet executor node. On an executor node you can run NCP commands to perform network management functions in a DECnet network.

2.1 Setting Up Your Node

You must choose a name and address to identify your node to other nodes in the network. Each node must have a unique name and address. In most cases, you obtain your node name and address from a system or network manager in your facility. The person coordinating network activities must keep a list of all node names and addresses to ensure there is no duplication.

Your local node address and name are established when you use the PATHWORKS for DOS installation procedure. You need not repeat this step after installation is completed. The PATHWORKS for DOS, *Installing and Configuring (with Diskettes)* provides a step-by-step explanation of the installation procedure.

However, if you need to change your node name and node address, you can change them with the DEFINE EXECUTOR command. The DEFINE EXECUTOR command establishes executor characteristics that include node name and address. The permanent database, DECPARM.DAT, stores the executor's node name and address. Therefore, the executor node becomes effective when you perform a network restart.

The following command enters the executor name as ROCKY in the permanent database, DECPARM.DAT:

```
NCP>DEFINE EXECUTOR NAME ROCKY ADDRESS 2.975 Return
```

This command does not take effect until the next network restart because it changes an entry in the permanent database. For the change to take effect immediately, type the following command:

```
NCP>SET EXECUTOR NAME ROCKY ADDRESS 2.975 Return
```

Note

This command is valid only if the executor state is off. If the executor state is on, use the DEFINE command and then unload and restart the network.

You should change database information in both the permanent and the volatile databases, using both the DEFINE and SET commands.

A **node name** is an identification string that consists of one to six alphanumeric characters and includes at least one alphabetic character. For example:

RM101

A **node address** is the unique numeric identification of a specific node. The node's address includes an area number and a node number:

area.node

Where:

area is a number in the range of 1 to 63.

node can be in the range of 1 to 1023. It is separated from the area number by a period. For example:

2.975

In this example, the area number is 2 and the node number is 975.

No two nodes can have duplicate node addresses in a single DECnet network.

2.1.1 Naming Remote Nodes

At your local node, you can create a list of the remote nodes that you want to access by name. This list can include access control information that allows default access to a specific account on a remote node. (This is the outgoing access control information that is stored in the DECALIAS.DAT database). To specify remote node information, use the DEFINE NODE command.

When you create a list of remote node names, the following information is required:

- **Node address**

  A numeric string that includes the area number in the range of 1 to 63 and the node number in the range of 1 to 1023. If you enter a node address without an area number, the default area number is that of your executor node.

- **Node name**

  A character string that consists of 1 to 6 alphanumeric characters with at least one alphabetic character.

For instance, to access node RM2, you must first enter the following command:

```
NCP>DEFINE NODE 2.65 NAME RM2 Return
```

Once RM2 is defined, you can access it without having to specify the node number. The number 2.65 is the node address for the node RM2.

2.1.2 Specifying Default Access Control Information

Access control information allows you to access a specified remote node with the privileges of a specified user. This information includes:

- **User name**

  A character string that consists of 1 to 39 alphanumeric characters that identifies the user at the remote node.

- **Password**

  A character string consisting of 1 to 39 alphanumeric characters that might be required to access files or programs on the remote node.

- **Account**

  A character string consisting of 1 to 39 alphanumeric characters. (This field is provided for systems that require a field containing information for system accounting. It is not required on all systems.)

To store the characters that represent the access information in uppercase letters, type the characters. For example:

Abc is stored in the database as **ABC**.

To store the characters exactly as you entered them, place the string in double quotation marks. Character strings that are not in quotation marks are forced to uppercase. For example:

"Abc" is stored in the database as **Abc**.

When you access a remote node by node name only, a DECnet–DOS utility follows this procedure to determine access control information:

1. The utility checks to see if access information has been provided directly to the utility. If this is true, then the utility uses that information to access the remote node.
2. The utility determines if default access information was specified with an NCP DEFINE NODE command by checking the permanent database file, DECALIAS.DAT.
3. If no default access control information was specified using NCP, the utility assumes that the remote node does not require access control information, and the requested operation is attempted using no access information.

2.1.3 Specifying Default Remote Host Type

You can also specify that the node you want to access is a SMB server for MS-NET. Use the following designations to specify the appropriate node:

MS-NET Designates this node as an MS-NET file server.

For example:

```
NCP>DEFINE NODE 2.375 NAME BLD3 USER SMITH PASSWORD JANE MS-NET Return
```

This command assigns node name BLD3 to node address 2.375. It specifies the user's name as SMITH, the password as JANE, and the node as a MS-NET file server.

2.1.4 Displaying Default Access Control Information

The LIST KNOWN NODES command displays a list of remote nodes and any default access control information that has been set for those nodes. If the access control information contains a password, that password is not displayed on the screen. Instead, it is represented by three dots (...).

To display information from the permanent database for the remote nodes known to your local node, enter the NCP command, LIST KNOWN NODES. This command displays the following information for nodes that are known to the local node:

- Node address
- Node name
- Number of current active logical links to this node
- Access control information

```
NCP>LIST KNOWN NODES Return
```

The system responds with a display similar to the following for a local node named ROCKY:

```
Known Permanent Nodes as of  4-May-1991 16:36:30
Executor node              = 2.975 (ROCKY)
State                        On
Executor Identification      PATHWORKS for DOS V4.1 client.

Node       Node     Active            Account
Address    Name     Links   MS-NET   Information

 2.375     BLD3     0                /SMITH/...
 4.30      BLD10    1       M
 4.60      BLD6     0
 4.62      BLD1     0
 4.64      RM101    0
55.86      FLR3     0                /DJONES
55.170     FLR5     0
NCP>
```

2.2 Understanding the DECnet Network Management Databases

The NCP utility uses information from six permanent databases. These databases act as storage areas and provide all of the information necessary for NCP to control your DECnet node. The installation and configuration procedure creates the appropriate databases for you during the installation process. You can then use NCP commands to change or delete specific information in each database file, or you can delete all of the information in a specific database file.

These permanent databases are retained across network restarts and they store the following information:

- Node name
- Node address
- Line characteristics
- Circuit characteristics
- Access information for connections to selected remote nodes
- Access information for connections from remote nodes
- Objects to be run by the Job Spawner (Refer to the PATHWORKS for DOS, *DECnet User's Guide* for more information about these objects.)

- NETBIOS remote adapter information (Refer to the PATHWORKS for DOS, *DECnet Programmer's Reference Manual* for more information about NETBIOS remote adapters.)

To enter, remove, or display information in the permanent database (which causes values to take effect at the next network restart), use these commands:

```
DEFINE
PURGE
LIST
```

The NCP utility also uses volatile databases which are stored in the running DECnet process in your computer's memory. They contain the values that you used since the time DECnet was started. These values take effect immediately but they are not retained across network restarts.

To enter, remove, or display information in the volatile database use these commands:

```
SET
CLEAR
SHOW
```

The following are the six permanent databases NCP uses:

- **DECACC.DAT**—This is the permanent incoming access database. You can set up information that allows other systems to access files or programs on your local node. The database stores the information on a user-name basis. Applications use the information in this database to check the access privileges of users who are trying to get information from your node. This allows you to control who can or cannot gain access to your node.
- **DECALIAS.DAT**—This is the permanent outgoing access database. You can store information here to be used when you want to access other nodes in the network. Outgoing access information includes a user name, account information, and a password. This information is assigned for each node.
- **DECNODE.DAT**—This is the permanent DECnet node database. It contains a list of node names and corresponding node addresses, a pointer to the outgoing access information, and an MS-NET indicator.
- **DECOBJ.DAT**—This is the permanent object database that the Job Spawner uses. It contains the definitions of the objects that the Job Spawner can access.
- **DECPARM.DAT**—This is the permanent DECnet database. It contains all of the configuration information about your local node, including executor, line, and circuit information.

- **DECREM.DAT**—This is the permanent remote adapter name database. MS-NET uses this database. It contains information that includes both a remote node name and an object associated with the adapter name.

PATHWORKS for DOS creates the appropriate databases during the installation procedure. The DECREM.DAT database is not automatically created during installation.

Note

When you start DECnet, the network process checks for a DECPARM.DAT file. If the file does not exist, the network process initializes using its own set of default parameters.

When you start NCP, the utility also checks for a DECPARM.DAT file. If the file does not exist, NCP uses the default parameters set up by the running network process to create the file. Use NCP commands to change or modify these parameters.

If you do not set the parameters, DECnet uses default values where appropriate. When you boot your system, these values are loaded into the volatile database.

Although you can use both the DEFINE and SET commands to modify network parameters, each command has a different result when you issue it:

- The DEFINE and PURGE commands establish and delete specific parameters, but the changes do not take effect until the next time you perform a network restart. The NCP utility stores the parameters in the permanent databases: DECPARM.DAT, DECNODE.DAT, DECALIAS.DAT, DECACC.DAT, DECREM.DAT, and DECOBJ.DAT. Values that you have changed with the DEFINE and PURGE commands are carried across network restarts.
- The SET and CLEAR commands also establish and remove specific parameters, but the changes take effect as soon as the command is issued. The volatile database stores these temporary parameters. The parameters established by the SET commands do not retain their values across a network restart unless you also modify them in the permanent database using the DEFINE or PURGE commands.

For detailed information about the database manipulation commands, see Chapter 6.

3 Operating a DECnet Executor Node

This chapter provides an overview of the NCP commands and parameters you can use to maintain and operate your node in a DECnet network. Node maintenance includes the following functions:

- Displaying network information
- Displaying executor node information
- Changing network information for your node
- Naming remote nodes
- Specifying and changing default access control information

3.1 Displaying Network Information for Your Node

NCP commands allow you to display the following local node and network information:

- Node, line, and circuit parameters.
- Node names, node addresses, and access control information for remote nodes (if they have been defined locally).
- Counters on the flow of network messages to and from your node. The counters also record error conditions.
- Network status (state) and other related information for a specific node, line, link, or circuit.

In DECnet, a circuit is the communications path established between the local node and its adjacent node. A line is the physical line connecting the local node to the adjacent node.

You can use NCP commands to change the executor node, line, and circuit parameters according to the needs of your network. Do not change the settings for these parameters unless you are familiar with DECnet network concepts and are aware of how all nodes (and their parameters and characteristics) interact in the network.

3.1.1 Displaying Default Access Control Information

The NCP SHOW KNOWN NODES command displays a list of remote nodes in the volatile database. Along with the node names, SHOW KNOWN NODES displays the default access information for nodes that have had connections to or from the local node since the last time the network was restarted. Enter the following command:

```
NCP>SHOW KNOWN NODES Return
```

The system responds with a display similar to the following:

```
Known Volatile Nodes as of  4-Aug-1990 16:36:30

Node      Node     Active          Account
Address   Name     Links   MS-NET  Information

 2.375    BLD10    0               /SMITH/...
 4.64     RM5J     0
55.86     RM101    0               /DJONES
55.335    BLD2     0
NCP>
```

To display permanent database information for the remote nodes known to your node, type the NCP command LIST KNOWN NODES.

```
NCP>LIST KNOWN NODES Return
```

This command displays the following information for nodes that are known to the local node:

- Node address
- Node name
- Number of current active logical links to this node
- Whether the node is an MS-NET server
- Access control information

The system responds with a display similar to the following when the local node is ROCKY:

```
Known Permanent Nodes as of  17-May-1991 16:36:30

Executor node              = 2.975 (ROCKY)

State                        On
Executor Identification      PATHWORKS for DOS V4.1 client.

Node      Node   Active        Account
Address   Name   Links   MS-NET  Information

 2.375    BLD3   0               /SMITH/...
 4.30     RM2    1       M
 4.60     FLR19  0
 4.62     RM1    0
 4.64     BLD1   0
 4.215    RM10   0
 4.216    RM1A   0
 4.298    FLR1   0
55.86     FLR3   0               /DJONES
55.124    BLDJ   1       M
55.170    FLR5   0
55.242    BLDK   0       M
55.261    RM110  0
55.335    RM5J   0
NCP>
```

3.1.2 Displaying Executor Node Information

To display information about the currently running executor node (your local node), enter the following command:

```
NCP> SHOW EXECUTOR CHARACTERISTICS Return
```

This command summarizes the network parameters for the currently running executor node as follows:

```
Executor Characteristics as of 24-May-1991 15:21:13

Executor Node                   = 2.375

Driver Version Number           = 4.0.0
State                           = ON
Executor Identification         = PATHWORKS for DOS V4.1 client.
Executor Version                = 4.2.0
Node Name                       = ROCKY
Node Address                    = 2.975
Incoming Timer                  = 45
Outgoing Timer                  = 60
Confidence Timer                = 15
Incoming Proxy                  = DISABLED
Outgoing Proxy                  = ENABLED
Active Links                    = 1
NSP Version                     = 4.0.0
Maximum links                   = 16
Delay Factor                    = 32
Delay Weight                    = 2
Inactivity Timer                = 30
Retransmit Factor               = 12
Routing Version                 = 2.0.0
Routing Type                    = Endnode IV
Maximum Buffers                 = 16
Buffer Size                     = 1498
Segment Buffer Size             = 557
Receive Password                = (password not set)
Transmit Password               = (password not set)
Database Path                   = C:\DECNET\
Transmit pipe Quota             = 6
Receive pipe Quota              = 6
PC Type                         = IBM PCAT
Nak Quota                       = 0
Autoboot                        = ENABLED
NCP>
```

To display the executor node values that take effect when you perform a network restart, use the LIST EXECUTOR CHARACTERISTICS command, which would produce a display similar to the following:

```
Executor Characteristics as of 24-May-1991 15:21:13

State                          = ON
Executor Identification        = PATHWORKS for DOS V4.1 client.
Node Name                      = ROCKY
Node Address                   = 2.975
Incoming Timer                 = 45
Outgoing Timer                 = 60
Confidence Timer               = 15
Incoming Proxy                 = DISABLED
Outgoing Proxy                 = ENABLED
Maximum Links                  = 4
Delay Factor                   = 32
Delay Weight                   = 2
Inactivity Timer               = 30
Retransmit Buffer              = 12
Maximum Buffers                = 16
Buffer Size                    = 1498
Segment Buffer Size            = 557
Receive Password               = (password not set)
Transmit Password              = (password not set)
Transmit pipe Quota            = 6
Receive pipe Quota             = 6
Nak Quota                      = 0
Autoboot                       = ENABLED
NCP>
```

3.2 Changing Network Information for Your Node

The following sections describe the network parameters that you can change.

Note

Some combinations of settings for buffer sizes, buffer counts, and timers can cause performance problems or network failures. Change network parameters only to solve specific problems. If new problems occur as a result, change the parameters back to the default settings.

3.2.1 Line State

The state of the communications line at network restart can be ON or OFF. If a line's state is OFF, it is not available for DECnet activity. Chapter 4 describes network line states and substates in detail.

3.2.2 Buffer Size

Every node in the network uses areas in memory, called buffers, for temporary storage of data. The buffers hold information being transferred to and from each node.

The buffer size parameter defines the maximum size of buffers for communications over LAN links.

You can change this parameter using only the DEFINE command. Therefore, you must perform a network restart for the change to take effect.

The maximum buffer size used by the system will be the lesser of the value entered by the user and the maximum buffer size that the network can support (determined by the data link driver.)

3.2.3 Segment Buffer Size

The segment buffers are used for transmitting DECnet network messages. The segment buffer size defines the maximum packet size for communications over WAN links. Do not change the segment buffer size unless you are very familiar with DECnet network concepts and know how the local network is set up.

You can change this parameter by using only the DEFINE command. Therefore, you must perform a network restart for the change to take effect.

3.2.4 Maximum Links

This parameter defines the maximum number of links available for the node. When you adjust this parameter, the size of buffers is adjusted accordingly. As a result, the higher the number for maximum links, the more buffers PATHWORKS for DOS reserves.

You must change this parameter using the DEFINE command. Therefore, you must perform a network restart for the change to take effect.

3.2.5 Executor State

When the executor state is ON, remote nodes can establish a logical link to the executor node. If you set the executor state OFF, the NCP utility deinstalls the DECnet Network Process (DNP.EXE) and stops the DECnet software.

3.2.6 Node Name and Node Address

To change your node name and node address, use the DEFINE EXECUTOR command. This command establishes executor characteristics that include node name and address. The executor node name and address become effective when you perform a network restart.

```
NCP>DEFINE EXECUTOR ADDRESS 2.975 NAME ROCKY Return
```

For this value to take effect immediately, use the SET command as follows:

```
NCP>SET EXECUTOR ADDRESS 2.975 NAME ROCKY Return
```

Note

The executor state must be OFF before you SET the executor address.

3.3 Naming Remote Nodes

At your local node, you can create a list of the remote nodes to access by name. This list can include access control information that allows default access to a specific account on a remote node. (These node names are stored in the node database, DECNODE.DAT.) To specify remote node information, use the DEFINE NODE command.

When you create a list of remote node names, the following information is required:

- Node address
- Node name

To define the remote node BLD3, enter the following command:

```
NCP>DEFINE NODE 2.375 NAME BLD3 Return
```

To determine what nodes are already defined, enter the LIST KNOWN NODES command.

3.3.1 Specifying Default Access Control Information for a Remote Node

Access control information allows you to access a specified remote node with the privileges of a specified user. (The DECALIAS.DAT database stores this outgoing access control information.) This information includes:

- User name
- Password
- Account

To store uppercase letters representing the access information, type the characters. Character strings that are not in quotation marks are forced to uppercase. For example:

Abc is stored in the database as **ABC**.

To store the characters as typed, type the string in double quotation marks.

For example:

"**Abc**" is stored in the database as **Abc**.

4

Monitoring a DECnet Network

This chapter provides an overview of Network Control Program (NCP) commands and parameters used to monitor your local node or any node in a DECnet network. The chapter is divided into two sections: one on monitoring your own node, and the other on monitoring a remote node.

Monitoring consists of the following tasks:

- Observing the current state of local and remote nodes and physical lines.
- Displaying the contents of the permanent and volatile databases for the local and remote systems.
- Inspecting and interpreting the contents of various counters on local and remote nodes.

4.1 Monitoring Your Node in a DECnet Network

DECnet maintains counters of network status at all times. To examine the performance of your node on the DECnet network, refer to these counters for the appropriate information. You can also issue a monitor command that lets you keep track of network events that occur on your node. This section explains all the information available to you and how to access it.

Observing the following counters can help diagnose certain problems and improve network performance.

4.1.1 General Counters

This section provides the information you need to understand each general counter.

Seconds since last zeroed

The length of time the counters have been accumulating.

4.1.2 Line Counters

This section provides the information you need to understand each line counter.

> **Note**
>
> Line counters may be invalid for Data Link Layer Network Driver Interface Specification (DLLNDIS) drivers as some vendors do not implement them in their drivers.

Collision detect check failures

Indicates that the heartbeat signal is not being properly supplied to the Ethernet device.

Receive failures

Indicate reception problems. It is typical to have some errors over time. However, many errors over a short period can indicate network hardware configuration problems.

Receive overruns

Indicate packets lost because the local adapter could not keep up with the network traffic. This is typical for slow systems or single-buffered Ethernet adapters. One way to improve the situation is to disable multicast reception with the SET CIRCUIT MULTICAST LISTENER DISABLED command.

System buffer unavailable

A packet arrived and no buffers were available in the data link pool to receive it.

User buffer unavailable

A control message could not be sent because no buffers were available in the data link pool.

4.1.3 Circuit Counters

This section provides the information you need to understand each circuit counter.

User buffer unavailable

A control message could not be sent because no buffers were available in the data link pool.

4.1.4 Executor Counters

This section provides the information you need to understand each executor counter.

CCB allocation failures

No internal control blocks were available when requested. The number of CCBs is controlled by the number of links and buffers.

LDB allocation failures

No Large Data Blocks (LDBs) were available when requested. This can occur for Ethernet configurations if the data link buffer pool is at its limit. The number of LDBs is controlled by the EXECUTOR BUFFERS parameter.

This can be because of a delay in acknowledging messages across the network. However, if it appears to affect performance, increase the number of buffers.

Link allocation failures

No links or sockets were available when requested. The number of links is controlled by the EXECUTOR MAXIMUM LINKS parameter.

Because PATHWORKS for DOS links are not closed when a program aborts (unless the program does so itself), you may abort any open network links with the NCP SET KNOWN LINKS STATE OFF command.

Receive connect resource errors

An incoming connect failed because no links were available. The EXECUTOR MAXIMUM LINKS parameter controls the number of links.

Response timeouts

Indicates the expiration of a message timeout.

SDB allocation failures

No more Small Data Blocks (SDBs) were available when requested. This occurs if many asynchronous operations are outstanding or if the network is having problems transmitting control messages. The maximum number of links controls the number of SDBs.

4.1.5 Displaying Counter Information

To view the contents of these counters, use the SHOW command. For example, to see the counters associated with your executor node, enter the SHOW EXECUTOR COUNTERS command.

Example 4–1 shows a sample Ethernet counter screen display, produced by the SHOW EXECUTOR COUNTERS command.

Example 4–1 SHOW EXECUTOR COUNTERS Sample Display

```
Executor Counters as of 1-May-1991 15:57:30

Executor node              = 2.975 (ROCKY)

Seconds since last zeroed           = 65
User bytes received                 = 210847
User bytes sent                     = 2662
User messages received              = 176
User messages sent                  = 53
Total bytes received                = 212777
Total bytes sent                    = 3776
Total messages received             = 225
Total messages sent                 = 129
Connects received                   = 0
Connects sent                       = 1
Response timeouts                   = 0
Receive connect resource error      = 0
Maximum logical links active        = 1
Packet format error                 = 0
Verification error                  = 0
CCB allocation failure              = 0
SDB allocation failure              = 0
LDB allocation failure              = 0
Link allocation failure             = 0

NCP>
```

Enter the SHOW LINE [ETHER-1 | ASYNC-1 | TOKEN-1] COUNTERS command to display information about line counters. Example 4–2 shows a sample screen display resulting from a SHOW LINE COUNTERS command.

Example 4–2 SHOW LINE COUNTERS Sample Display

```
Line Counters as of 1-May-1991 16:03:48

Line = ETHER-1

Seconds since last zeroed        = 763
Bytes received                   = 1346481
Bytes sent                       = 1980
Data blocks received             = 12477
Data blocks sent                 = 33
Multicast bytes received         = 899219
Multicast blocks received        = 9135
Blocks sent, initially deferred  = 0
Blocks sent, single collision    = 0
Blocks sent, multiple collisions = 0
Send failure                     = 0
Receive failure                  = 0
Unrecognized frame destination   = 0
Data overrun                     = 0
System buffer unavailable        = 0
User buffer unavailable          = 0
Collision detect check failure   = 0

NCP>
```

Enter the SHOW CIRCUIT [ETHER-1 | ASYNC-1 | TOKEN-1] COUNTERS command to display information about circuit counters. Example 4–3 shows a sample screen display resulting from a SHOW CIRCUIT COUNTERS command.

Example 4–3 SHOW CIRCUIT COUNTERS Sample Display

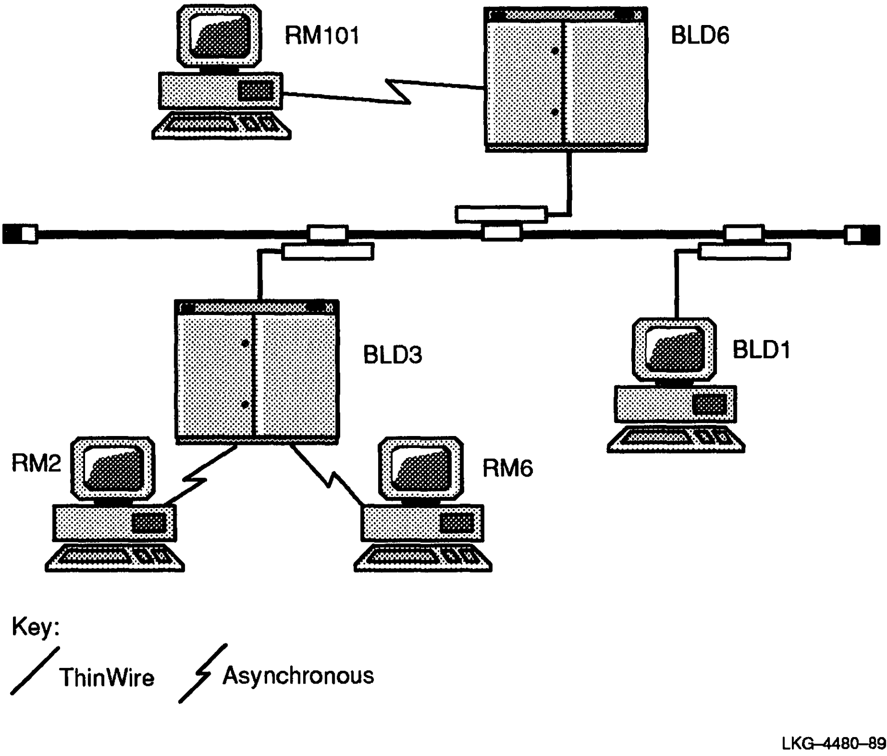
```
Circuit Counters as of 1-May-1991 16:26:44

Circuit = ETHER-1

Seconds since last zeroed          = 2097
Terminating packets received       = 0
Originating packets sent           = 0
Circuit down                       = 0
Initialization failure             = 0
Bytes received                     = 35053
Bytes sent                         = 2310
Data blocks received               = 252
Data blocks sent                   = 70
User buffers unavailable           = 0

NCP>
```

4.1.6 Displaying Default Access Control Information

The NCP SHOW KNOWN NODES command displays a list of recently accessed remote nodes and any associated default access control information. If the access control information contains a password, that password is not displayed on the screen. The password is represented by three dots (...).

Figure 4–1 shows remote nodes in a sample DECnet network. Each node has a specific node name.

Figure 4–1 Known Remote Node Names



To display information for the remote nodes that appear in Figure 4–1, type the NCP command SHOW KNOWN NODES. This command displays the following information for nodes that are known to the local node ROCKY:

- Node address
- Node name
- Number of current active logical links to this node
- Delay
- Next node

Example 4–4 shows a sample display resulting from the SHOW KNOWN NODES command.

Example 4–4 SHOW KNOWN NODES Sample Display

```
Known Volatile Nodes as of  4-May-1991 16:36:30

Node      Node    Active  Delay    Next Node
Address   Name    Links            Address   Name

 2.975    ROCKY   0           5
 4.64     BLD1    0
55.86     FLR3    0
55.335    RM5J    0           1
NCP>
```

4.1.7 Displaying Executor Node Information

To display information about the currently running executor node, enter the following command:

```
NCP>SHOW EXECUTOR CHARACTERISTICS Return
```

The resulting display summarizes the network parameters for the currently running executor node.

```
Executor Characteristics as of 24-May-1991 15:21:13

Executor Node                   = 2.975

Driver Version Number           = 4.0.0
State                           = ON
Executor Identification         = PATHWORKS for DOS V4.1 client.
Executor Version                = 4.2.0
Node Name                       = ROCKY
Node Address                    = 2.975
Incoming Timer                  = 45
Outgoing Timer                  = 60
Confidence Timer                = 15
Incoming Proxy                  = DISABLED
Outgoing Proxy                  = ENABLED
Active Links                    = 1
NSP Version                     = 4.0.0
Maximum  links                  = 16
Delay Factor                    = 32
Delay Weight                    = 2
Inactivity Timer                = 30
Retransmit Factor               = 12
Routing Version                 = 2.0.0
Routing Type                    = Endnode IV
Maximum Buffers                 = 24
Buffer Size                     = 1498
Segment Buffer Size             = 557
Receive Password                = (password not set)
Transmit Password               = (password not set)
Database Path                   = C:\DECNET\
Transmit pipe Quota             = 6
Receive pipe Quota              = 6
PC Type                         = IBM PCAT
Nak Quota                       = 0
Autoboot                        = ENABLED
NCP>
```

4.1.8 Displaying Remote Node Information

To determine if you defined a specific node with the DEFINE NODE command, enter LIST NODE followed by the node name. For example, if the node you want to check is WALKER, enter the following command:

```
NCP>LIST NODE WALKER Return
```

Example 4–5 displays a sample node name, node address, access control information, and the number of current active logical links for node WALKER from the local node.

Example 4–5 LIST NODE WALKER Sample Display

```
Known Permanent Nodes as of  4-May-1991 16:36:30

Executor node              = 2.975 (ROCKY)

State                        On
Executor Identification      PATHWORKS for DOS V4.1 client.

Node      Node    Active          Account
Address   Name    Links   MS-NET  Information

 2.379    WALKER  0               /SMITH/...
 4.64     BLD1    1
NCP>
```

If you want information only for nodes with active logical links, use the SHOW ACTIVE NODES command. Example 4–6 shows a sample display of a node name and address, access control information, and number of current active links from the local node to the adjacent node and any other nodes with active logical links to a sample node.

Example 4–6 SHOW ACTIVE NODES Sample Display

```
Known Volatile Nodes as of  4-May-1991 16:36:30

Node      Node    Active          Account
Address   Name    Links   MS-NET  Information

 4.64     BLD1    1               /SMITH/...
NCP>
```

4.1.9 Displaying Line Information

In the context of PATHWORKS for DOS, a **line** is the physical line that connects the local node with the adjacent node. There are two possible line states:

- **ON**—The line is ready for use by DECnet and unavailable to other software.
- **OFF**—The line's database and parameters are present and the line is available to other software, but it is not available for any DECnet network activity.

There are three substates. They can be observed only when the line state is ON. The three substates are:

- **SYNCHRONIZING**—The line is initializing.
- **RUNNING**—The line is in normal running state.
- **SERVICE**—The line is reserved for loopback testing.

To display status information about the physical line's characteristics, use the SHOW LINE CHARACTERISTICS command. Example 4–7 displays sample line characteristics.

Example 4–7 SHOW LINE CHARACTERISTICS Sample Display

```
Line Characteristics as of 24-May-1991 15:36:13

Line = ETHER-1

Line State                     = ON
Line Substate                  = Running
Device ID                      = 3CM-1
Receive Buffers                = 8
Protocol Type                  = Ethernet
Hardware Address               = 08-00-2B-09-9C-89
Station Address                = AA-00-04-00-E8-13
NCP>
```

4.1.10 Displaying Circuit Information

In the context of PATHWORKS for DOS, a **circuit** is the communications path established between the local node and a remote node with which you are communicating. There are two possible circuit states:

- **ON**—The circuit is ready for use.
- **OFF**—The circuit's database and parameters are present, but the circuit is not available for any DECnet network activity.

There are three circuit substates that further describe the circuit when it is ON. The substates are:

- **SYNCHRONIZING**—The circuit is initializing.
- **STARTING**—The circuit is initialized.
- **RUNNING**—The circuit is active.

To display status information about the circuit's characteristics, use the SHOW CIRCUIT CHARACTERISTICS command. Example 4–8 displays sample circuit characteristics information.

Example 4–8 SHOW CIRCUIT CHARACTERISTICS Sample Display

```
Circuit Characteristics as of 24-May-1991 15:39:44

Circuit = Ether-1

Circuit State                   = ON
Circuit Substate                = Running
Service                         = Disabled
Designated Router               = 4.378()
Block Size                      = 1498
Hello Timer                     = 30
Listen Timer                    = 90
Verification                    = 0
User                            = DECnet
Owner                           = DECnet
Line ID                         = Ether-1
Protocol type                   = Ethernet
Multicast Listener State        = Disabled
NCP>
```

4.1.11 Monitoring Events on Your Local Node

The NCP utility provides several commands that allow you to monitor the activity of nodes on your network. For example, the MONITOR LOGGING command displays network events as they are being logged. An **event** is an occurrence that has potential significance in the operation and/or maintenance of a network. **Logging** is the process that generates a record of the event. Capturing this information in a file can be useful to you if network problems occur. The READ LOG command displays the contents of the event-logging buffer.

For PATHWORKS for DOS systems, event-logging is always turned on, and all events are displayed. (You cannot select only a few events to be displayed.) After the events are displayed or redirected to a file, they are discarded.

If an event is lost, NCP displays an error message before it displays the rest of the logged events. For example:

```
Event type 0.0  Event records lost
```

For each event, NCP displays the following information:

Event type *nn.nn name-of-event-type*
Occurred *dd-mmm-yyyy hh:mm:ss*
other-event-dependent-text

See Chapter 6 for more information on the MONITOR LOGGING command.

4.2 Monitoring Other Nodes on the DECnet Network

The Network Management Listener (NML) utility responds to network management queries from remote DECnet systems. These systems must be running NCP or other network management tools that use the NICE V4.0 protocol.

The NML utility enables other nodes on the network to query your node for network information. Similarly, you can query other nodes on your DECnet network for network information, if the nodes are running NML.

The NML utility runs in the background so that it does not interfere with normal use of your personal computer. As a background task, it is limited to examining the currently running system (the volatile databases, whose information is displayed by using the SHOW commands). NML cannot access any information residing in the file system (the permanent database whose information is displayed by using the LIST commands).

5 Testing the Network

This chapter provides an overview of the NCP commands that you can use to diagnose problems connecting your personal computer to the network. These commands allow you to display information about your node and the way it interacts with the network. You can also display information about other nodes in the network.

You do not need special technical knowledge to use these commands, but it is helpful to know the **topology** of your network. The topology indicates which nodes are in the network, where they are located, and how they are connected. A basic understanding of network testing concepts (such as loopback) is also helpful.

This chapter is divided into two sections:

- **Overview of the NCP test commands**—Describes the NCP test commands and lists the functions you can perform with them.
- **Testing the network**—Describes the sequence of operations performed by the loopback commands as they test the network components (such as hardware, loopback connectors, or cables).

5.1 Overview of the NCP Test Commands

The NCP utility provides a way for you to interact with the network. This interaction includes setting network parameters and displaying information about your node and the network. In addition, NCP provides a way to troubleshoot your network by issuing loopback tests from your node and allowing other nodes to send loopback tests to your node.

Because problems can occur in any of the network components, NCP commands let you test the operation of each component. The tests check the operation of your local node, the connection to the remote node, and the communication hardware between them.

By starting with your local node and progressing outward to each of the components in your network, you can determine which specific component is not operating properly. If all components appear to be working properly, you should contact your network manager or the manager of the node you are trying to reach. Have the manager perform further tests to determine the network problem.

Table 5-1 lists and explains the commands you can use to display network information and send loopback tests for your node.

Table 5–1 NCP Commands for Displaying Network Information and Sending Loopback Tests

If you want to ...	Use this NCP command ...
Check the operation of your local node, the connection to the remote node, and the communication hardware that connects them.	**LOOP**
Continuously display event-logging information on the terminal screen.	**MONITOR LOGGING**
Display the contents of the event-logging buffer.	**READ LOG**
Display statistics about node, line, and circuit characteristics and counters.	**SHOW**
Instruct a remote node to display information about its lines or circuits.	**TELL**
Reset line, circuit, or executor counters.	**ZERO**

5.2 Testing the Network

You can use loop messages to test the following parts of your network:

- The local node
- The communications controller
- The Ethernet interface to the communications controller
- A remote node at the circuit level (for Ethernet and Token Ring)
- A remote node at the application level

The loop messages are sent through the communications controller. PATHWORKS for DOS V4.1 supports Ethernet connections using the integrated Ethernet communications controller, or asynchronous network connections using the integrated asynchronous serial port.

Some of the tests use a device called a loopback connector to check the operation of the various components. The **loopback connector** is a hardware device that you attach to each component to isolate it from the others while you test it. You use the loopback connector with the LOOP LINE command.

Loopback test messages are sent to the connector or network and echoed back to the local node. If the messages do not return or if they do not match the original messages that your node sent, the problem is likely with the component you are currently testing.

Loop tests require that the line and circuit states are ON before you run them. When you start the test, NCP displays a message indicating that the line or circuit is turned on. NCP turns lines and circuits on before running the loop test.

The **loopback mirror** is the network application used when a loopback test is performed from a remote node to your local node or when a loop test is performed on your local node software. The loopback mirror allows test messages from a remote node to be echoed back to that remote node. For loopback testing to your local node, enter the loopback mirror as an object in the DECnet object database by typing the following command:

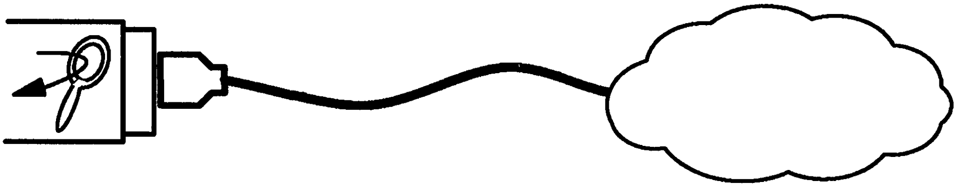
```
[C:\]NCP DEFINE OBJECT MIRROR NUMBER 25 FILE MIRROR.EXE Return
```

After you enter the loopback mirror in the object database, the Spawner automatically runs it whenever there is a loop request from a remote node or whenever you issue a LOOP EXECUTOR request from your local node. The loopback mirror does not run until a request is received. When the Spawner receives a request for the loopback mirror, the Spawner runs the loopback mirror in the background.

Use the MIRROR command on the local node when you perform a loopback test from a remote node to the local node. The MIRROR command allows test messages from a remote node to be echoed back to that remote node.

Figure 5–1 through Figure 5–4 provide conceptual views of the loopback tests on your local node. For specific examples of loopback tests and the LOOP command, see Chapter 6.

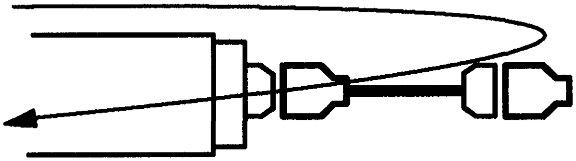
Figure 5–1 The Loop Executor Command



LKG–3211–89I

Figure 5–1 illustrates the function of the LOOP EXECUTOR command. This command tests your node software. It sends a loop message through the DNP and routing software to the loopback mirror (defined as an object in DECOBJ.DAT), which returns the message through DNP and the routing software. A successful LOOP EXECUTOR test verifies that your network software is working properly. If this test does not complete successfully, reinstall the network software.

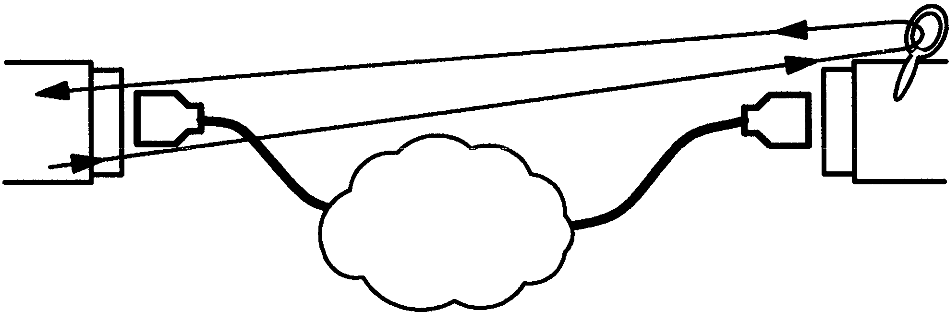
Figure 5–2 Loop Line Controller



LKG-3212-89I

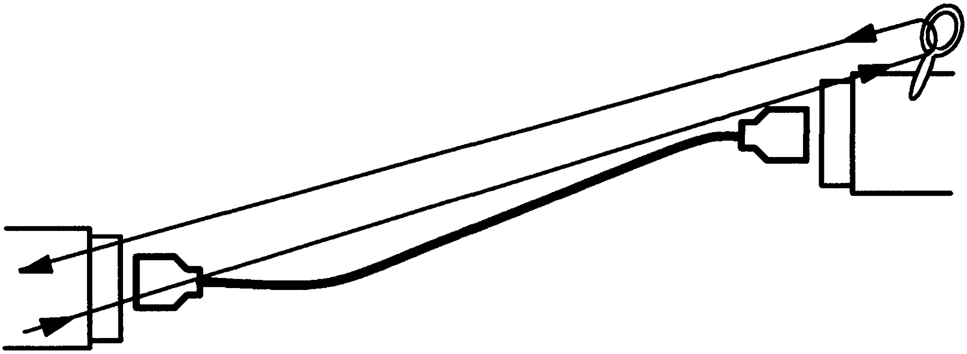
Figure 5–2 shows how you can test the network controller board on your personal computer by using the LOOP LINE CONTROLLER command. You must attach the loopback connector, included in your PATHWORKS for DOS software kit, to run this test. The message is sent out through the network controller and is turned around at the loopback connector. If this test does not complete successfully, check the hardware controller on your personal computer.

Figure 5–3 Loop Node



You can check the connection to a node on your network by issuing the LOOP NODE command. With this command, you can send test messages through the network software, the network controller, and over the line to another node. A loopback mirror should be running on the node to echo the test message back to your node, as shown in Figure 5–3. If this test does not complete successfully, it indicates a problem with the connection to this remote node.

Figure 5–4 Loop Circuit



LKG–3213–89I

You can check the logical link connection to an adjacent node by using the LOOP CIRCUIT command. With this command, you can send test messages to a remote node through the network and routing software, the network controller, and over the circuit to an adjacent node. The loopback mirror on the adjacent node echoes the message back to your local node using the same logical link, as shown in Figure 5–4. If this test does not complete successfully, it indicates a problem along the link to the adjacent node.

5.2.1 Loopback Counters

When you run a loopback test, network counters record events that occur while the test is running. The counters record the error and traffic information on the network, including events for your local node, line, or circuit. You can zero these counters prior to running the loop test by using the ZERO CIRCUIT, ZERO LINE, and ZERO EXECUTOR commands. The test records the errors and traffic that reflect the most recent network activity.

To display the counters after you run the test, use one of the following commands:

- SHOW CIRCUIT [ETHER-1 | ASYNC-1 | TOKEN-1] COUNTERS
- SHOW LINE [ETHER-1 | ASYNC-1 | TOKEN-1] COUNTERS
- SHOW EXECUTOR COUNTERS

Refer to Example 4–1, Example 4–2, and Example 4–3 for sample screen displays resulting from these commands.

Note

Loop commands that call the data link directly from NCP are not supported in Microsoft Windows Enhanced Mode.

NCP performs the loop tests using the following information:

- The length of the test message
- The number of times to send the test message
- The type of format for the test message
- The node that will receive the test message and return it

Refer to Chapter 6 for more information on LOOP commands.

6
Controlling Your Node

This chapter provides detailed descriptions of all the NCP commands that you can use for setting up, maintaining, updating, and monitoring your node in a DECnet network.

6.1 Network Control Program Commands

The NCP commands are presented in alphabetical order in the following sections. Each section explains how to use the command and provides command examples and sample system responses to the command.

Table 6–1 summarizes the NCP commands and their definitions.

Table 6–1 Network Control Program Command Summary

To Perform This Function...	Use This Command...
Remove executor node parameters stored for the currently running system.	CLEAR EXECUTOR
Clear a local adapter name from the volatile database.	CLEAR LOCAL-ADAPTER-NAME
Clear a remote adapter name from the volatile database DECREM.DAT.	CLEAR REMOTE-ADAPTER-NAME
Copy the node database to a specified file. Access information is not copied. (Refer to the COPY NODE command description for more information about this command.)	COPY KNOWN NODES
Copy information about a single node name from a specified remote node into the node database.	COPY NODE

(continued on next page)

Table 6–1 (Cont.) Network Control Program Command Summary

To Perform This Function...	Use This Command...
Define incoming access information for the designated user in the incoming access database (DECACC.DAT). Access information includes a user ID, a password, and the type of access.	DEFINE ACCESS
Define the circuit characteristics that take effect when you restart DECnet.	DEFINE CIRCUIT
Define executor node characteristics that take effect when you restart DECnet.	DEFINE EXECUTOR
Define line characteristics that take effect when you restart DECnet.	DEFINE LINE
Assign a node name and optional access control information to a unique node address.	DEFINE NODE
Define object information for the Job Spawner database (DECOBJ.DAT).	DEFINE OBJECT
Define a remote adapter name for NETBIOS to use. The adapter name includes a node name and an object associated with that node.	DEFINE REMOTE-ADAPTER-NAME
Exit from NCP.	EXIT
Display a summary of NCP commands on your screen.	HELP
Display entries in the incoming access database (DECACC.DAT).	LIST ACCESS
Display permanent information about the circuit from the DECnet database (DECPARM.DAT).	LIST CIRCUIT
Display permanent information from the DECnet (DECPARM.DAT)database for the executor node.	LIST EXECUTOR
Display permanent information about all occurrences of an entity from the DECnet database (DECPARM.DAT, DECNODE.DAT, DECOBJ.DAT, DECREM.DAT, and DECACC.DAT). (Refer to the LIST NODE command description for more information about this command.)	LIST KNOWN
Display permanent information about the line from the DECnet database (DECPARM.DAT).	LIST LINE

(continued on next page)

Table 6–1 (Cont.) Network Control Program Command Summary

To Perform This Function...	Use This Command...
Display permanent information about a remote node that is contained in the node database (DECNODE.DAT).	LIST NODE
Display information about an object in the Job Spawner database (DECOBJ.DAT).	LIST OBJECT
Display information about the remote adapter name in the remote adapter name database (DECREM.DAT).	LIST REMOTE-ADAPTER-NAME
Run loopback tests to a specific Ethernet address, a multicast Ethernet address, or an adjacent node on a DDCMP line.	LOOP CIRCUIT
Run a loopback test within the local node to a local loopback mirror.	LOOP EXECUTOR
Run a loopback test within the local node to the Ethernet controller. This test requires the use of a loopback connector.	LOOP LINE CONTROLLER
Run a loopback test to a remote node that has a loopback mirror.	LOOP NODE
Perform loop node tests from remote nodes.	MIRROR
Display event-logging information on the screen.	MONITOR LOGGING
Delete incoming access information for the specified user from the incoming access database (DECACC.DAT).	PURGE ACCESS
Remove specified executor node parameters from the DECnet database (DECPARM.DAT).	PURGE EXECUTOR
Delete information in the local node database for the specified node.	PURGE NODE
Delete information in the Job Spawner database (DECOBJ.DAT) using either the object name or object number.	PURGE OBJECT
Delete an entry from the remote adapter name database (DECREM.DAT).	PURGE REMOTE-ADAPTER-NAME
Display the contents of the event-logging buffer either on the screen or in a redirected file.	READ LOG

(continued on next page)

Table 6–1 (Cont.) Network Control Program Command Summary

To Perform This Function...	Use This Command...
Set or modify current circuit parameters.	SET CIRCUIT
Determine whether NCP displays the contents of a command file or a redirected file on your screen as the file is executed.	SET ECHO
Set or modify the executor node's current parameters.	SET EXECUTOR
Set or modify the line's current parameters.	SET LINE
Terminate the logical link connections associated with a specific socket.	SET LINK
Enter a local adapter name in the volatile database.	SET LOCAL-ADAPTER-NAME
Cause a long NCP display to pause at the end of a screen. This prevents the entire display from scrolling by. This command also issues a prompt so that you can view the rest of the display.	SET PAUSE
Enter a remote adapter name in the volatile database.	SET REMOTE-ADAPTER-NAME
Display information to your local node about current logical links.	SHOW ACTIVE
Display information about the node that is physically adjacent to your local node.	SHOW ADJACENT NODE
Display current circuit information.	SHOW CIRCUIT
Display current executor node information.	SHOW EXECUTOR
Display communication line information.	SHOW LINE
Display information about all logical links to the local node.	SHOW LINKS
Display information about local adapter names.	SHOW LOCAL-ADAPTER-NAME
Display information about nodes that have been accessed by PATHWORKS for DOS since the last time you stopped DECnet. To see permanent node information, use the LIST KNOWN NODES command.	SHOW NODE

(continued on next page)

Table 6–1 (Cont.) Network Control Program Command Summary

To Perform This Function...	Use This Command...
Display information about remote adapter names.	SHOW REMOTE-ADAPTER-NAME
Instruct a remote node to display information about its executor, lines, or circuits.	TELL
Zero the counters for the circuit.	ZERO CIRCUIT
Zero the counters for the executor node.	ZERO EXECUTOR
Zero the counters for the line.	ZERO LINE

CLEAR EXECUTOR

The CLEAR EXECUTOR command removes specified executor node parameters from the local volatile database for the running system. These parameters include a RECEIVE PASSWORD and a TRANSMIT PASSWORD. To remove these parameters from the permanent database, use the PURGE EXECUTOR command.

If RECEIVE and TRANSMIT passwords have been enabled on the executor node and the adjacent node, the passwords are automatically exchanged when either node turns on the communication line to the other node. The network coordinator defines these passwords for Digital Data Communications Message Protocol (DDCMP) use. The exchange of passwords also occurs when the communication line is restarted from either node because of errors.

Format

```
CLEAR EXECUTOR  { RECEIVE PASSWORD  }
                { TRANSMIT PASSWORD }
```

Where:

TRANSMIT PASSWORD
Removes the password the executor node must transmit to the adjacent node to exchange messages with the adjacent node. The password is a character string consisting of eight alphanumeric characters.

RECEIVE PASSWORD
Removes the password the executor node must receive from the adjacent node to exchange messages with the adjacent node. The password is a character string consisting of eight alphanumeric characters.

Note

These passwords are used for asynchronous DDCMP connections.

Example

The following command removes the password that the executor node must transmit to the adjacent node when the executor node first begins communications. Until the network coordinator defines a new transmit password, the executor node cannot open the communication line between itself and the adjacent node.

```
NCP>CLEAR EXECUTOR TRANSMIT PASSWORD [Return]
```

CLEAR LOCAL-ADAPTER-NAME

This command clears a specific local adapter name from the volatile database. The local adapter name identifies a network application written for the NETBIOS emulator interface. The network application resides on your personal computer.

The CLEAR KNOWN LOCAL-ADAPTER-NAMES command clears all the local NETBIOS names from the volatile database.

The CLEAR command for adapter name information is case sensitive. This means that you must specify the exact name to clear by enclosing the name in quotation marks. If you have two local adapter names that use the same letters but are in different cases, such as "BEST" and "best", you must be very specific when using the CLEAR LOCAL-ADAPTER-NAME command. If you do not use quotation marks with the CLEAR command, NCP removes the name stored in all capital letters.

Format

CLEAR LOCAL-ADAPTER-NAME *name* [**BYTE**16 *nn*]

or

CLEAR KNOWN LOCAL-ADAPTER-NAMES/HEX *hh*

or

CLEAR KNOWN LOCAL-ADAPTER-NAMES

Where:

name Specifies a 1- to 16-character ASCII printable string that specifies the name of the local application to remove from the volatile database.

nn Specifies a value for byte 16. The 16th character can be a non-printable ASCII character, entered in decimal. This value can be any decimal value from 0 to 255. The default is 20.

hh Specifies a number representing the adapter name in hexadecimal up to 16 pairs of hex digits. The first byte cannot be the hexadecimal value of 0 or 2a.

Note

Do not leave a space between the CLEAR LOCAL-ADAPTER-NAME command and the /HEX parameter.

Examples

1. In the following example, the entry, "Charlie", is removed from the volatile database:

```
NCP>CLEAR LOCAL-ADAPTER-NAME "Charlie" Return
```

2. The following commands remove local adapter name definitions from the volatile database. These two commands do the same thing:

```
NCP>CLEAR LOCAL-ADAPTER-NAME OTHERONE BYTE16 1
```

or

```
NCP>CLE LOC/HEX 4F-54-48-45-52-4F-4E-45-20-20-20-20-20-20-20-01
```

CLEAR REMOTE-ADAPTER-NAME

This command clears a specific remote adapter name from the volatile database. The remote adapter name identifies a network application written for the NETBIOS emulator interface. The network application resides on a remote personal computer.

The CLEAR KNOWN REMOTE-ADAPTER-NAMES command clears all the remote NETBIOS names from the volatile database.

The CLEAR command for adapter name information is case sensitive. This means that you must specify the exact name to clear by enclosing the name in quotation marks. If you have two local adapter names that use the same letters but are in different cases, such as "ANCHOR" and "anchor", you must be very specific when using the CLEAR REMOTE-ADAPTER-NAME command. If you do not use quotation marks with the CLEAR command, NCP removes the name stored in all capital letters.

Format

CLEAR **REM**OTE-ADAPTER-NAME *name* [**BY**TE 16 *nn*]

or

CLEAR **REM**OTE-ADAPTER-NAME/HEX *hh*

or

CLEAR **KNOWN** **REM**OTE-ADAPTER-NAMES

Where:

name	Specifies a 1- to 16-character alphabetic string that specifies the name of the remote application to remove from the volatile database.
nn	Specifies a value for byte 16. The 16th character can be a non-printable ASCII character, entered in decimal. This value can be any decimal value from 0 to 255. The default is 20.

hh Specifies a number representing the adapter name in hexadecimal up to 16 pairs of hex digits. The first byte cannot be the hexadecimal value of 0 or 2a.

Note

Do not leave a space between the CLEAR REMOTE-ADAPTER-NAME command and the /HEX parameter.

Examples

1. In the following example, the entry, "Hoosier", is removed from the volatile database:

```
NCP>CLEAR REMOTE-ADAPTER-NAME "Hoosier" Return
```

2. The following commands remove remote adapter definitions from the volatile database. These two commands do the same thing:

```
NCP>CLEAR REMOTE-ADAPTER-NAME SAINTSTEVEN BYTE16 11
```

or

```
NCP>CLE REM/HEX 53-41-49-4E-54-53-54-45-56-45-4E-20-20-20-20-0B
```

COPY NODE

The COPY NODE *node-id* FROM command copies information about a single node name (or a group of nodes within a single area) from a specified remote node into the node database (DECNODE.DAT). Use this command to add information from a particular node to your database. Use this command also to copy information from a node with a large node database if you want to copy information about nodes in a particular area.

The COPY KNOWN NODES FROM command specifies a remote node from which to copy the complete database of node names and node addresses to your local node. This set of node names does not include any access information. If the database on the remote node is large (for example, in excess of 100 nodes), this can significantly slow down the access time to the remote node's database. You can interrupt the copying of node names at any time by pressing any key on the keyboard.

The COPY KNOWN NODES TO command copies the local node database to a file. If the database contains access information, NCP does not copy that information.

Format

```
COPY NODE  { node-id }  FROM remote-node-id
           { area.*  }
```

or

```
COPY KNOWN NODES FROM   remote-node-id
```

Where:

node-id Specifies the name or the address of the node for which to collect information.

remote-node-id Specifies the name or the address of the remote node from which to collect information.

*area.** Specifies the particular node area for which you want to collect information. The asterisk (*) is a wildcard that indicates all the node information for the specified area. (The use of wildcards works if the remote system supports the wildcard option for node names for network management.)

Examples

1. The following command copies information about the node BASBAL from the node STADIA:

   ```
   NCP>COPY NODE BASBAL FROM STADIA [Return]
   ```

2. The following command copies the node database to the file NODEDATA.DAT:

   ```
   NCP>COPY KNOWN NODES TO NODEDATA.DAT [Return]
   ```

DEFINE ACCESS

The DEFINE ACCESS command defines incoming access information for the designated user. The access information is stored for each user, and it includes the password and access type. The access type describes the type of privilege the user has to access information on your node. If no access type is defined for the user, the access for that user is NONE.

The NCP utility stores the information you define using the DEFINE ACCESS command in the DECACC.DAT database. The DECACC.DAT file is used by the File Access Listener (FAL) and other DECnet utilities to limit access from remote systems.

To store the characters representing the access information in uppercase letters, type the characters. For example:

Abc is stored in the database as **ABC**.

To store the characters to as typed, place the string within quotation marks. For example:

"**Abc**" is stored in the database as **Abc**.

Format

```
                                                  [ NONE  ]
DEFINE ACCESS USER  username [PASSWORD password]  [ READ  ]
                                                  [ WRITE ]
                                                  [ ALL   ]
```

Where:

username	Represents a 1- to 39-character alphabetic string that defines the user.
password	Represents a 1- to 39-character alphabetic string that defines the user's password. The default is no password.

access-type Describes the type of privilege the user has for accessing information on this node. The four types of access are:

- NONE – no access for this user.
- READ – provides read-only access.
- WRITE – provides write-only access.
- ALL – provides both read and write access.

Example

The following command defines access information for the user BARKER. In this example, the password is JUMPER and the access type is READ. This means that the user BARKER can read information on your node.

```
    NCP>DEFINE ACCESS USER BARKER PASSWORD JUMPER READ Return
```

DEFINE CIRCUIT

The DEFINE CIRCUIT command defines permanent characteristics for the circuit. The value you specify does not take effect until the next time you boot your system or restart the network software. (For the value to take effect immediately but temporarily, use the SET CIRCUIT command.)

Format

```
                              ( HELLO TIMER seconds                        )
                              | MULTICAST-LISTENER { DISABLED }            |
                              |                    { ENABLED  }            |
                              |                                            |
                              |        { DECNET }                          |
                              | OWNER  { DLX    }                          |
DEFINE CIRCUIT circuit-id    {         { MOP    }                          }
                              |                                            |
                              | SERVICE { DISABLED }                       |
                              |         { ENABLED  }                       |
                              |                                            |
                              | STATE { ON  }                              |
                              (       { OFF }                              )
```

Where:

circuit-id — Specifies the circuit type (ETHER-1, ASYNC-1, or TOKEN-1.)

HELLO TIMER *seconds* — Specifies the frequency of routing hello messages sent to the adjacent node over the circuit. The valid range is 1 to 8191 seconds. The default is 30.

MULTICAST-LISTENER — Sets the circuit's multicast-listener state. The default is ENABLED.

- DISABLED – Specifies that the circuit will not respond to multicast requests. This applies to Ethernet communications.
- ENABLED – Specifies that the circuit responds to multicast requests. This applies to Ethernet communications.

OWNER — Specifies the circuit owner. The default is DECNET.

SERVICE

Sets the circuit's service state. The default is ENABLED.

- DISABLED – Specifies that the circuit is unavailable for service operations.
- ENABLED – Specifies that the circuit is available for special network activity such as loopback testing.

STATE

Sets the line's operational state to ON or OFF. The default is ON.

- ON – Allows network traffic to flow over the circuit. This is the normal operational state.
- OFF – Allows no traffic to flow over the circuit. The circuit is unavailable for any network activity.

Example

The following command sets the circuit state to ON for the next time you restart DECnet:

```
NCP>DEFINE CIRCUIT STATE ON Return
```

DEFINE EXECUTOR

The DEFINE EXECUTOR command defines executor characteristics that become effective after you restart DECnet. (For the value to take effect immediately but temporarily, use the SET EXECUTOR command.)

Format

```
                 ADDRESS node-address
                 CONFIDENCE TIMER seconds
                 DELAY FACTOR 1/16 of a second
                 DELAY WEIGHT number
                 IDENTIFICATION id-string
                 INACTIVITY TIMER seconds
                 INCOMING TIMER seconds
                 MAXIMUM BUFFERS number
                 MAXIMUM LINKS number
                 NAK QUOTA number
DEFINE EXECUTOR  NAME node-name
                 OUTGOING TIMER seconds
                 RECEIVE PASSWORD password
                 RECEIVE PIPE QUOTA number
                 RETRANSMIT FACTOR number
                 BUFFER SIZE number
                 SEGMENT BUFFER SIZE number
                 STATE [ON | OFF]
                 TRANSMIT PASSWORD password
                 TRANSMIT PIPE QUOTA number
                 AUTOBOOT
```

Where:

ADDRESS *node-address*
Specifies the address assigned to the executor node. The node address is a unique numeric identification for the node. It must include an area number and a node number.

CONFIDENCE TIMER *seconds*
Specifies an optional extra amount of time to wait before timing out a link. The value range for this timer is 1 to 32K. The default value is 15K.

DELAY FACTOR
1/16 of a second

Determines the amount of time to wait before a message is retransmitted to a node. The range is 16 to 4095. The default number is 32.

DELAY WEIGHT
number

Represents the weight to apply to a current round-trip delay to a remote node when updating the estimated round-trip delay to a node. The range is 1 to 255. The default weight is 1.

IDENTIFICATION
id-string

Specifies a text identification string for the executor node. The string can be 1 to 32 alphanumeric characters. You must use a set of double quotation marks (" ") to delimit any string containing blanks or tabs. You cannot use a single set of quotation marks within the identification string.

You can use the IDENTIFICATION parameter to identify the executor node to other nodes in the network (for example, by including your name or department in the string).

INACTIVITY TIMER
seconds

Represents the maximum duration of inactivity on a logical link before the node checks to see if the logical link still works. If no activity occurs within the minimum number of seconds, the node generates traffic to test the link. The value range is 1 to 32K. The default value is 30.

INCOMING TIMER
seconds

Represents the maximum number of seconds between the time a connection request is received at the executor node and the time that it takes to accept or reject the request. If the user program does not accept or reject the request within the specified number of seconds, the connection request is automatically rejected. If no value is set, there is no timer. The value range is 1 to 32K. The default value is 45.

MAXIMUM BUFFERS
number

Specifies the maximum number of network communication buffers available for use by DECnet. The valid range is 1 to 255. The actual number of buffers is restricted by available memory. This affects the memory usage. The default is 16.

MAXIMUM LINKS *number*

Specifies the maximum number of active logical links for the executor node. The valid range is 1 to 32. The default is 4.

NAK QUOTA *number*

Specifies the number of out-of-order segments to receive before sending a negative acknowledgment (NAK) out on the logical link. This number must be less than or equal to the RECEIVE PIPE QUOTA. The value range for this quota is 0 to 4095. The default is 0 (or 6 for slower devices and processors).

NAME *node-name*

Specifies the node name for the executor node. The node name is an identification string consisting of one to six alphanumeric characters that includes at least one alphabetic character.

OUTGOING TIMER *seconds*

Represents the duration between the time that the executor node requests a connection and the time that it takes the remote node to acknowledge the request. The value range is 1 to 65535. The default is 60.

RECEIVE PASSWORD *password*

Represents the password the local node must receive from the adjacent node to exchange messages with the adjacent node. The password is a character string that consists of eight alphanumeric characters.

RECEIVE PIPE QUOTA *number*

Represents the number of segments the network receives for a link by flow control. The valid range is 1 to 65535. The default is 6.

Note

Changing the default parameters can cause a performance difference. Refer to the PATHWORKS for DOS, *Installing and Configuring (with Diskettes)* before changing default parameters.

RETRANSMIT FACTOR *number*

Represents the maximum number of times that the executor node restarts the retransmission timer when it expires. If the executor node exceeds the number, it disconnects the logical link. The range is 1 to 65535. The default is 12.

BUFFER SIZE *number*

Allows you to define the maximum buffer size for wide area network messages. This buffer size is for messages within the local area network. The maximum value is 1498 bytes. This is also the default value.

The maximum buffer size used will be the lesser of the value entered by the user and the maximum buffer size that the network can support (determined by the data link driver.)

SEGMENT BUFFER SIZE *number*

Specifies the size of the segment buffers in bytes. These buffers are used for transmitting DECnet messages from the wide area network. The size is a decimal integer in the range of 1 to 16383 for DDCMP communications. The default is the user buffer size 576.

STATE

Defines the executor state to ON or OFF.

- ON – Allows logical links to be established to the executor node.
- OFF – Disallows logical links to be established to the executor node. (The executor's state is OFF if the DECnet network process fails to find the DECnet database file,DECPARM.DAT.) When the state is set to OFF, the system restarts with the executor state OFF.

TRANSMIT PASSWORD *password*

Represents the password the local node must transmit to the adjacent node to exchange messages with the adjacent node. The password is a character string that consists of eight alphanumeric characters.

TRANSMIT PIPE QUOTA *number*

Represents the number of segments the driver buffers into the network before it blocks the user on a transmission. The valid range is 1 to 65535. The default is 6.

AUTOBOOT	Reserved.

Examples

1. The following command defines the executor address as 2.975:

   ```
   NCP>DEFINE EXECUTOR ADDRESS 2.975 Return
   ```

2. The following command defines the executor address as 2.975 and assigns ROCKY as the executor name:

   ```
   NCP>DEFINE EXECUTOR ADDRESS 2.975 NAME ROCKY Return
   ```

3. The following command defines the maximum number of logical links for the executor node as 6:

   ```
   NCP>DEFINE EXECUTOR MAXIMUM LINKS 6 Return
   ```

DEFINE LINE

The DEFINE LINE command defines the state of the line at network restart. It becomes effective with the next network restart. (For the value to take effect immediately but temporarily, use the SET LINE command.)

Format

```
                        ⎧ DEVICE [device-id]             ⎫
                        ⎪ MODEM { NULL }                 ⎪
                        ⎪       { FULL }                 ⎪
DEFINE LINE [line-id]   ⎨ RECEIVE BUFFERS number         ⎬
                        ⎪ SPEED baud-rate                ⎪
                        ⎪ STATE { ON  }                  ⎪
                        ⎩       { OFF }                  ⎭
```

Where:

LINE [*line-id*]

Specifies either ASYNC-1, ETHER-1, or TOKEN-1 as the line for which parameters are to be created or modified. Because there is one line for each configuration, the *line-id* is optional.

DEVICE [*device-id*]

Specifies the device for controlling the asynchronous DDCMP line (COM-1, COM-2).

MODEM

Specifies the type of modem control to use. Modem control is used with asynchronous DDCMP connections.

- NULL – Specifies that modem control signals will not be checked.
- FULL – Specifies that modem control signals are checked. The line is shut down if the modem loses its connection. This is the default.

RECEIVE BUFFERS *number*

Specifies the number of buffers for receiving information. Increasing this number affects network response time and memory usage (for use in Ethernet communications). The valid range is 1 to 1024. The installation procedure sets the default based on the configuration of your node.

SPEED *baud-rate*

Sets the DDCMP line speed to one of the following:

50
75
110
134
150
200
300
600
1200
1800
2000
2400
3600
4800
7200
9600
19200

The default is 9600. For high performance systems, a speed of 19200 is available. Line speed 19200 does not work on 8086 machines.

STATE

Defines the line's restart state to ON or OFF.

- ON – Specifies that the line is available for network use.
- OFF – Specifies that the line's database and parameters are present, but the line is not enabled for any type of network activity. For asynchronous communications, the port is available for non-DECnet use.

If there is no DECPARM.DAT file, the default line state for restart is OFF.

Note

Information about modem, line speed, and service state applies to asynchronous DDCMP communications.

Examples

1. The following command defines the line state as ON. When you restart DECnet, the line comes up in the ON state.

```
NCP>DEFINE LINE STATE ON [Return]
```

2. The following command defines the maximum number of receive buffers for Ethernet communication as 6:

```
NCP>DEFINE LINE RECEIVE BUFFERS 6 [Return]
```

DEFINE NODE

The DEFINE NODE command assigns a node name plus access control information to a unique node address. To store the characters representing the access information in uppercase letters, type the characters. For example:

Abc is stored in the database as **ABC**.

To store the characters as typed, place the string within quotation marks. For example:

"Abc" is stored in the database as **Abc**.

Format

```
DEFINE NODE  node-address NAME node-name [ access-info ]
                                         [ MS-NET      ]
```

Where:

NODE *node-address*
Specifies the address of the node to be defined. The node address is a unique numeric identification for the node. It must include an area number and a node number.

NAME *node-name*
Specifies the name to be associated with the node address. The node name is an identification string of one to six alphanumeric characters, including at least one alphabetic character.

access-info
Specifies access control information for the remote node. This information uses the following format:

USER *user-name* [PASSWORD *password*] [ACCOUNT *account*]

Where:

user-name is a character string of 1 to 39 alphanumeric characters that identifies the user at the remote node.

password is a character string of 1 to 39 alphanumeric characters that might be required to access files or programs on the remote node.

account is a character string of 1 to 39 alphanumeric characters.

MS-NET — Indicates that the node is an MS-NET server. If you have PATHWORKS for DOS installed, this command causes an entry in the NETBIOS server volatile database. (Also refer to the PURGE NODE command.)

Examples

1. The following command assigns the node name RM2 to the node at address 2.65. It also specifies the user-name as MAGIC and the password as WAND.

   ```
   NCP>DEFINE NODE 2.65 NAME RM2 USER MAGIC PASSWORD WAND Return
   ```

2. To add access control information or a node designation to a node that you have already defined in the database, you do not have to type both the node name and node address to add the information. You can use either the node's name or its address. For instance, if you have already defined node BORIS with its address as 1.2, you can use either the name or the number in the following example.

   ```
   NCP>DEFINE NODE BORIS USER BENCH Return
   ```

   This command assigns access information for the user BENCH to the previously defined node BORIS using the node name.

DEFINE OBJECT

The DEFINE OBJECT command defines object information for the Job Spawner utility. The DECOBJ.DAT database stores the information. The Job Spawner utility uses the database information to start a program or application whenever it receives a connect request from other nodes in the network. For more information about using the Job Spawner utility, refer to the PATHWORKS for DOS, *DECnet User's Guide*.

To store the object information to be in uppercase letters, type the characters. For example:

Abc is stored in the database as **ABC**.

To store the characters as uppercase, place the string within quotation marks. For example:

"Abc" is stored in the database as **Abc**.

Format

DEFINE **OBJ**ECT *obj* [**NUM**BER *num*] **FILE** *file* [**ARG**UMENTS *xx*]

Where:

obj Represents the name of the DECnet object. The object name can have a maximum length of 16 characters.

num Represents the number of the DECnet object. The range for this number is 0 to 255.

file Represents the name of a command file or program for the Job Spawner to start. Command file names must end with the .CMD extension. (The file name can include a path specification.)

xx Represents the command-line arguments for the program specified with FILE. If there are spaces embedded within the string, be sure to enclose the string in quotation marks.

PATHWORKS for DOS reserves the numbers 1-127 for DECnet objects. Each number is predefined for a network program. The object numbers 128-255 are available for user-written or user-supplied applications. The following table lists the PATHWORKS for DOS predefined object numbers:

Object #	Object Name	Object File Name
0	Any object	Any object file name
17	FAL	FAL.EXE
63	DTR	DTR.EXE
128-255	Any user-written/ provided network object	Any user-defined name

The object number 0 has a special meaning. An unlimited number of objects can be defined as object 0. When DECnet receives a connection request of object number 0, it makes the connection based on name.

Note

The PATHWORKS for DOS installation procedure defines some objects for you during the installation process. The installation procedure lets you decide what objects to install. You can use NCP commands to change or delete these objects or to define new objects.

Examples

1. The following command defines the File Access Listener (FAL) as object number 17:

```
NCP>DEFINE OBJECT FAL NUMBER 17 FILE FAL Return
```

2. The following command defines a command file (MYCOM.CMD) as object number 0. When a connection for MYCOM.CMD is requested, DECnet matches the connection request object name with the object MYCOM.CMD in the DECOBJ.DAT database. The connection is not based on the object number 0.

```
NCP>DEFINE OBJECT MYCOM FILE NUMBER 0 MYCOM.CMD ARGUMENTS a b Return
```

3. The following command defines a command file (TST.CMD) as object number 128. The arguments for this file are stored in the database exactly as indicated in the quoted string.

```
NCP>DEFINE OBJECT TST NUMBER 128 FILE TST.CMD ARGUMENTS "A B" Return
```

DEFINE REMOTE-ADAPTER-NAME

The DEFINE REMOTE-ADAPTER-NAME command associates a remote adapter name with a remote node and object. The DECREM.DAT database stores the name. The adapter name includes a node name and a particular object associated with that node name.

A remote adapter is a network application written for the NETBIOS emulator interface. The application resides on a remote node.

For the value to take effect immediately but temporarily, use the SET REMOTE-ADAPTER-NAME command.

To store the user information to be in uppercase letters, type the characters. For example:

Abc is stored in the database as **ABC**.

To store the characters as uppercase, place the string within quotation marks. For example:

"**Abc**" is stored in the database as **Abc**.

Format

```
                                                               [ USER user-id   ]
DEFINE REMOTE-ADAPTER-NAME nam [BYTE16 n] NODE node OBJECT num [ PASSWORD pswd  ]
                                                               [ ACCOUNT acct   ]
```

or

```
                                                         [ USER user-id   ]
DEFINE REMOTE-ADAPTER-NAME/HEX hh NODE node OBJECT num   [ PASSWORD pswrd ]
                                                         [ ACCOUNT acct   ]
```

Where:

nam Represents a 1- to 16-character alphabetic string that specifies the name to assign to the remote application. To store the characters representing the name to be in uppercase letters, type the characters. If you do not want to store the characters as uppercase, place the string in double quotation marks.

node Identifies the node that is associated with this remote adapter. The node you use must be one that is already defined in the node database, DECNODE.DAT. You must identify a node for each remote adapter name that you define. The node-id can be either the node's name or its address. A node name is a 1- to 6-character alphanumeric string that identifies a specific node. A node address is a numeric character string consisting of an area number and a specific node number. The node address also serves as a unique identification of the node in the network.

num Represents the number of the DECnet object. The range for this number is 0 to 255. The default is 0.

n Specifies a value for byte 16. The 16th character can be a non-printable ASCII character, entered in decimal. This value can be any decimal value from 0 to 255. The default is 20.

user-id Represents the user information that accesses the loopback mirror on the remote node. A user ID consists of 1 to 39 alphabetic characters.

pswd Represents a unique character string that accesses the loopback mirror on the remote node. A user's password consists of 1 to 39 alphanumeric characters (some systems restrict password lengths to 8 characters).

acct Represents alphanumeric information needed to access the adjacent node's loopback mirror. The information consists of 1 to 39 alphanumeric characters.

hh Specifies a number representing the adapter name in hexadecimal up to 16 pairs of hex digits. The first byte cannot be the hexadecimal value of 0 or 2a.

Note

Do not leave a space between the DEFINE REMOTE-ADAPTER-NAME command and the /HEX parameter.

Example

1. The following command causes the DECREM.DAT database to store the remote adapter name WINSTON. Whenever a connection request with the number 42 is made, the Job Spawner accesses node WINSTON for the network application.

   ```
   NCP>DEFINE REMOTE-ADAPTER-NAME WINSTON NODE 2.67 OBJECT 42 [Return]
   ```

2. To enter a hexadecimal string, enter the following command to define the name abcde:

   ```
   NCP>DEFINE REMOTE-ADAPTER-NAME/HEX 41-42-43-44-45 NODE 2.90 [Return]
   ```

3. The following commands define a remote adapter name in the permanent database. These two commands do the same thing:

   ```
   NCP>DEFINE REMOTE-ADAPTER-NAME SAINTSTEVEN BYTE16 11 NODE LUVLIT
   ```

   or

   ```
   NCP>DEF REM/HEX 53-41-49-4E-54-53-54-45-56-45-4E-20-20-20-20-0B NODE LUVLIT
   ```

EXIT

The EXIT command lets you exit from NCP. You can also use Ctrl/Z Return to exit from NCP.

Format

```
EXIT
```

Example

The following command causes you to exit from NCP and returns you to the prompt or system you were previously using:

```
NCP>EXIT Return
```

HELP

The HELP command displays information about NCP commands on your screen.

Format

HELP [*command-verb* [*entity*]]

Where:

command-verb Specifies the NCP command for which you need information.

entity Specifies that you need information about the specific command as it relates to this component.

Example

The following command displays HELP information for the LIST NODE command:

```
NCP>HELP LIST NODE Return
```

The system responds with:

```
Display information about a remote node contained in the permanent
database.

    LIST NODE [node-id]            [TO file-name]
    LIST NODE [area.*]             [TO file-name]
    LIST NODE [*.*]                [TO file-name]
```

LIST ACCESS

The LIST ACCESS command displays an entry in the incoming access database (DECACC.DAT) for a specific user. The DECACC.DAT stores the access information, and it includes the password and access type. The access type describes the type of privilege the user has to access information on your node, such as READ or WRITE privileges.

If passwords are included as part of the incoming access information, they are not displayed when you issue the LIST ACCESS command.

The LIST KNOWN ACCESS command displays all entries contained in the incoming access database. You can also redirect this information to an output file for future reference.

Format

LIST ACCESS USER *user-name* [TO *file-id*]

or

LIST KNOWN ACCESS [TO *file-id*]

Where:

user-name	Represents a 1 to 39 alphabetic character string that defines the user whose access information you want to display. For the username to appear in lowercase letters, be sure to enclose the string in quotation marks.
TO *file-id*	Specifies the name of an output file to direct information to.

Example

The following command displays the contents of the incoming database:

```
NCP>LIST KNOWN ACCESS Return
```

The system responds with:

```
Default incoming access is NONE

  Access
  Type           User/Password

  ALL            GEORGE
  ALL            PAUL/...
  READ           RINGO

NCP>
```

LIST CIRCUIT

The LIST CIRCUIT command displays the permanent parameters you can change with the DEFINE CIRCUIT command. You can either display the information on your screen or redirect it to an output file at the local node.

This command displays information that you can change. To view all the information about the circuit, use the SHOW CIRCUIT command.

Format

```
                         ┌ CHARACTERISTICS ┐
LIST  CIRCUIT [circuit-id] │ STATUS          │ [TO file-id]
                         └ SUMMARY         ┘
```

Where:

CIRCUIT [*circuit-id*] Identifies the circuit (either ASYNC-1 or one of these: ETHER-1 and TOKEN-1) for which information is to be displayed. Ethernet and Token Ring are listed together because they share the same DECnet parameter database (DECPARM.DAT). Because there is one circuit to use, the *circuit-id* is optional.

CHARACTERISTICS Displays parameters set for the executor, line, or circuit.

STATUS Displays the availability of the circuit for network activity.

SUMMARY Displays a summary of information for the circuit.

TO *file-id* Specifies the name of an output file to direct information to.

Examples

1. The following command displays the permanent attributes of circuit ETHER-1:

```
NCP>LIST CIRCUIT ETHER-1 CHARACTERISTICS Return

Circuit Characteristics as of 9-Mar-1991 9:28:12

Circuit state                  = On
Service                        = Enabled
Hello timer                    = 30
Verification                   = 0
Owner                          = DECnet
Multicast listener state       = Enabled
NCP>
```

2. The following command redirects the permanent attribute information about circuit ETHER-1 to an output file named CIR.DAT. To see the information in CIR.DAT you must exit NCP, by typing Ctrl/Z Return, and TYPE the file.

```
NCP>LIST CIRCUIT ETHER-1 SUMMARY TO CIR.DAT Return
C:\>TYPE CIR.DAT Return
Circuit Summary as of 9-Mar-1991 9:30:31

Circuit           State/Substate  Designated      Block
                                  Router          Size

ETHER-1/TOKEN-1   On/Running                      1498
NCP>
```

LIST EXECUTOR

The LIST EXECUTOR command displays permanent information about the local node. This command displays only information that you can change. To view all the current information about the local node, use the SHOW EXECUTOR command.

Format

```
                ┌ CHARACTERISTICS ┐
LIST  EXECUTOR  │ STATUS          │  [TO file-id]
                └ SUMMARY         ┘
```

Where:

CHARACTERISTICS Displays parameters set for the executor, line, or circuit.

STATUS Displays the availability of the executor, line, or circuit for network activity.

SUMMARY Displays a summary of information for the executor, line, circuit, or links.

TO *file-id* Specifies the name of an output file to direct information to.

Example

The following command displays the permanent characteristics that you can set for your executor node:

```
NCP>LIST EXECUTOR CHARACTERISTICS Return
```

A sample system response follows:

```
Executor Characteristics as of  23-Mar-1991 16:17:53

Executor node              = 2.975 (ROCKY)

State                        = On
Executor Identification      = PATHWORKS for DOS V4.1 client.
Node name                    = ROCKY
Node address                 = 2.975
Incoming timer               = 45
Outgoing timer               = 60
Confidence timer             = 15
Incoming proxy               = Disabled
Outgoing proxy               = Enabled
Maximum links                = 8
Delay factor                 = 16
Delay weight                 = 2
Inactivity timer             = 30
Retransmit factor            = 6
Maximum buffers              = 16
Buffer size                  = 1498
Segment buffer size          = 557
Receive password             = (password is not set)
Transmit password            = (password is not set)
Transmit pipe quota          = 8
Receive pipe quota           = 3
Out-of-order nak quota       = 0
Autoboot                     = Enabled
```

LIST LINE

The LIST LINE command displays the permanent parameters that you can change for the DECnet line (ASYNC-1, ETHER-1, or TOKEN-1).

This command displays information that you can change. To view all the information about the line, use the SHOW LINE command.

Format

```
                    ┌ CHARACTERISTICS ┐
LIST  LINE [line-id]│ STATUS          │ [TO file-id]
                    └ SUMMARY         ┘
```

Where:

line-id Identifies the line (either ASYNC-1 or one of these: ETHER-1 and TOKEN-1) for which you want information displayed. Ethernet and Token Ring are listed together because they share the same DECnet parameter database (DECPARM.DAT). Because one line is used for each configuration, the *line-id* is optional.

CHARACTERISTICS Displays parameters set for the executor, line, or circuit.

STATUS Displays the availability of the line for network activity.

SUMMARY Displays a summary of information for the line.

TO *file-id* Specifies the name of an output file to direct information to.

Example

The following command displays permanent information about your line:

```
NCP>LIST LINE CHARACTERISTICS [Return]
```

The system responds with:

```
Line Characteristics as of  23-Mar-1991 16:17:57

Line =  ETHER-1/TOKEN-1

Line state                         = On
Receive buffers                    = 8
NCP>
```

LIST NODE

The LIST NODE command displays permanent local information about a specific remote node.

If you are checking the status of a particular node, that node's name and address must already be defined in the node database. If there is no node database, use the NCP DEFINE NODE command to establish a database.

The LIST NODE *area.** command displays all the nodes in the local database in one specified area.

The LIST KNOWN NODES command displays all the nodes in the local node database.

Format

LIST NODE $\left\{ \begin{matrix} \textit{node-id} \\ \textit{area.*} \end{matrix} \right\}$ [TO *file-id*]

or

LIST KNOWN NODES { MS-NET } [TO *file-id*]

Where:

area Specifies the node area.

node-id Specifies the name or the address of the node for which to display information. A node name is a 1- to 6-character alphanumeric string that identifies a specific node. A node address is a numeric character string consisting of an area number and a specific node number. The node address also serves as a unique identification of the node in the network.

TO *file-id* Specifies the name of an output file to direct information to.

MS-NET Indicates those nodes that are defined as MS-NET servers.

Examples

1. The following command displays information for the node RM5J:

```
NCP>LIST NODE RM5J Return
```

Here is a sample system response:

```
Known permanent nodes as of  20-Mar-1991 16:36:30

Node       Node     Active          Account
Address    Name     Links    MS-NET Information

 4.30      RM5J   1       M
```

2. The following command displays the nodes defined in your network database:

   ```
   NCP>LIST KNOWN NODES Return
   ```

   Here is a sample system response:

   ```
   Known permanent nodes as of  20-Mar-1991 16:36:30

   Executor node            = 2.975 (ROCKY)

   State                      On
   Executor Identification    PATHWORKS for DOS V4.1 client.

   Node       Node     Active          Account
   Address    Name     Links    MS-NET Information

    2.375     BLD3     0               /SMITH/...
    4.30      PRSDNT   1        M
    4.60      BASBAL   0
    4.62      PONTI    0
    4.64      AGNES    0
    4.215     RM10     0
    4.216     RM1A     0
    4.298     DEMBO    0
   55.86      FLR3     0               /D JONES
   55.124     BLDJ     1        M
   55.170     FLR5     0
   55.242     BLDK     0        M
   55.261     RM110    0
   55.335     RM5J     0
   ```

3. The following command displays all the nodes defined for area 4 and have been designated as MS-NET servers:

   ```
   NCP>LIST NODE 4.* MSNET Return
   ```

   Here is a sample system response:

   ```
   Known Permanent Nodes as of 25-Mar-1991 14:06:25

   Executor node              = 2.975 (ROCKY)

   State                        On
   Executor Identification      PATHWORKS for DOS V4.1 client.

   Node       Node     Active          Account
   Address    Name     Links   MS-NET  Information

    4.30      PRSDNT   0       M
    4.60      BLDJ     0       M
    4.298     BLDK     0       M
   ```

LIST OBJECT

The LIST OBJECT command displays information about individual objects contained in the Job Spawner object database (DECOBJ.DAT).

The LIST KNOWN OBJECTS command displays all the objects contained in the object database.

The LIST command is case sensitive. This means that you must specify the object to display by enclosing the name in quotation marks. If you have objects stored in both all capital letters and in mixed-case letters (such as "FAL" and "tell"), you must specify the object exactly as it is stored in the database.

Format

LIST **OBJECT** $\left\{ \begin{array}{l} \textit{object-name} \\ \textit{object-number} \end{array} \right\}$ [TO *file-id*]

or

LIST **KNOWN OBJECTS** [TO *file-id*]

Where:

object-name Specifies the name of the object for which you want to list information. The name can be from 1 to 15 characters long.

object-number Specifies the number of the object for which to list information. The value for this number can be from 0 to 255.

TO *file-id* Specifies the name of an output file to direct information to.

Examples

1. The following command displays information in DECOBJ.DAT about the object FAL:

   ```
   NCP>LIST OBJECT FAL Return
   ```

   The system responds with:

   ```
   DECnet Objects
    #   Taskname  File "Arguments"

    17   FAL      FAL      " "
   ```

2. The following command displays information about all the objects in DECOBJ.DAT:

   ```
   NCP>LIST KNOWN OBJECTS Return
   ```

   The system responds with:

   ```
     DECnet Objects
      #   Taskname  File "Arguments"

      0    tell     tells   " "
      17   FAL      FAL     " "
      63   DTR      DTR     " "
   ```

3. The following command displays information about the object in the Spawner database:

   ```
   NCP>LIST OBJECT Return
   ```

   The default object number is 0. Therefore, if you do not request an object number, NCP displays all objects of number 0. For example:

   ```
     DECnet Objects
      #   Taskname  File "Arguments"

      0    tell     tells   " "
   ```

LIST REMOTE-ADAPTER-NAME

The LIST REMOTE-ADAPTER-NAME command displays information about a specific remote adapter name in the DECREM.DAT database. The remote adapter name identifies a network application that is written for the NETBIOS emulator interface. The application is a server on a remote DECnet system (for example, VMS Services for MS–DOS).

The LIST KNOWN REMOTE-ADAPTER-NAMES command displays information about all the remote adapter names in the DECREM.DAT database.

The LIST command for adapter name information is case sensitive. This means that you must specify the exact name to display by enclosing the name in quotation marks. If you have two remote adapter names that use the same letters but are in different cases, such as "BEST" and "best", you must be very specific when using the LIST REMOTE-ADAPTER-NAME command. If you do not use quotation marks with the LIST command, NCP displays the name that is stored in all capital letters.

Format

LIST REMOTE-ADAPTER-NAME *name* [**BYTE**16 *nn*] [TO *file-id*]

or

CLEAR REMOTE-ADAPTER-NAME/HEX *hh*

or

LIST KNOWN REMOTE-ADAPTER-NAMES [TO *file-id*]

Where:

name Represents a 1- to 16-character alphabetic string that specifies the remote application for which to display information.

TO *file-id* Specifies the name of an output file to direct information to.

nn Specifies a value for byte 16. The 16th character can be a non-printable ASCII character, entered in decimal. This value can be any decimal value from 0 to 255. The default is 20.

hh Specifies a number representing the adapter name in hexadecimal up to 16 pairs of hex digits. The first byte cannot be the hexadecimal value of 0 or 2a.

Note

Do not leave a space between the LIST REMOTE-ADAPTER-NAME command and the /HEX parameter.

Examples

1. The following command displays information about a remote adapter named WINTER:

```
NCP>LIST REMOTE-ADAPTER-NAME Winter Return
```

The system responds with:

```
Permanent Remote-Adapter database as of 9-Mar-1991 9:35:59

Remote-Adapter          Node    Node     Obj     Access
                        Name    Address   #

WINTER                  BLD3   10.497   24

NCP>
```

2. The following commands list remote adapter definitions in the permanent database. These two commands do the same thing.

```
NCP>LIST REMOTE-ADAPTER-NAME SAINTSTEVEN BYTE16 11
```

or

```
NCP>LIS REM/HEX 53-41-49-4E-54-53-54-45-56-45-4E-20-20-20-20-0B
```

The following display appears as a result of the above commands.

```
Permanent Remote-Adapter database as of 25-Mar-1991 15:28:22

Remote-Adapter   BYTE16   Node    Node     Obj  Access
                 in hex   Name    Address

SAINTSTEVEN        0B     LUVLIT  1.2       0
                         "53 41 49 4E 54 53 54 45 56 45 4E 20 20 20 20 0B"
```

LOOP CIRCUIT

The LOOP CIRCUIT command performs loop tests for the DECnet circuit. You can use the loop circuit tests for either asynchronous DDCMP configurations or for Ethernet and Token Ring configurations.

If you are using DDCMP, the tests verify that network communication is possible between your local node and the following:

- The loopback plug or controller.
- The loopback plug or cable.
- The local modem.
- The remote modem.
- The adjacent node.

If you are using Ethernet, or Token Ring, the tests verify that network communication is possible between your local node and the following:

- A specific remote node.
- Any remote node that responds to the loopback request.

When you issue the LOOP CIRCUIT command, NCP determines whether you are using DDCMP or Ethernet/Token Ring on your node. The prompts that NCP returns for this command varies depending upon your node's configuration.

When you run the LOOP CIRCUIT test for DDCMP, NCP prompts you for information needed to run the test. You need to perform one of the following operations:

- Place the loopback plug on the controller.
- Place the loopback plug on the end of the cable that is attached to the controller.
- Place the local modem in loopback mode (by pressing a switch on the modem).
- Place the remote modem in loopback mode (by pressing a switch on the modem).
- If you are performing a loop circuit test to an adjacent node, make sure the adjacent node is set with CIRCUIT STATE SERVICE. (This is necessary for DDCMP configurations.)

When you run the LOOP CIRCUIT test for Ethernet or Token Ring, you do not need to perform these operations. If you issue the command without specifying a responding node, NCP sends out a loop request to the multicast loopback assistance address. This means that any node on the Ethernet (or Token Ring) can reply to the loop request. After a node responds to this request, NCP displays information about the loop message that was sent and the number of the responding node.

When you run any of the LOOP CIRCUIT tests, they check the network path from the local node to the hardware level on the remote node, the modem, or the loopback plug.

The NCP utility performs each test *n* times, as long as the test is successful. The default value for *n* is 1. You can change this value by using the COUNT *n* option after the LOOP command.

When the test is complete, NCP displays a success message. If the test fails, NCP displays an error message.

Format for DDCMP Configurations

```
                                     ⎧ MIXED  ⎫
LOOP CIRCUIT  [ COUNT n ]  WITH  ⎨ ONES   ⎬ [ LENGTH n ]
                                     ⎩ ZEROES ⎭
```

Where:

COUNT *n* — Represents the number of times to repeat the loop test. The default is 1.

LENGTH *n* — Represents the length of the loop message to be sent. The maximum value you can use is 512. The default is 40.

WITH — Specifies the message type. The default is MIXED.

- MIXED – Indicates that the message type should be mixed ones and zeroes.
- ONES – Indicates that the message type should be all ones.
- ZEROS – Indicates that the message type should be all zeroes.

Format for Ethernet and Token Ring Configurations

```
                                                  ⎧ MIXED  ⎫
LOOP CIRCUIT  NODE node-id          [COUNT n] [WITH ⎨ ONES   ⎬] [LENGTH n]
              STATION-ADDR addr                   ⎩ ZEROES ⎭
```

Where:

COUNT *n* — Represents the number of times to repeat the loop test. The default is 1.

LENGTH *n* — Represents the length of the loop message to be sent. The maximum value you can use is 512. The minimum value is 46. This is also the default.

NODE *node-id* — Specifies the name of the designated node to which to send loop messages. You can use either the node's name or its address. You must have defined the node using the the SET NODE command.

STATION-ADDRESS *addr* — Specifies a hexadecimal number *hh-hh-hh-hh-hh-hh* representing the actual physical address of the node you are trying to contact on the local Ethernet or Token Ring. The range for this number is 0 - F.

WITH — Specifies the message type. The default is MIXED.

- MIXED – Indicates that the message type should be mixed ones and zeroes.
- ONES – Indicates that the message type should be all ones.
- ZEROS – Indicates that the message type should be all zeroes.

Examples

1. The following example illustrates a DDCMP loop circuit test:

```
NCP>LOOP CIRCUIT Return
```

The system responds with:

```
LOOP CIRCUIT (DDCMP) - Do one of the following, then press any key to
begin test.  Check the Network Management Guide for details.

1.  Place loopback plug on controller.

2.  Place loopback plug on end of cable attached to controller.

3.  Place local modem in loopback.
```

```
4. Place remote modem in loopback.

5. For circuit loopback to the adjacent node, be sure the adjacent
   node is set with CIRCUIT STATE SERVICE.
```

2. The following example illustrates an Ethernet loop circuit test to a multicast address:

   ```
   NCP>LOOP CIRCUIT Return
   ```

   The system responds with:

   ```
   LOOP CIRCUIT test started at  21-Mar-1991 10:22:21

   Sending loop message  1, 46.
   Message echoed by remote circuit loopback 1, 46 bytes.

   Multicast Loop Circuit echoed by node 55.62

   LOOP CIRCUIT test finished successfully at 21-Mar-1991 10:22:21

   NCP>
   ```

3. The following example illustrates an Ethernet loop circuit test to the node 55.101 with a count of 3:

   ```
   NCP>LOOP CIRCUIT NODE 55.101 COUNT 3 Return
   ```

   The system responds with:

   ```
   LOOP CIRCUIT test started at 21-Mar-1991 16:29:08
   Sending loop message 1, 46.
   Message echoed by remote circuit loopback 1, 46 bytes.
   Sending loop message 2, 46.
   Message echoed by remote circuit loopback 2, 46 bytes.
   Sending loop message 3, 46.
   Message echoed by remote circuit loopback 3, 46 bytes.
   LOOP CIRCUIT test finished successfully at 21-Mar-1991 16:29:43
   ```

LOOP EXECUTOR

The LOOP EXECUTOR command invokes the loop test for your local (executor) node. It verifies the operation of your local node by checking the local network software. This test verifies that your local node software has been properly installed.

When you issue this command, NCP performs the test *n* times, as long as the test is successful. The default value for *n* is 1. You can change this value by using the COUNT *n* option after the LOOP command.

Format

$$\text{LOOP EXECUTOR} \quad \text{[COUNT } n\text{] WITH} \left\{ \begin{array}{l} \text{MIXED} \\ \text{ONES} \\ \text{ZEROES} \end{array} \right\} \text{[LENGTH } n\text{]}$$

Where:

COUNT *n* — Represents the number of times to repeat the loop test. The default is 1.

LENGTH *n* — Represents the length of the loop message to be sent. The maximum value you can use is 512. The default is 46.

WITH — Specifies the message type. The default is MIXED.

- MIXED – Indicates that the message type should be mixed ones and zeroes.
- ONES – Indicates that the message type should be all ones.
- ZEROS – Indicates that the message type should be all zeroes.

Example

The following example requests that the loop test be performed 10 times, with mixed ones and zeroes and a message length of 50:

```
NCP>LOOP EXECUTOR COUNT 10 WITH MIXED LENGTH 50 Return
```

The system responds with:

```
LOOP NODE test started at 9-Mar-1991 9:39:05

Connect complete to node ROCKY
Remote node maximum buffer size for loopback: 512
Successful send and receive, message 1.
Successful send and receive, message 2.
Successful send and receive, message 3.
Successful send and receive, message 4.
Successful send and receive, message 5.
Successful send and receive, message 6.
Successful send and receive, message 7.
Successful send and receive, message 8.
Successful send and receive, message 9.
Successful send and receive, message 10.
LOOP NODE test finished successfully at 9-Mar-1991 9:39:08

NCP>
```

LOOP LINE CONTROLLER

The LOOP LINE CONTROLLER command performs a loopback test within the local node at the device (or controller) level.

The LOOP LINE CONTROLLER test requires the use of the loopback connector. You must attach this connector to the Ethernet controller to perform this test.

When you issue the LOOP LINE CONTROLLER command, NCP prompts you for information needed to run the test. There are no arguments or switches associated with this command.

This command is not supported on the 3Com EtherLink (3C503).

Format

LOOP **LINE** **CONT**ROLLER

Example

The following example illustrates a sample loop line controller test and possible system response:

```
NCP>LOOP LINE CONTROLLER Return

LOOP LINE CONTROLLER (Ethernet)

1.  Unplug network cable from Controller on back of your computer.

2.  Place the loopback plug on the Controller.  Test will fail if

    loopback plug is not in place.

3.  Press any key to begin test.

4.  At the end of the test, reconnect the Controller to the network.

LOOP LINE CONTROLLER test started at 19-Mar-1991  11:08:47

LOOP LINE CONTROLLER test finished successfully at 19-Mar-1991  11:08:49

    Please remove loopback plug and reconnect your node to the network.

    Press any key to continue.
```

LOOP NODE

The LOOP NODE command invokes the loop test to the remote node. It verifies that your node can communicate with a specified remote node. To perform this test, you need to include any network parameters necessary for accessing the loopback mirror on the remote node. These parameters include the node ID, the user ID, the password, and the account.

It is not necessary to include access information for all systems. Most systems do not require this information for performing the remote node test.

When you issue this command, NCP performs the test *n* times, as long as the test is successful. The default value for *n* is 1. You can change this value by using the COUNT *n* option after the LOOP command.

Before running the test, the LOOP NODE command sets the line and circuit states to ON. When the test is complete, NCP displays a success message. If the test is unsuccessful, NCP displays an error message.

If you do not specify user information for this command, NCP uses the default information that was set with the DEFINE NODE command.

Format

```
                     ( USER user-id          )
                     | PASSWORD password     |
                     | ACCOUNT account       |
LOOP NODE node-id   <  COUNT n                >
                     |      ( MIXED   )      |
                     | WITH < ONES     >     |
                     |      ( ZEROES  )      |
                     ( LENGTH n              )
```

Where:

node-id Specifies the name or the address of a remote node. A node name is a 1- to 6-character alphanumeric string that identifies a specific node. A node address is a numeric character string consisting of an area number and a specific node number. The node address also serves as a unique identification of the node in the network.

user-id Represents the user information you can use for accessing the loopback mirror on the remote node. A user ID consists of 1 to 39 alphabetic characters.

password Represents a unique character string that you can use for accessing the loopback mirror on the remote node. A user's password consists of 1 to 39 alphanumeric characters.

account Represents alphanumeric information for accessing the adjacent node's loopback mirror. The information can consist of 1 to 39 alphanumeric characters.

COUNT *n* Represents the number of times to repeat the loop test. The default is 1.

LENGTH *n* Represents the length of the loop message to be sent. The default is 46.

WITH Specifies the message type. The default is MIXED.

- MIXED – Indicates that the message type should be mixed ones and zeroes.
- ONES – Indicates that the message type should be all ones.
- ZEROS – Indicates that the message type should be all zeroes.

Example

The following command specifies the node RM5J as the remote node to be used in the remote node test. It specifies that the loop message should be repeated 5 times with a length of 46.

```
NCP>LOOP NODE RM5J COUNT 5 LENGTH 46 Return
```

Here is a sample system response:

```
LOOP NODE test started at 4-Mar-1991  14:32:14
  Connect complete to node RM5J
  Remote node maximum buffer size for loopback:  4096

Sending loop message 1, 46 bytes.
Successful send and receive, message 1.
Sending loop message 2, 46 bytes.
Successful send and receive, message 2.
Sending loop message 3, 46 bytes.
Successful send and receive, message 3.
Sending loop message 4, 46 bytes.
Sending loop message 5, 46 bytes.
Successful send and receive, message 4.
Successful send and receive, message 5.

LOOP NODE test finished successfully at 4-Mar-1991 14:32:15
```

The messages do not necessarily return before the next message is sent (messages 4 and 5 in this example). This is normal output for the LOOP NODE command.

MIRROR

Use the MIRROR command for LOOP NODE tests from a remote node. The MIRROR command allows the LOOP NODE test to be run from remote systems that support the NCP loop node test to your local node. The maximum length of a message that NCP loops back is 512 bytes.

Not all loop tests are available on all remote nodes. The tests you can use depend on the tests that are available at the remote node. The information required to run the test also depends on the remote node you are using.

The MIRROR command starts up a loopback mirror (DECnet object ID 25), that echoes test messages that it receives from a remote node back to that same remote node. When the test run from the remote node is complete, press any key to return to the NCP command level.

Format

MIRROR

Example

The following command starts the loopback mirror on your node:

```
NCP>MIRROR Return
```

NCP displays the following information:

```
Loopback Mirror waiting to echo LOOP NODE test from remote node.
Press any key to stop Loopback Mirror.
```

If NCP detects a loop node test from a remote node, a message like this appears:

```
  Connect request received from node  4.299 (FREMI), object R SMITH
```

After you press a key to stop the mirror, NCP displays a message and returns you to the NCP prompt.

```
    Loopback mirror stopped
NCP>
```

MONITOR LOGGING

The MONITOR LOGGING command continually displays network events as they occur. The logging of network events pauses when there are no more events to display. You can stop the display of information logging and return to the NCP prompt by pressing any key on the keyboard.

After the events are displayed, NCP disregards the the information in the event buffer. To retain the logging information, you can redirect it to an output file. This is useful if you wish to store the information to analyze it later or if you need to show the information to a network support person. If you specify a file for storing log information, you can stop the input of information into the file by pressing any key.

Format

MONITOR **LOG**GING [TO *file-id*]

Where:

TO *file-id* Specifies the name of an output file to direct information to.

Example

The following command displays logging information on your screen.

```
NCP>MONITOR LOGGING Return
```

Here is a sample system response:

```
Events logged as of 17-Mar-1991 11:17:41

Event type 4.10  Circuit up
Occurred 17-Mar-1991  11:18:00
Adjacent Node:  55.2  (LOWEND)

Event type 4.10   Circuit up
Occurred 17-Mar-1991  11:18:15
Adjacent Node:  55.2  (LOWEND)
.
.
.
```

PURGE ACCESS

The PURGE ACCESS command removes incoming access information for the designated user. The access information is stored for each user, and includes the password and access type.

Note

If access for a particular user is not defined, NCP uses the default access instead.

The PURGE KNOWN ACCESS command removes all incoming access information for all users in the incoming access database.

The PURGE command is case sensitive. This means that you must specify the user information in quotation marks if it contains any lowercase letters. If you do not specify the proper format when you issue the PURGE command, NCP does not purge the information from the database.

Format

PURGE **ACCESS USER** *user-name*

or

PURGE **KNOWN ACCESS**

Where:

user-name Represents a 1- to 39-character alphabetic string that defines the user whose access information you want to remove.

Examples

1. The following command removes the entire entry for the user BARKER:

   ```
   NCP>PURGE ACCESS USER BARKER Return
   ```

2. The following command removes the entire entry for the user paul:

   ```
   NCP>PURGE ACCESS USER "paul" Return
   ```

   If you typed the user name paul without quotes, NCP attempts to purge an entry for PAUL. If one existed, NCP removes it; if not, NCP sends an error message.

PURGE EXECUTOR

The PURGE EXECUTOR command removes specified executor node parameters from the permanent database. The parameters you can remove are RECEIVE PASSWORD and TRANSMIT PASSWORD. (To remove these parameters from the running system, use the CLEAR EXECUTOR command.)

The RECEIVE and TRANSMIT passwords are automatically exchanged when the executor or remote node turns on the communication line between itself and the adjacent node. The network coordinator defines these passwords for DDCMP use. The exchange of passwords also occurs when the communication line is restarted from either node because of errors.

Format

```
PURGE EXECUTOR  { RECEIVE PASSWORD  }
                { TRANSMIT PASSWORD }
```

Where:

RECEIVE PASSWORD	Removes the password the executor node receives from the adjacent nod to exchange messages with the adjacent node. The receive password is used if password checking is enabled on both nodes.
TRANSMIT PASSWORD	Removes the password the executor node must transmit to the adjacent node to exchange messages with the adjacent node.

Note

These passwords are used for asynchronous DDCMP connections. They are used if password checking is enabled on both nodes.

Example

The following command removes the password that the executor node must transmit to the adjacent node when the executor node first begins communications. Until the network coordinator defines a new transmit password, the executor node cannot open the communication line between itself and the adjacent node.

```
NCP>PURGE EXECUTOR TRANSMIT PASSWORD [Return]
```

PURGE NODE

The PURGE NODE command removes parameters stored on the local node for the specified remote node or all remote nodes known to the local node.

The PURGE KNOWN NODES command removes information from the local node permanent database (DECNODE.DAT), including any related user, password, and account information.

Format

```
                       ┌ USER      ┐
PURGE NODE  node-id    │ PASSWORD  │
                       │ ACCOUNT   │
                       └ MS-NET    ┘
```

or

```
PURGE KNOWN NODES
```

Where:

NODE *node-id*	Specifies the node name (or node address) whose parameters are to be removed.
USER	Removes the user-name, password, and accounting information for the specified node.
PASSWORD	Removes the password associated with the user-name.
ACCOUNT	Removes accounting information for the specified node.
MS-NET	Removes the indicator that identifies the node as an MS-NET server.

Example

The following command removes all parameters for node PARIS from the local node's permanent database.

```
NCP>PURGE NODE PARIS [Return]
```

PURGE OBJECT

The PURGE OBJECT command deletes information in the Job Spawner database (DECOBJ.DAT) for a specific object. You can delete an object by indicating either the object name or the object number.

The PURGE KNOWN OBJECTS command deletes all the information about all the objects in DECOBJ.DAT.

The PURGE command is case sensitive. This means that you must specify the object to remove by enclosing the name in quotation marks if the name contains any lowercase letters. If you do not specify the proper object form when you issue the PURGE command, NCP does not find the matching object in the database.

Format

```
               { object-name   }
PURGE OBJECT   { object-number }
```

or

```
PURGE   KNOWN OBJECTS
```

Where:

object-name Specifies the name of the object for which to delete information. The name can be from 1 to 15 characters long.

object-number Specifies the number of the object for which to delete information. The value for this number can be from 0 to 255.

Example

The following command deletes information about the object FAL:

```
NCP>PURGE OBJECT FAL [Return]
```

PURGE REMOTE-ADAPTER-NAME

The PURGE REMOTE-ADAPTER-NAME command deletes an entry from the permanent remote adapter name database (DECREM.DAT). The remote adapter name identifies a network application that is written for the NETBIOS emulator interface. The application resides on a remote personal computer.

The PURGE KNOWN REMOTE-ADAPTER-NAMES command deletes all the remote adapter names (and associated information) from DECREM.DAT.

Format

```
PURGE REMOTE-ADAPTER-NAME  name [BYTE16 nn]
```

or

```
PURGE  REMOTE-ADAPTER-NAME/HEX hh
```

or

```
PURGE  KNOWN REMOTE-ADAPTER-NAMES
```

Where:

name	Represents a 1- to 16-character alphabetic string that specifies the remote application to remove from the permanent database.
nn	Specifies a value for byte 16. The 16th character can be a non-printable ASCII character, entered in decimal. This value can be any decimal value from 0 to 255. The default is 20.
hh	Specifies a number representing the adapter name in hexadecimal up to 16 pairs of hex digits. The first byte cannot be the hexadecimal value of 0 or 2a.

Note

Do not leave a space between the PURGE REMOTE-ADAPTER-NAME command and the /HEX parameter.

Examples

1. The following command deletes information about the remote adapter named WINSTON:

   ```
   NCP>PURGE REMOTE-ADAPTER-NAME WINSTON Return
   ```

2. The following commands remove remote adapter name definitions from the permanent database. These two commands do the same thing:

   ```
   NCP>PURGE REMOTE-ADAPTER-NAME SAINTSTEVEN BYTE16 11
   ```

   or

   ```
   NCP>PUR REM/HEX 53-41-49-4E-54-53-54-45-56-45-4E-20-20-20-20-0B
   ```

READ LOG

The READ LOG command displays the contents of the event-logging buffer. After the READ LOG command displays the contents of the buffer, it disregards the contents of the event-logging buffer. To retain the logging information, you can redirect it to an output file. This is useful if you wish to store the information in order to analyze it later or if you need to show the information to a network support person.

Format

READ LOG [TO *file-id*]

Where:

TO *file-id* Specifies the name of an output file to direct information to.

Examples

1. The following command redirects the information in the event-logging buffer to the file LOGFILE.TXT. After the information has been redirected to the output file, it is discarded from the event-logging buffer.

   ```
   NCP>READ LOG TO LOGFILE.TXT [Return]
   ```

   You can now exit from NCP and display the contents of this particular event buffer by viewing the output file you created.

2. The following command displays the contents of the event-logging buffer on the screen:

   ```
   NCP>READ LOG [Return]
   ```

Here is a sample system response:

```
Events logged as of 17-Jun-1991 11:17:41
Event type 4.10  Circuit up
Occurred 17-Jun-1991  11:18:00
Adjacent Node:  55.2  (LOWEND)

Event type 4.7   Circuit down - circuit fault
Occurred 17-Jun-1991  11:18:09
Adjacent Node:  55.2  (LOWEND)
Reason: Line synchronization lost

Event type 4.10   Circuit up
Occurred 17-Jun-1991  11:18:15
Adjacent Node:  55.2  (LOWEND)

NCP>
```

3. To view the contents of LOGFILE.TXT, enter the following command:

```
C:\>TYPE LOGFILE.TXT Return
```

The system responds by displaying the contents of your output file :

```
Events Logged as of 17-Aug-1991 16:46:24

Event type  0.0  Event records lost
Occurred  17-Aug-1991 10:12:48
Records lost:  22

Event type  4.18  Adjacency down - circuit fault
Occurred  17-Aug-1991 10:22:19
Adjacent node:  55.261 (DSRVC1)
Reason:  Adjacency address change

Event type  4.15  Adjacency up
Occurred  17-Aug-1991 10:34:19
Adjacent node:  55.152

Event type  4.18  Adjacency down - circuit fault
Occurred  17-Aug-1991 10:50:02
Adjacent node:  55.261 (DSRVC1)
Reason:  Adjacency address change

Event type  4.15  Adjacency up
Occurred  17-Aug-1991 10:59:02
Adjacent node:  55.152
NCP>
```

SET CIRCUIT

The SET CIRCUIT command modifies current circuit information.

Most parameters are set by your network coordinator. Therefore, you must obtain this information from that person.

Format

```
                          ⎧ HELLO TIMER seconds                       ⎫
                          ⎪ MULTICAST-LISTENER { DISABLED }           ⎪
                          ⎪                    { ENABLED  }           ⎪
                          ⎪ OWNER { DECNET }                          ⎪
SET CIRCUIT  [circuit-id] ⎨       { MOP    }                          ⎬
                          ⎪ SERVICE { DISABLED }                      ⎪
                          ⎪         { ENABLED  }                      ⎪
                          ⎪ STATE { ON  }                             ⎪
                          ⎩       { OFF }                             ⎭
```

Where:

CIRCUIT [*circuit-id*]
Specifies the circuit (ASYNC-1, ETHER-1, or TOKEN-1) for which parameters are to be defined or modified. Because one circuit is used for each configuration, the *circuit-id* is optional.

HELLO TIMER *seconds*
Specifies the frequency of routing hello messages sent to the adjacent node over the circuit. The valid range is 1 to 8191 seconds. The default is 30.

MULTICAST-LISTENER
Sets the circuit's multicast-listener state. The default is enabled.

- DISABLED – Specifies that the circuit will not respond to multicast requests. This applies to Ethernet communications.
- ENABLED – Specifies that the circuit responds to multicast requests. This applies to Ethernet communications.

OWNER
Specifies the circuit owner.

SERVICE

Sets the circuit's service state. The default is ENABLED.

- DISABLED – Specifies that the circuit is unavailable for service operations.
- ENABLED – Specifies that the circuit is available for special network activity such as loopback testing.

STATE

Sets the line's operational state to ON or OFF. The default state is ON.

- ON – Allows network traffic to flow over the circuit. This is the normal operational state.
- OFF – Allows no traffic to flow over the circuit. The circuit is unavailable for any network activity.

Example

The following command identifies the owner of circuit ETHER-1 as DECnet:

```
NCP>SET CIRCUIT ETHER-1 OWNER DECNET Return
```

SET ECHO

The SET ECHO command determines whether NCP displays a redirected input file on your screen as the file is executed. The default is OFF (no echo).

You can use the SET ECHO command within a command file. You cannot use SET ECHO at the NCP prompt (using it this way will have no effect).

Format

$$\text{SET ECHO} \quad \begin{Bmatrix} \text{ON} \\ \text{OFF} \end{Bmatrix}$$

Where:

ON Turns the echo on, causing the file's contents to be displayed on the screen as the file is executed.

OFF Turns the echo off.

To use SET ECHO, you should set up a command file that contains the SET ECHO command as well as other NCP commands.

Example

This is a sample command file, named SAMPLE.TXT:

```
SET ECHO ON
SHOW EXECUTOR CHARACTERISTICS
EXIT
```

If you run your SAMPLE.TXT file as a command file, you see the results of the SHOW EXECUTOR CHARACTERISTICS command on the screen. Then, NCP exits. For example:

```
C:\>NCP < SAMPLE.TXT Return
NCP>SHOW EXECUTOR CHARACTERISTICS Return


Executor Characteristics as of  23-Oct-1991 16:17:39

Executor node                = 2.975 (ROCKY)

Driver version number        = 4.0.0
State                        = On
Executor Identification      = PATHWORKS for DOS V4.1 client.
Executor version             = 4.0.1
Node name                    = ROCKY
Node address                 = 2.975
Incoming timer               = 45
Outgoing timer               = 60
Confidence timer             = 15
Incoming proxy               = Disabled
Outgoing proxy               = Enabled
Active links                 = 3
NSP version                  = 4.0.0
Maximum links                = 8
Delay factor                 = 16
Delay weight                 = 2
Inactivity timer             = 30
Retransmit factor            = 6
Routing version              = 2.0.0
Routing types                = Nonrouting IV
Maximum buffers              = 24
Buffer size                  = 1498
Segment buffer size          = 557
Receive password             = (password is not set)
Transmit password            = (password is not set)
Database path                = C:\DECNET\
Transmit pipe quota          = 8
Receive pipe quota           = 3
PC type                      = IBM PCAT
Out-of-order nak quota       = 0
Autoboot                     = Enabled
NCP>EXIT Return

C:\>
```

SET EXECUTOR

The SET EXECUTOR command sets the executor node's current parameters or characteristics. These settings are not preserved across network restarts. (For the settings to take effect the next time you boot your system, use the DEFINE EXECUTOR command.)

Format

```
              ADDRESS node-address
              CONFIDENCE TIMER seconds
              DELAY FACTOR 1/16 of a second
              DELAY WEIGHT number
              IDENTIFICATION id-string
              INACTIVITY TIMER seconds
              INCOMING TIMER seconds
              NAK QUOTA number
SET EXECUTOR  NAME node-name
              OUTGOING TIMER seconds
              RECEIVE PASSWORD password
              RECEIVE PIPE QUOTA number
              RETRANSMIT FACTOR number
                     { ON  }
              STATE  { OFF }
              TRANSMIT PASSWORD password
              TRANSMIT PIPE QUOTA number
```

Where:

ADDRESS
node-address

Specifies the address assigned to the executor node. The node address is a unique numeric identification for the node. It must include an area number and a node number. You can set ADDRESS when DNP is installed and the executor's state is OFF. The executor's state is OFF if the DECnet network process fails to find the DECnet database file, DECPARM.DAT or it is defined as OFF.

CONFIDENCE
TIMER *seconds*

Specifies an optional extra amount of time to wait before timing a link out. The value range for this timer is 1 to 65535. The default is 15.

DELAY FACTOR *1 /16 of a second*

Determines the amount of time to wait before a message is retransmitted to a node. The range is 16 to 4095. The default number is 32.

DELAY WEIGHT *number*

Represents the weight to apply to a current round trip delay to a remote node when updating the estimated round-trip delay to a node. The range is 1 to 255. The default is 1.

IDENTIFICATION *id-string*

Specifies a text identification string for the executor node. The string can consist of 1 to 32 alphanumeric characters. You must use a set of double quotation marks (" ") to delimit any string containing blanks or tabs. You cannot use a single set of quotation marks within the identification string.

You can use the IDENTIFICATION parameter to identify the executor node to other nodes in the network (for example, by including your name or department in the string).

INACTIVITY TIMER *seconds*

Represents the maximum duration of inactivity on a logical link before the node checks to see if the logical link still works. The value range is 1 to 65535. The default is 30.

INCOMING TIMER *seconds*

Represents the maximum duration between the time a connection request is received at the executor node and the time that it takes to accept or reject the request. If the request is not accepted or rejected by the user program within the specified number of seconds, the connection request is automatically rejected. If no value is set, there is no timer. The value range is 1 to 65535. The default is 45.

NAK QUOTA *number*

Represents the number of out-of-order segments to receive before sending a negative acknowledgment (NAK) out on the logical link. This number must be less than or equal to the RECEIVE PIPE QUOTA. The value range for this quota is 0 to 65535. The default is 0 (or 6 for the 3Com driver).

NAME *node-name*

Specifies the node name for the executor node. The node name is an identification string consisting of one to six alphanumeric characters including at least one alphabetic character. You can set NAME when DNP is installed and the executor's state is OFF.

OUTGOING TIMER *seconds*

Represents the duration between the time that the executor node requests a connection and the time that it takes for the request to be acknowledged at the remote node. The value range is 1 to 32K. The default is 60.

RECEIVE PASSWORD *password*

Represents the password the local node must receive from the adjacent node in order to exchange messages with the adjacent node. The password is a character string consisting of eight alphanumeric characters.

RECEIVE PIPE QUOTA *number*

Represents the number of segments the network allows to be received for a link by flow control. The valid range is 1 to 4095. The default is 6.

RETRANSMIT FACTOR *number*

Represents the maximum number of times that the executor node restarts the retransmission timer when it expires. If the number is exceeded, the logical link is disconnected. The number is a decimal value in the range of 1 to 65535. The default is 12.

STATE

Sets the executor state to ON or OFF.

- ON – Allows logical links to be established to the executor node. You must SET EXECUTOR ADDRESS (and NAME, if desired) before you can SET EXECUTOR STATE ON.
- OFF – Deinstalls DNP and stops DECnet. To restart DECnet you must rerun DNP. If DNP is loaded in EMS, you cannot set the EXECUTOR STATE OFF. You will receive a "Permission Denied" message.

TRANSMIT PASSWORD *password*

Represents the password the local node must transmit to the adjacent node to exchange messages with the adjacent node. The password is a character string of eight alphanumeric characters.

TRANSMIT PIPE QUOTA *number*

Represents the number of segments the driver buffers into NSP before it blocks the user on a transmission. (NSP is a Network Services Protocol that NCP uses to perform network management and to exchange messages over logical links.) The valid range is 1 to 65535. The default is 6.

Note

Information about RECEIVE PASSWORD and TRANSMIT PASSWORD applies to asynchronous DDCMP communications.

Examples

1. The following command indicates that the executor node waits 10 seconds between the time that a connection request is sent and the time it takes for an acknowledgment to be sent by the destination node.

   ```
   NCP>SET EXECUTOR OUTGOING TIMER 10 Return
   ```

2. The following command sets the executor node's address to 2.975 and its name to ROCKY:

   ```
   NCP>SET EXECUTOR ADDRESS 2.975 NAME ROCKY Return
   ```

3. The following command sets the executor node's state to ON. You must also turn the line state to ON with the SET LINE command. The node can then establish logical links with other nodes.

   ```
   NCP>SET EXECUTOR STATE ON Return
   ```

SET LINE

The SET LINE command sets line information in the volatile database. The values you set are not preserved across network restarts. (For the settings to take effect the next time you boot your system or restart DECnet, use the DEFINE LINE command.)

The person who coordinates activity on your network sets most parameters, such as the baud rate. Therefore, you must obtain this information from that person.

The executor state must be ON for you to define or modify any line parameters. In addition, the line state must be OFF to specify any parameters other than STATE OFF or RETRANSMIT TIMER. You must execute a SET LINE STATE OFF command before you define or modify any other line parameters. When you are finished, execute a SET LINE STATE ON command to turn the line back on.

Setting the state of the line to OFF in an asynchronous configuration shuts down network processing, but does NOT hang up the phone, even if the circuit's modem parameter is set to full. If you must hang up the phone under program control, use the NCP command SET LINE STATE HANGUP.

Format

```
                 ( DEVICE [device-id]         )
                 | MODEM { NULL }             |
                 |       { FULL }             |
SET LINE [line-id] { SPEED baud-rate          }
                 |       ( OFF    )           |
                 | STATE { ON     }           |
                 (       ( HANGUP )           )
```

Where:

LINE [*line-id*] — Specifies either (ASYNC-1, ETHER-1, or TOKEN-1) as the line for which parameters are to be created or modified. Because one line is used for each configuration, the *line-id* is optional.

DEVICE *device-id* — Specifies the device for controlling the line (COM-1, COM-2). The line must be in the OFF state.

MODEM

Specifies the type of modem control to use. Modem control is used with asynchronous DDCMP connections.

- NULL – Specifies that modem control signals will not be checked.
- FULL – Specifies that modem control signals will be checked. The line will be shut down if the modem loses its connection.

SPEED *baud-rate*

Sets the DDCMP line speed to one of the following:

50
75
110
134
150
200
300
600
1200
1800
2000
2400
3600
4800
7200
9600
19200

The default is 9600. For high performance systems, a speed of 19200 is available. Line speed 19200 does not work on 8086 machines. Information about line speed applies to asynchronous DDCMP connections.

STATE

Sets the line's operational state to one of the following options:

- ON – Sets the line state to ON (To set this parameter, the executor line state must be ON.)
- OFF – Sets the line state to OFF
- HANGUP – Sets the line state to OFF and hangs up modem

For asynchronous configurations, you must set the line state to off to use the communications port for nonDECnet communications.

Note

Information about modem, line speed, and device applies to asynchronous DDCMP communications.

Examples

1. The following command sets the speed of line ASYNC-1 to 9600. Line speed is set if you are using asynchronous DDCMP connections. You can change the line speed if the line state is OFF.

   ```
   NCP>SET LINE ASYNC-1 SPEED 9600 Return
   ```

2. The following command specifies the operational state of the line as off:

   ```
   NCP>SET LINE STATE OFF Return
   ```

SET LINK

The SET LINK command terminates logical link connections associated with a specific socket. The socket is also detached.

The SET KNOWN LINKS command disconnects all link connections not set with the socket option, SO_KEEPALIVE, and releases the associated sockets.

To determine what links are active, type SHOW KNOWN LINKS.

A socket is an addressable endpoint of communication within a program or a task. A task uses the socket to send and receive data to and from another task. The SO_KEEPALIVE option keeps any links associated with the socket active, even if you try to disconnect the links. As an example, links established by the Network Device Utility (NDU) for the virtual disk facility are set as KEEPALIVE. Thus, using the SET KNOWN LINKS command would not affect the NDU links.

For more information about sockets and socket options, consult the PATHWORKS for DOS, *DECnet Programmer's Reference Manual.*

Use the SET LINK or SET KNOWN LINKS commands if you have run out of links and you want to recover without performing a network restart. **You should be very familiar with the purpose of the socket (or sockets) whose state you want to change.**

Format

SET LINK *socket* **STATE OFF**

or

SET KNOWN LINKS STATE OFF

Where:

LINK *socket* — Specifies the socket to be detached.

STATE OFF — Turns the logical link off.

KNOWN LINKS STATE OFF — Breaks all current links whose sockets are not set with SO_KEEPALIVE. The sockets can be reassigned to other tasks.

Examples

1. The following command detaches socket 3 and breaks its logical links:

   ```
   NCP>SET LINK 3 STATE OFF Return
   ```

2. The following command aborts all current links and releases the associated sockets:

   ```
   NCP>SET KNOWN LINKS STATE OFF Return
   ```

SET LOCAL-ADAPTER-NAME

The SET LOCAL-ADAPTER-NAME command enters a local NETBIOS name into the volatile database. This information is not preserved across network restarts. The local adapter name identifies a network application that is written for the NETBIOS emulator interface. The application resides on your local personal computer.

The SET command for adapter name information is case sensitive. This means that you must specify the name to set by enclosing the name in quotation marks if the name contains any lowercase letters. If you have two local adapter names that use the same letters but are in different cases, such as "BEST" and "best", you must be very specific when using the SET LOCAL-ADAPTER-NAME command.

Format

```
SET LOCAL-ADAPTER-NAME  name [BYTE16 nn]
```

or

```
SET LOCAL-ADAPTER-NAME/HEX hh
```

Where:

name	Represents a 1- to 16-character ASCII string that specifies the name of the local application to add to the volatile database.
nn	Specifies a value for byte 16. The 16th character can be a non-printable ASCII character, entered in decimal. This value can be any decimal value from 0 to 255. The default is 20.
hh	Specifies a number representing the adapter name in hexadecimal up to 16 pairs of hex digits. The first byte cannot be the hexadecimal value of 0 or 2a.

Note

Do not leave a space between the SET LOCAL-ADAPTER-NAME command and the /HEX parameter.

Example

1. The following command enters myapp in the volatile database:

   ```
   NCP>SET LOCAL-ADAPTER-NAME "myapp" Return
   ```

2. The following commands add local adapter definitions to the volatile database. These two commands do the same thing:

   ```
   NCP>SET LOCAL-ADAPTER-NAME OTHERONE BYTE16 1
   ```

   or

   ```
   NCP>SET LOC/HEX 4F-54-48-45-52-4F-4E-45-20-20-20-20-20-20-20-01
   ```

SET PAUSE

The SET PAUSE command enables or disables display characteristics. If you issue an NCP command that displays several screens of data (such as LIST or SHOW), some of the information can scroll by before you can read it. SET PAUSE lets you control this display of information. For example, if you issue a SET PAUSE command and then issue a LIST command, the display stops at the end of the first screen of data. The NCP utility issues a –More– prompt for you to continue. When you see this prompt, you can press Return to view the rest of the display.

Format

$$\text{SET PAUSE} \begin{Bmatrix} \text{ON} \\ \text{OFF} \end{Bmatrix}$$

Example

The following command turns the PAUSE option ON.

```
NCP>SET PAUSE ON Return
```

Now when you issue a SHOW EXECUTOR CHARACTERISTICS command, the display stops at the end of the first screen and the –More– prompt appears. You must then press Return before you can view the rest of the display.

For example:

```
NCP>SHOW EXECUTOR CHARACTERISTICS Return

Executor Characteristics as of  23-Oct-1991 16:17:39

Executor node                 = 2.975 (ROCKY)

Driver version number         = 4.0.0
State                         = On
Executor Identification       = PATHWORKS for DOS V4.1 client.
Executor version              = 4.0.1
Node name                     = ROCKY
Node address                  = 2.975
Incoming timer                = 45
Outgoing timer                = 60
Confidence timer              = 15
Incoming proxy                = Disabled
Outgoing proxy                = Enabled
Active links                  = 3
NSP version                   = 4.0.0
Maximum links                 = 8
Delay factor                  = 16
Delay weight                  = 2
Inactivity timer              = 30
Retransmit factor             = 6
Routing version               = 2.0.0
Routing types                 = Nonrouting IV

--More-- Return

Maximum buffers               = 24
Buffer size                   = 1498
Segment buffer size           = 557
Receive password              = (password is not set)
Transmit password             = (password is not set)
Database path                 = C:\DECNET\
Transmit pipe quota           = 8
Receive pipe quota            = 3
PC type                       = IBM/PCAT
Out-of-order nak quota        = 0
Autoboot                      = Enabled
NCP>
```

SET REMOTE-ADAPTER-NAME

The SET REMOTE-ADAPTER-NAME command enters a remote adapter name into the volatile database. This information is not preserved across network restarts. (For the settings to take effect the next time you boot your system, use the DEFINE REMOTE-ADAPTER-NAME command.)

The remote adapter name identifies a network application that is written for the NETBIOS emulator interface. The application resides on a remote personal computer.

The SET command for adapter name information is case sensitive. This means that you must specify the name to set by enclosing the name in quotation marks if the name contains any lowercase letters. If you have two remote adapter names that use the same letters but are in different cases, such as "BEST" and "best", you must be very specific when using the SET REMOTE-ADAPTER-NAME command.

Format

```
                                                             ┌USER user-id  ┐
SET REMOTE-ADAPTER-NAME nam [BYTE16 n] NODE node [OBJECT num]│PASSWORD pswd │
                                                             └ACCOUNT acct  ┘
```

or

```
                                                      ┌USER user-id  ┐
SET REMOTE-ADAPTER-NAME/HEX hh NODE node [OBJECT num] │PASSWORD pswd │
                                                      └ACCOUNT acct  ┘
```

Where:

nam Represents a 1- to 16-character alphabetic string that specifies the name to assign to the remote application. To store the characters representing the name in uppercase letters, type the characters. If you do not want to store the characters as uppercase, place the string within quotation marks.

node Identifies the node associated with this remote adapter. The node you use must be already defined in the node database, DECNODE.DAT. You must identify a node for each remote adapter name that you set. The node ID can be either the node's name or its address. A node name is a 1- to 6-character alphanumeric string that identifies a specific node. A node address is a numeric character string consisting of an area number and a specific node number. The node address also serves as a unique identification of the node in the network.

num Represents the number of the DECnet object. The range for this number is 0 to 255. The default is 0.

user-id Represents the user information you can use for accessing the loopback mirror on the remote node. A user ID consists of 1 to 39 alphabetic characters.

pswd Specifies a unique character string that you use for accessing the loopback mirror on the remote node. A user's password consists of 1 to 39 alphanumeric characters (some systems restrict password lengths to 8 characters).

acct Represents alphanumeric information for accessing the adjacent node's loopback mirror. The information can consist of 1 to 39 alphanumeric characters.

n Specifies a value for byte 16. The 16th character can be a non-printable ASCII character, entered in decimal. This value can be any decimal value from 0 to 255. The default is 20.

hh Specifies a number representing the adapter name in hexadecimal up to 16 pairs of hex digits. The first byte cannot be the hexadecimal value of 0 or 2a.

Note

Do not leave a space between the SET REMOTE-ADAPTER-NAME command and the /HEX parameter.

To store the USER, PASSWORD, or ACCOUNT information in uppercase letters, type the characters. For example:

Abc is stored in the database as **ABC**.

If you do not want to store the characters as uppercase, place the string within quotation marks. For example:

"Abc" is stored in the database as **Abc**.

Example

1. The following command enters the REMOTE-ADAPTER-NAME NORMAL in the volatile database:

   ```
   NCP>SET REMOTE-ADAPTER-NAME NORMAL [Return]
   ```

2. The following commands define a remote adapter name in the volatile database. These two commands do the same thing.

   ```
   NCP>SET REMOTE-ADAPTER-NAME STEVEN BYTE16 11 NODE LUVLIT
   ```

   or

   ```
   NCP>SET REM/HEX 53-41-49-4E-54-53-54-45-56-45-4E-20-20-20-20-0B NODE LUVLIT
   ```

SHOW ACTIVE

The SHOW ACTIVE command displays summary type information for all active links between the executor node and the adjacent node, as well as information about the executor node, the adjacent node, and the remote nodes that currently have active links to your local node.

Format

SHOW ACTIVE LINKS [TO *file-id*]

or

SHOW ACTIVE NODES [TO *file-id*]

Where:

TO *file-id* Specifies the name of an output file to direct information to.

Examples

1. The following command lists all the active logical links to your node:

```
NCP>SHOW ACTIVE LINKS Return
Active Links Status as of 13-Aug-1991 9:42:47

State       Socket  Node    Local   Remote  Local           Remote
                            Addr    Addr    #   Name        #   Name

Running     1       55.124  35392   17455   0    LLA35392   64
Running     2       4.30    35425   52272   0    LLA35425   64
NCP>
```

2. This command lists the active nodes that have logical links to your node:

```
NCP>SHOW ACTIVE NODES Return
Active Volatile Nodes as of 13-Aug-1991 9:42:58

Node       Node    Active   Delay     Next Node
Address    Name    Links              Address    Name

55.124     BLDJ    1
4.30       ORIOLE  1
NCP>
```

SHOW ADJACENT NODE

On asynchronous nodes, the SHOW ADJACENT command displays information for the adjacent node. On Ethernet nodes, this command displays the designated router.

Format

SHOW ADJACENT NODE [TO *file-id*]

Where:

TO *file-id* — Specifies the name of an output file to which information is to be directed.

Example

This command displays the information for the adjacent node:

```
NCP> SHOW ADJACENT NODE Return
```

Here is a sample system response:

```
Known Volatile Nodes as of 21-Aug-1991 16:15:47

Node      Node    State       Active   Circuit
Address   Name                Links    ID

55.261    DSRVC1  Reachable   0        ETHER-1

NCP>
```

SHOW CIRCUIT

The SHOW CIRCUIT command displays all the values of the running system. You can either display the information on your screen or redirect it to an output file at the local node.

Circuit counters monitor traffic and errors on individual circuits. The network coordinator can periodically check each circuit's counters to assess circuit performance and determine potential problems.

Format

```
                          ┌ CHARACTERISTICS ┐
SHOW  CIRCUIT [circuit-id] │ COUNTERS        │ [TO file-id]
                          │ STATUS          │
                          └ SUMMARY         ┘
```

Where:

CIRCUIT [*circuit-id*]	Specifies the circuit (ASYNC-1, ETHER-1, or TOKEN-1) for which information is to be displayed. Because one circuit is used for each configuration, the *circuit-id* is optional.
CHARACTERISTICS	Displays parameters set for the circuit.
COUNTERS	Displays error and traffic information for the circuit.
STATUS	Displays the availability of the circuit for network activity.
SUMMARY	Displays a summary of information for the circuit.
TO *file-id*	Specifies the name of an output file to direct information to.

Examples

1. The following command displays the current characteristics of circuit ASYNC-1:

   ```
   NCP>SHOW CIRCUIT ASYNC-1 CHARACTERISTICS [Return]
   ```

2. The following command displays the current counters for the circuit:

   ```
   NCP>SHOW CIRCUIT COUNTERS [Return]
   ```

3. The following command redirects the current status information about circuit ETHER-1 to an output file named CIR.DAT:

```
NCP>SHOW CIRCUIT ETHER-1 STATUS TO CIR.DAT Return

Circuit counters as of 22-Mar-1991 15:20:23

Circuit = ETHER-1

Seconds since last zeroed                    = 1290
Terminating packets received                 = 4
Originating packets sent                     = 0
Circuit down                                 = 0
Initialization failure                       = 0
Bytes received                               = 32020
Bytes sent                                   = 1452
Data blocks received                         = 187
Data blocks sent                             = 44
User buffers unavailable                     = 0

NCP>
```

SHOW EXECUTOR

The SHOW EXECUTOR command displays the current executor node information. You can either display the information on your screen or redirect it to an output file at the local node.

Format

```
                ┌ CHARACTERISTICS ┐
                │ COUNTERS        │
SHOW  EXECUTOR  │ STATUS          │  [TO file-id]
                └ SUMMARY         ┘
```

Where:

CHARACTERISTICS Displays parameters set for the executor.

COUNTERS Displays error and traffic information for the executor.

STATUS Displays the availability of the executor, for network activity.

SUMMARY Displays a summary of information for the executor.

TO *file-id* Specifies the name of an output file to direct information to.

Examples

1. The following command redirects the status information for the executor node to an output file named STAT.NOD:

   ```
   NCP>SHOW EXECUTOR STATUS TO STAT.NOD Return
   ```

   To display this file, you must exit NCP by pressing Ctrl/Z Return, and then enter the TYPE command:

   ```
   C:\>TYPE STAT.NOD Return

   Executor Status as of 9-Mar-1991 9:47:19

   Executor node     = 2.975 (ROCKY)

   State     = On
   Executor identification   = PATHWORKS for DOS V4.1 client.

   NCP>
   ```

2. The following command displays the characteristics for the executor node:

   ```
   NCP>SHOW EXECUTOR CHARACTERISTICS Return
   ```

SHOW LINE

The SHOW LINE command displays line information. You can either display the information on your screen or redirect it to an output file at the local node.

Line counters monitor procedures on individual lines. The network coordinator can periodically check each line's counters to assess line performance and determine potential problems.

Format

```
                      ┌ CHARACTERISTICS ┐
SHOW  LINE [line-id]  │ COUNTERS        │  [TO file-id]
                      │ STATUS          │
                      └ SUMMARY         ┘
```

Where:

LINE *line-id* — Specifies either (ASYNC-1, ETHER-1, or TOKEN-1) as the line for which information is to be displayed. Because there is one line for each configuration, the *line-id* is optional.

CHARACTERISTICS — Displays parameters set for the line.

COUNTERS — Displays error and traffic information for the line.

STATUS — Displays the availability of the line for network activity.

SUMMARY — Displays a summary of information for the line.

TO *file-id* — Specifies the name of an output file to direct information to.

Example

The following command redirects counter information about line ETHER-1 to an output file named STORE.LIN:

```
NCP>SHOW LINE ETHER-1 CHARACTERISTICS TO STORE.LIN [Return]
```

If you enter the TYPE command to display its contents, you see the following:

```
Line Characteristics as of 22-Mar-1991 15:26:41

Line = ETHER-1

Line state                              = ON
Line substate                           = Running
Device Id                               = 3CM-1
Receive buffers                         = 8
Protocol type                           = Ethernet
Hardware address                        = 08-00-2B-09-9C-89
Station Address                         = AA-00-04-00-E8-13

NCP>
```

SHOW LOCAL-ADAPTER-NAME

The SHOW LOCAL-ADAPTER-NAME command displays information about the local adapter name in the volatile database.

The local adapter name identifies a network application that is written for the NETBIOS emulator interface. The application resides on your local personal computer.

The SHOW command for adapter name information is case sensitive. This means that you must specify the name to display by enclosing the name in quotation marks if the name contains any lowercase letters. If you have two local adapter names that use the same letters but are in different cases, such as "BEST" and "best", you must be very specific when using the SHOW LOCAL-ADAPTER-NAME command. If you do not use quotation marks with the SHOW command, NCP displays the name stored in all capital letters.

Format

SHOW LOCAL-ADAPTER-NAME *name* [**BYT**E16 *nn*]

or

SHOW LOCAL-ADAPTER-NAME/HEX *hh*

or

SHOW KNOWN LOCAL-ADAPTER-NAMES

Where:

name Represents a 1- to 16-character ASCII string that specifies the local application for which to display information.

nn Specifies a value for byte 16. The 16th character can be a non-printable ASCII character, entered in decimal. This value can be any decimal value from 0 to 255. The default is 20.

hh Specifies a number representing the adapter name in hexadecimal up to 16 pairs of hex digits. The first byte cannot be the hexadecimal value of 0 or 2a.

Note

Do not leave a space between the SHOW LOCAL-ADAPTER-NAME command and the /HEX parameter.

Example

1. The following command displays information about the local adapter named REGAL:

```
NCP>SHOW LOCAL-ADAPTER-NAME REGAL Return
```

2. The following commands display the local adapter names. These two commands do the same thing:

```
NCP>SHOW LOCAL-ADAPTER-NAME OTHERONE BYTE16 1
```

or

```
NCP>SHO LOC/HEX 4F-54-48-45-52-4F-4E-45-20-20-20-20-20-20-20-01
```

The following display appears as a result of the above commands:

```
Volatile Local-Adapter database as of 25-Sep-1991 15:44:29

"Local-Adapter  "  Num Type  Local-Adapter in hex

"ALTHEA         "  2         41 4C 54 48 45 41 20 20 20 20 20 20 20 20 20 20
"ALTHEA        #"  3         41 4C 54 48 45 41 20 20 20 20 20 20 20 20 20 00
"OTHERONE      #"  4         4F 54 48 45 52 4F 4E 45 20 20 20 20 20 20 20 01
```

SHOW NODE

The SHOW NODE command displays information about a specific node if you have made a DECnet connection since the last network restart.

The SHOW KNOWN NODES command displays information about all the nodes to which you have made DECnet connections since the last network restart. This command always displays the executor node and the adjacent node.

Format

SHOW NODE *node-id*

or

SHOW KNOWN NODES

Where:

node-id Specifies the name or the address of the node for which to display information.

Examples

1. The following command displays information for the node ORIOLE:

```
NCP>SHOW NODE ORIOLE Return

Known Volatile Nodes as of 17-Oct-1991  15:12:11

Node     Node     Active     Delay   Next Node
Address  Name     Links              Address      Name

 4.30    ORIOLE   0          3
NCP>
```

2. The following command displays information about nodes that have been accessed through PATHWORKS for DOS since network restart:

```
NCP>SHOW KNOWN NODES Return

Known Volatile Nodes as of 17-Oct-1991 15:20:33

Node     Node     Active     Delay    Next Node
Address  Name     Links               Address    Name

2.975    ROCKY
55.161
 4.298   DELON    0          1
NCP>
```

SHOW REMOTE-ADAPTER-NAME

The SHOW REMOTE-ADAPTER-NAME command displays information about the remote adapter name in the volatile database.

The remote adapter name identifies a network application that is written for the NETBIOS emulator interface. The application resides on a remote personal computer.

The SHOW command for adapter name information is case sensitive. This means that you must specify the name to display by enclosing the name in quotation marks if the name contains any lowercase letters. If you have two remote adapter names that use the same letters but are in different cases, such as "BEST" and "best", you must be very specific when using the SHOW REMOTE-ADAPTER-NAME command. If you do not use quotation marks with the SHOW command, NCP displays the name stored in all capital letters.

Format

SHOW REMOTE-ADAPTER-NAME *name* [**BYTE**16 *nn*]

or

SHOW REMOTE-ADAPTER-NAME/HEX *hh*

or

SHOW KNOWN REMOTE-ADAPTER-NAMES

Where:

name	Represents a 1- to 16-character alphabetic string that specifies the remote application for which to display information.
nn	Specifies a value for byte 16. The 16th character can be a non-printable ASCII character, entered in decimal. This value can be any decimal value from 0 to 255. The default is 20.
hh	Specifies a number representing the adapter name in hexadecimal up to 16 pairs of hex digits. The first byte cannot be the hexadecimal value of 0 or 2a.

Note

Do not leave a space between the SHOW REMOTE-ADAPTER-NAME command and the /HEX parameter.

Example

1. The following command displays information about the REMOTE-ADAPTER-NAME PERSIS in the volatile database:

   ```
       NCP>SHOW REMOTE-ADAPTER-NAME PERSIS Return
   ```

2. The following commands display the remote adapter definitions in the volatile database. These two commands do the same thing:

   ```
   NCP>SHOW REMOTE-ADAPTER-NAME SAINTSTEVEN BYTE16 11
   ```

   or

   ```
   NCP>SHO REM/HEX 53-41-49-4E-54-53-54-45-56-45-4E-20-20-20-20-0B
   ```

   The following display appears as a result of the above commands:

   ```
   Volatile Remote-Adapter database as of 25-Sep-1991 15:28:22

   Remote-Adapter   BYTE16   Node   Node     Obj  Access
                    in hex   Name   Address   #

   SAINTSTEVEN        0B     LUVLIT  1.2     0
                            "53 41 49 4E 54 53 54 45 56 45 4E 20 20 20 20 0B"
   ```

TELL

The TELL command instructs a remote DECnet node to display information about its executor, lines, nodes, or circuits.

If the remote system requires user and password information, these must be set using the DEFINE NODE command.

Format

```
                               ┌ CHARACTER ┐
TELL  node-id SHOW EXECUTOR    │ COUNTERS  │ [TO file-id]
                               │ STATUS    │
                               └ SUMMARY   ┘

                                      ┌ CHARACTER ┐
TELL  node-id SHOW CIRCUIT circuit-id │ COUNTERS  │ [TO file-id]
                                      │ STATUS    │
                                      └ SUMMARY   ┘

                                      ┌ CHARACTER ┐
TELL  node-id SHOW KNOWN CIRCUIT      │ COUNTERS  │ [TO file-id]
                                      │ STATUS    │
                                      └ SUMMARY   ┘

                               ┌ CHARACTER ┐
TELL  node-id SHOW LINE line-id│ COUNTERS  │ [TO file-id]
                               │ STATUS    │
                               └ SUMMARY   ┘

                                   ┌ CHARACTER ┐
TELL  node-id SHOW KNOWN LINES     │ COUNTERS  │ [TO file-id]
                                   │ STATUS    │
                                   └ SUMMARY   ┘

TELL  node-id SHOW KNOWN NODES [TO file-id]

TELL  node-id SHOW AREA number [TO file-id]
TELL  node-id SHOW KNOWN AREA [TO file-id]
```

```
                          ( node-id   )
TELL  node-id SHOW NODE  {  *.node-id  } [TO file-id]
                          ( area.*    )
```

or

```
                     ( CIRCUIT circuit-id )
TELL  node-id ZERO  {  EXECUTOR            } [COUNTER] [TO file-id]
                     ( LINE line-id       )
```

Where:

CIRCUIT *circuit-id* — Specifies the circuit on the remote node for which information is to be displayed.

LINE *line-id* — Specifies the line on the remote node for which information is to be displayed.

number — Specifies a certain area number for which you want to view node information. The value of *number* can be from 1 to 63. SHOW KNOWN AREA is for use with Level 2 routers.

*area.** — Specifies the particular node area for which to display information. The asterisk (*) is a wildcard that indicates all the node information for the specified area. (The use of wildcards works if the remote system supports the wildcard option for node names for network management.)

**.node-id* — Specifies a particular node id, regardless of area. Indicates that you want information on all nodes of that node-id.

CHARACTERISTICS — Displays parameters set for the executor, line, or circuit.

COUNTERS — Displays error and traffic information for the executor, line, or circuit.

STATUS — Displays the availability of the executor, line, or circuit for network activity.

SUMMARY — Displays a summary of information for the executor, line, circuit, or links.

TO *file-id*   Specifies the name of an output file to direct information to.

Example

The following command instructs the node RM5J to display information about its executor characteristics:

```
NCP>TELL RM5J SHOW EXECUTOR CHARACTERISTICS Return
```

Here is a sample system response:

```
Executor address                 = 4.30
Executor name                    = RM5J

Driver version number            = 4.0.0
State                            = ON
Executor Identification          = PATHWORKS for DOS V4.1 client.
Executor Version                 = 4 2 0
Incoming timer                   = 45
Outgoing timer                   = 60
Confidence timer                 = 15
Incoming proxy                   = 1
Outgoing proxy                   = 0
Active links                     = 3
NSP version                      = 4 0 0
Maximum links                    = 16
Delay factor                     = 32
Delay weight                     = 3
Inactivity timer                 = 60
Retransmit factor                = 12
Routing version                  = 2 0 0
Routing types                    = Endnode IV
Maximum buffers                  = 127
Buffer size                      = 576
Segment Buffer Size              = 557
Transmit password                =
Receive password                 =
Database Path                    = C:\DECnet\
Transmit pipe quota              = 6
Receive pipe quota               = 6
PC type                          = 7
Nak quota                        = 0
Autoboot                         = 0
NCP>
```

ZERO CIRCUIT

The ZERO CIRCUIT command zeroes the counters for the circuit.

Format

ZERO CIRCUIT *[circuit-id]* **[COUNTERS]**

Where:

CIRCUIT [*circuit-id*] Specifies either (ASYNC-1, ETHER-1, or TOKEN-1) as the circuit for which counters are to be zeroed.

COUNTERS Sets the circuit counters to zero.

Example

The following command zeroes the counters for circuit ETHER-1:

```
NCP>ZERO CIRCUIT ETHER-1 Return
```

ZERO EXECUTOR

The ZERO EXECUTOR command zeroes the counters associated with and maintained on the executor node.

Format

ZERO EXECUTOR [**COU**NTERS]

Where:

COUNTERS Sets the executor node's counters to zero.

Example

The following command zeroes the counters associated with the executor node:

```
NCP>ZERO EXECUTOR COUNTERS Return
```

ZERO LINE

The ZERO LINE command zeroes the counters for the line.

Format

ZERO LINE [*line-id*] [**COUNTERS**]

Where:

LINE [*line-id*]	Specifies either (ASYNC-1, ETHER-1, or TOKEN-1) as the line for which counters are to be zeroed. Because there is one line for each configuration, the *line-id* is optional.
COUNTERS	Sets the line counters to zero.

Example

The following command zeroes the counters for line ETHER-1:

```
NCP>ZERO LINE ETHER-1 Return
```

A

Network Control Program Error Messages

A.1 Introduction

Some common Network Control Program (NCP) error messages are listed on the following pages. An explanation and the appropriate user response are also included with each error message.

A.2 Types of Network Control Program Error Messages

There are three types of NCP error messages:

- **Command line errors.** When you use the incorrect syntax for a command line, the system displays these messages.
- **File input/output (I/O) errors.** When you try to access MS–DOS files, the system displays these messages.
- **Network errors**. When you try to run NCP on a system where network support, or some part of it, is missing or not working properly, the system displays these messages.

A.3 Command Line Errors

The messages listed in this section are all related to errors in command line entry.

Cannot set DDCMP parameter in Ethernet configurations.

Explanation: You are running DECnet over an Ethernet connection, and you tried to set a parameter used only by DDCMP asynchronous lines.

User Action: Use a different parameter, or reconfigure your connection for Ethernet communication.

Cannot set Ethernet parameter in DDCMP configurations.

Explanation: You are running DECnet on an asynchronous line, and you tried to set a parameter used by Ethernet.

User Action: Use a different parameter, or reconfigure your connection for DDCMP communications.

Could not find node address with node name = *node-name.*

Explanation: The node address and node name pair did not match.

User Action: Check your list of node names and node addresses and attempt to execute the NCP command again.

Could not find node name with node address = *node-address.*

Explanation: NCP was unable to find the node name defined by the specific node address.

User Action: Check your list of node names and node addresses and try to execute the NCP command again.

Executor is in wrong state.

Explanation: The executor is not in the proper state for changing the parameter you specified. (For example, you cannot set the EXECUTOR NAME or EXECUTOR ADDRESS when the state is ON.)

User Action: Use the DEFINE EXECUTOR command rather than SET EXECUTOR.

Line is in wrong state.

Explanation: The line is not in the proper state for changing the parameter you specified. (For example, you cannot change the line speed while the line state is ON.)

User Action: Change the line state first, then change the parameter.

Node address must be in the range 1 to 1023.

Explanation: The node address has exceeded the valid range.

User Action: Enter a node address that is in the range of 1 to 1023.

Node area must be in the range 1 to 63.

Explanation: The area number has exceeded the valid range.

User Action: Enter a node area that is in the range of 1 to 63.

Node name already defined for another node number.

Explanation: You tried to define a node name that has already been matched to a different node number.

User Action: Select another name to apply to this node number.

Node name must contain an alphabetic character.

Explanation: The node name you defined did not contain any alphabetic characters. Node names can consist of 1 to 6 alphanumeric characters, but at least one character must be a letter.

User Action: Redefine the node name using at least one letter.

Node number already defined for another node name.

Explanation: You tried to define a node number that has already been matched to a different node name.

User Action: Select another number to apply to this node name.

Value out of range; check the documentation or type HELP.

Explanation: The value you selected for this parameter is not within the valid range.

User Action: The values for NCP parameters are defined in Chapter 2. Typing HELP will also provide the valid ranges for these parameters. Check the valid range for the parameter you want to define, then enter a correct value.

String too long. Maximum length is *n*.

Explanation: The string you entered has too many characters in it.

User Action: Enter a string that is in the range specified in the message (this message can appear for several different commands).

^ User command error

Explanation: The command line contains an invalid or missing keyword or parameter. The system displays an incorrect command line with this error message on the next line. The circumflex (^) points to the portion of the command line that was incorrect.

User Action: Use the NCP HELP command to request on-line help for a specific command.

A.4 File Input/Output Errors

The messages listed in this section are all related to accessing MS–DOS files.

Could not open file: {*file-name*}

Explanation: The file was not found on the default database device in the \DECNET directory or an MS–DOS error occurred while you attempted to create or open the file.

User Action: Refer to the appropriate installation guide for details on the \DECNET directory.

Could not read from file: {*file-name*}

Explanation: An error occurred while you attempted to read from the file. The file may not have been properly created/opened, an attempt to read past the end of the file may have occurred, or the file may not have been found in the default database device in the \DECNET directory.

User Action: Contact the person responsible for your network for assistance.

Could not seek in file: {*file-name*}

Explanation: An error occurred while you attempted to modify the current seek position in the file.

User Action: Contact the person responsible for your network for assistance.

Could not write to file: {*file-name*}

Explanation: An error occurred while you attempted to write data to the file. The file may not have been properly created/opened, or the disk may be full.

User Action: Contact the person responsible for your network for assistance.

A.5 Network Error Messages

You may see the following messages when you try to run NCP on a system where network support (or some part of it) is missing or not working properly.

Network Error: Can't assign requested address

Explanation: A request to set up a node name and node address from a remote node has failed. The node name and node address supplied by the user was not recognized by the local or remote node.

User Action: You should specify a different node name and node address. Retry the operation.

Network Error: Connection timed out

Explanation: A request to set up a node name and node address from a remote node has failed. The time period that is allowed for connecting to the remote Network Management Listener has expired.

User Action: Set the executor's timeout parameters to longer periods of time. Try the command again.

Network Error: Executor is in wrong state

Explanation: An attempt to set a specific network parameter has failed. This occurs when you try to set the executor's node name or node address and the executor's state is already ON.

User Action: Delete the DECPARM.DAT file in your \DECNET directory. Run the SETUP program and @NCP.TXT file as detailed in the appropriate installation guide.

Network Error: Host is unreachable

Explanation: The remote node is not reachable. NCP cannot grant the request to set up a node name and node address from a remote node.

User Action: Try a different remote node.

Network Error: Line is in wrong state

Explanation: An attempt has been made to set certain line parameters when the line state is already ON.

User Action: You must first set the line state to OFF. Then use the SET LINE command to modify certain line parameters. Then turn the line state to ON again.

Network Error: Network is down

Explanation: An attempt was made to access the network with the executor state OFF.

User Action: Verify that the executor's node address (and name) has been set. Set the executor state to ON with the SET EXECUTOR command, and turn the line state to ON with the SET LINE command.

Network Error: Network is unreachable

Explanation: The DECnet driver was not properly installed.

User Action: Refer to the appropriate installation guide for installation instructions. Repeat the installation procedure.

Network Error: No response from object

Explanation: The logical link connection is broken, or the remote network management listener has failed. The NCP utility cannot grant the request to set up a node name and node address from a remote node.

User Action: Try to execute your network request again. You can increase the executor's timeout period or try to make a connection to a different remote node.

Network Error: No such process

Explanation: The remote node has no network management listener object. The NCP utility cannot grant a request to set up a node name and node address from a remote node.

User Action: Try another remote node.

Network Error: Permission denied

Explanation: An attempt was made to turn the executor state to ON when the executor's address was not set. An attempt was made to set the line state to ON when the executor's state is OFF.

Either an attempt was made to unload the network when DNP was in EMS, or a TSR was loaded into conventional memory after DNP and the unloading of DNP would leave a memory hole.

User Action: Use the SET EXECUTOR command to set the executor's node address, then turn the executor state ON with the SET EXECUTOR command.

Explanation: turn the line ON, first set the executor's state to ON. Then set the line state to ON with the SET LINE command.

Network Error: Socket operation on non-socket

Explanation: The specified socket number was invalid.

User Action: Specify a valid socket number using the SET LINK command. Refer to the PATHWORKS for DOS, *DECnet Programmer's Reference Manual* for more details.

A.6 Loop Test Error Messages

When you test the network using loop commands, you may see the following error messages in addition to other network error messages.

Please run NCP to SET NODE *n* NAME.

Explanation: There is no node name defined for performing the loop tests.

User Action: Be sure you have defined the proper node name to use for performing loop tests. You can define the name using the NCP command SET NODE *n* NAME.

Cannot access information about executor node. Please check that the DECnet network driver is installed and that NCP runs to:

SET EXECUTOR ADDRESS *node-address* NAME *node-name*

SET EXECUTOR STATE ON

Explanation: You have not defined the executor node name and address.

User Action: Make sure DECnet is installed and that you can run NCP. Then use the NCP commands to define the executor node name and address and to turn the executor state ON.

B
Maintenance Operations Protocol (MOP) Load & Dump Programs

B.1 MOP for DOS

MOP and MOPCONF are programs which facilitate the loading and dumping of devices such as DECservers and DECrouters. The programs are part of the PATHWORKS for DOS Client V4.1 kit. They are dedicated programs which will reserve the personal computer (or workstation) for their own use. If MS-Windows is running and MOP.EXE (not MOPCONF.EXE) is executing, window switching is disabled.

MOPCONF.EXE allows the user to populate a dump/load database (ROUCONFG.DAT) which contains load and dump information for certain nodes. There is also a default file, MOPCONF.DEF, which contains default values for various device types such as a DECserver 200 (DS200) or a DECrouter 250 (DR250). You can add a device type to this text file, but there is no guarantee that MOP will be able to load this new device type. (The file contains comments that explain the proper format to use.)

MOP.EXE is a load/dump server. It also lets you trigger a reboot of a specific node. MOP.EXE functions mostly as a server that waits for load or dump requests. Based on the information found in the dump/load database, MOP can either start the load/dump operation or ignore the request.

The following files are assumed to be in the default DECnet directory:

- System Load
- Dump
- MOPCONF.DEF
- ROUCONFG.DAT

You can override both system load and dump file locations during a node addition or node modification by using the MOPCONF program. The other files (MOP.EXE and MOPCONF.EXE) must be accessible using the Path environment variable.

B.1.1 MOPCONF.EXE

MOPCONF.EXE lets you manipulate information in the dump/load database (ROUCONFG.DAT) in four ways:

- You can list all known nodes (and their information)
- You can add a node
- You can change information for a particular node
- You can delete a node

When MOPCONF.EXE starts, it first attempts to find the default file (MOPCONF.DEF) in the default DECnet directory. This file contains a number of device types with their default load and dump file names. These files are also assumed to be in the default DECnet directory. This information is then read in and stored internally.

MOPCONF then lists your options. If you choose List, MOPCONF opens the ROUCONFG.DAT database and displays information on the screen.

Note

Since there is no screen scroll support, information can be lost.

If you choose to add a node, MOPCONF displays the default device types and asks you to choose one from the list or gives you the option of indicating a different device type. MOPCONF then requests the:

- DECnet node name
- DECnet node address
- Ethernet hardware address.

MOPCONF then asks you for the load file name and the dump file name. If you choose a default device type, MOPCONF provides a default file name. However, you do have the option of overriding that file name by providing either a DOS file name (8-character maximum with a 3-character maximum extension) or a full file specification including the device drive and directories. If you provide only a simple file name, the DECnet default directory is assumed. Remember, the default DECnet directory is the location where the following files are stored:

- DECPARM.DAT
- DLLNDIS.EXE
- PROTOCOL.INI

For some systems (such as DECrouters), a template is used for the load and dump file names. The name must be unique for a load file and is typically in the following format:

R2*xxxxxx*.SYS (where *xxxxxx* is the node name)

MOPCONF does not provide the actual system files. You must provide these files. MOPCONF provides only storage of load/dump information.

You can modify information in the database. You can modify the device type, the DECnet name, the DECnet address, the hardware address, and the load and dump files.

You also have the option to delete a node from the database. Currently, MOPCONF copies all the information from the database to a temporary file (excluding the information for the deleted node). Once MOPCONF copies all the information, it deletes the old ROUCONFG.DAT and renames the temporary file to ROUCONFG.DAT.

B.1.2 MOP.EXE

MOP is an executable program which acts as a load/dump server and can be used to trigger a reboot of a node. If you use the trigger option, MOP sends a reboot message to the specified node before it begins to act as a server.

MOP handles both downline loads or upline dumps. It relies on the information found in the ROUCONFG.DAT database to determine if it can load/dump a particular node.

MOP operates in a mode of waiting for a load or dump request to come in. Based on the information in the actual request, MOP looks for node information using either the node's hardware address or the software ID. (There are values in the load or dump request message to determine if the software ID is to be used.)

If the load is using a software ID, the file name becomes the software ID and MOP looks for that file in the default DECnet directory. If the software ID is larger than 8 characters, then MOP depends upon DOS to truncate it.

If the load is using the hardware address, MOP locates the file name by looking in the ROUCONFG.DAT database for a match on the hardware address. If MOP finds a match, then it uses the load file for that node. Once MOP locates a valid file, it then transfers the file to the requesting system.

A dump request works the same way, except that MOP receives the dump file and places it in the file that was specified in the ROUCONFG.DAT database.

For example, MOP uses the hardware address for a DECrouter 200 to find its node entry in the ROUCONFG.DAT database because the load request stated that the hardware address was to be used as a key. The DECserver 200 is different in that its load request contains a software ID and the request states to use this ID in finding the load file. MOP looks for a file in the default DECnet directory having the same name as the software ID. MOP does not use the ROUCONFG.DAT database to access information for this down-line load operation.

If you request a reboot of a node with the MOP TRIGGER command, MOP looks for load information in the ROUCONFG.DAT database. Even though certain devices load using their software IDs, MOP does not know which devices do so. Hence, all nodes that are triggered through MOP must have an entry in the ROUCONFG.DAT database.

Two boot messages are sent to the node. One is sent to the station address (an Ethernet address which is only recognized if DECnet is up and running on the node) and one is sent to the hardware address (the original Ethernet address of the communication device, used prior to DECnet being started).

MOP sends the reboot message to these two addresses since it is unsure of the state of the device and requires one of the messages to get to the device.

B.2 USER INTERFACE

B.2.1 MOPCONF.EXE

The MOPCONF.EXE program implements a simple screen format in which the first screen offers you a menu of options. The options are as follows:

1. List known nodes
2. Add a node
3. Modify an existing node
4. Delete an existing node
5. Exit from this procedure

When you start MOPCONF.EXE, you see the following information:

```
 1 - List known nodes
 2 - Add a node
 3 - Modify an existing node
 4 - Delete an existing node
 5 - Exit from this procedure

Your selection?  ___
```

Select an option by entering its corresponding number, then press Return.

The following table lists and describes the available options.

Option	Description
1 - List known nodes	Presents you with a list of all nodes defined in the ROUCONFG.DAT database (in the default DECnet directory). Each node has the following information associated with it: • DECnet address • DECnet name • Device type • Service Circuit • Ethernet (hardware) address • Load file name • Dump file name (the dump file name is just below the load file name). A loss of node information may occur due to scrolling beyond a screen size.

Option	Description
2 - Add a node	Presents you with a list of default device types (or none if there is no default MOPCONF.DEF file). You can choose one of the default device types by entering its corresponding number, or you can choose OTHER, which indicates a different device type. If you choose OTHER, you will be prompted for the device type. In either case, MOPCONF next prompts you for the following information: • DECnet node name • DECnet node address • Ethernet (hardware) address • Load file name • Dump file name If you select one of the default device types, MOPCONF presents you with default load and dump file names. You can then choose either: • The default name (the default file name is selected by just pressing Return) • A full file name (by entering the disk drive, directory, and file name) • A simple file name (by entering no disk drive or directory, although MOPCONF assumes the default is the DECnet directory)

Option	Description
3 - Modify an existing node	Prompts you for the DECnet node name. When you enter a valid node name (a name already in the ROUCONFG.DAT database), you are prompted for the following information: • Device type • DECnet node name • DECnet node address • Ethernet (hardware) address • Load file name • Dump file name
4 - Purge an existing node	Prompts you for the DECnet node name. When you enter a valid node name (a name already in the ROUCONFG.DAT database), MOPCONF deletes that node and its associated dump and load information from the database.
5 - Exit from this procedure	MOPCONF.EXE terminates and returns you to the DOS prompt.

Note

By pressing Return for each of the above requests, you will not change the current values.

Important

Do not issue a Ctrl/C during the execution of MOPCONF.EXE. This could leave files (especially ROUCONFG.DAT) in an opened state.

B.2.2 MOP.EXE

There is a very simple interface into MOP. To start the LOAD/DUMP server, you can use MOP with either the W option or no option at all.

```
Prompt> MOP
```

or

```
Prompt> MOP W
```

To issue a TRIGGER command, (which would be issued prior to the server starting), use the following format:

Prompt> MOP T *node-id* {*password*}

Where :

node-id Is one of the following:

1. DECnet node name
2. DECnet node address
3. Ethernet hardware address

password Is optional

If a load or dump request is received, MOP will print out various information for you. If the request was for a load, you will see information in the following format:

```
Program Load Request from : hardware-address at DATE TIME
Program Type : system
Software ID : standard operating system
```

Where:

system Is either the system, secondary, or tertiary load file that is requested

standard operating system Can be a valid software ID consisting of a character string

This information is followed by 3 possible strings:

1. If the load is successful, you see this message:

   ```
   Load complete at DATE TIME
   ```

2. If the load request is for a standard operating system (using the hardware address to find the node entry) but the node entry is not in ROUCONFG.DAT, you see this message:

   ```
   ?System load file not defined in nodes database
   ```

3. If the load request is for a standard operating system using the software ID to find the file (for example, JASPER) but the file was not found, you see the following message:

   ```
   ?Load file A:\DECNET\JASPER not found
   ```

   This example assumes that A:\DECNET is the default DECnet directory.

If you request an upline dump, you see the following message:

```
Up Line Dump request from: hardware address at DATE TIME
performing upline dump
```

There is the potential for a file write error due to the dump file being too large for the disk. In the case of any file write error, you see the following message:

```
?Dump file write error
```

Note

You can exit from MOP by using the ESC key or the Ctrl/C key sequence.

B.3 RESTRICTIONS/WARNINGS

B.3.1 MOPCONF.EXE

- You cannot use Ctrl/C to exit this program. Such action could leave the ROUCONFG.DAT and ROUCONFG.DEF files in an indeterminate state.
- Entering a series of blanks for a file name will not trigger an error.
- When purging an existing node (option 4), MOP creates a temporary file for copying all the data from the ROUCONFG.DAT database, excluding that data belonging to the purged node. When MOP copies all the information, it deletes the old ROUCONFG.DAT and renames the temporary file to ROUCONFG.DAT. (Make sure you have sufficient space in the default DECnet directory for this temporary file.)
- If you choose List Known Nodes, there is a potential for information loss due to screen scrolling.

- In order for the MOPCONF program to utilize the default file (MOPCONF.DEF), the default file must be physically moved to the default DECnet directory from its current location. The default DECnet directory is that directory which is pointed to by the PCSA environment variable (a user can issue a DOS SET command to find out the current value of the PCSA variable).

B.3.2 MOP.EXE

- When you exit from a previous MOP session, MOP occasionally does not receive load/dump requests upon starting a new session. If this occurs, it takes a couple of minutes for the condition to clear up. Allow the MOP server to run and it will soon begin to receive load/dump requests.
- If you run MOP in the DOS box for Windows, the load/dump information does not display. You will see only the MOP banner.

B.3.3 Default Device Types (Defined in MOPCONF.DEF)

The following device types are listed in the default file for the MOPCONF program:

DS200 - DECserver 200
DS300 - DECserver 300
DR200 - DECrouter 200
DR250 - DECrouter 250
DR2000 - DECrouter 2000

MOP and MOPCONF programs have been tested against these devices to make sure you can load and dump them through MOP.

You can add device types to the default file. However, the load and dump functionality of MOP is not guaranteed for the device types you add.

MOPCONF.DEF is a text file which you can edit in order to add or delete nodes from the default file. The comment header at the beginning of the file describes the format you should use for adding or deleting nodes.

For information about loading a specific device, refer to the management guide for that device and other related documentation.

The DECrouter 2000 differs from the other four devices in that it makes two load requests: one for the system file using the information in the ROUCONFG.DATA database; the other for the configuration file. The configuration file must be in the default DECnet directory and is named ROU*xxxxx*, where *xxxxx* is the DECnet node name (truncated if necessary). The MOPCONF program does not place the configuration file information into ROUCONFG.DAT.

Glossary

ASCII

An acronym for American Standard Code for Information Interchange. An 8-bit standard code adopted to facilitate the interchange of data between various types of data processing and data communications equipment.

access-control information

Optional security information that you might need in order to access a remote node. See also **default access-control information**.

account

A character string consisting of 1 to 39 alphanumeric characters.

adjacent node

A node in which you are physically connected by a single line.

alphanumeric

A character string that contains alphabetic or numeric characters, or a combination of both. The term *alphanumeric* is a contraction of the words *alpha*betic and *numeric*.

application

A program (other than the operating system) that performs specific functions to meet user requirements. Applications available with DECnet–DOS include the Network File Transfer (NFT) utility, the Network Device Utility (NDU), the Network Control Program (NCP), and SETHOST utility with network virtual terminal services.

asynchronous mode

A data transmission method that sends one character at a time. Also refers to commands (as in a windowing environment), that may be sent without waiting for a response from the previous command. See also **synchronous mode**.

asynchronous device

A device that transmits signals at irregular intervals to the system with which it is communicating.

binary

The number system with a radix of 2, or a property involving a condition of exactly 2 possibilities.

blocking I/O

A data-accessing technique in which the program manipulates the blocks (physical records) that make up a file, instead of its logical records; allows for the direct access to the blocks in a file without regard for the file organization or record format. See also **non-blocking I/O**.

blocking synchronous mode

When a process/application issues a call in blocking synchronous mode, the application thread blocks, or waits, until the network process has completed the call and returns status to the application. See also **non-blocking mode**.

buffer

A temporary storage area in a node's memory. Buffers temporarily hold data being transferred to and from the node. The size and the number of buffers determine the amount of data that a buffer can store.

circuit

The communications path that operates over a physical line connecting two nodes.

command

An instruction typed in at a terminal. A command requests the software monitoring a terminal or reading a command procedure to perform some predefined operation.

command file

A file containing commands and data that the command interpreter can accept in lieu of the user's typing the commands individually on a terminal.

command switch

A word or character string that modifies the way a command operates.

communication path

The route a message takes, through various hardware devices, when it is sent from one node to another.

control character

A character whose occurrence in a particular context initiates, modifies, or stops a function.

counter

A register or storage location that displays statistics about the flow of network messages for your node. Counters record error conditions and accumulate their totals.

database

A collection of interrelated data on one or more mass storage devices. The collection is organized to facilitate efficient and accurate inquiry and update.

datagram

An unacknowledged packet of data in a network as opposed to packets that require acknowledgment of receipt (also known as reliable communications).

data link

Equipment for the transmission of information in data format.

data type

An interpretation applied to a string of bits such as integers, reals, or characters.

DEC Multiport Repeater (DEMPR)

A piece of Digital networking hardware that can connect up to eight ThinWire Ethernet segments and optionally connect them to a backbone cable, or DELNI. See also **DELNI**.

DECnet

A family of Digital software products that extend the capabilities of Digital computers to perform various operations over a network. DECnet provides many standard applications for accessing files and databases that exist on other nodes.

DECnet–DOS

The version of DECnet for the personal computer or MS–DOS operating systems.

DECnet Logical Link

A cooperative venture between two tasks with an agreement to communicate. A logical link connection is required before data can be exchanged between two tasks.

default

An assumption made by a system or language translator when no specific choice is given by the program.

default access control information

The Network File Transfer (NFT) utility allows you to use security information, including user name, password, and account number. It uses this information you last specified for the node name. This is the default access-control information. See also **access control information**.

default directory

The directory in which the monitor searches when a user has not provided a directory specification.

default value

An assigned quantity for a device or program that is set by the manufacturer. A default value in a program is usually the most common or safest answer.

DELNI

See Digital Ethernet Local Network Interconnect.

demodulation

The process of reversing back to an original state, some feature of a signal.

Digital Ethernet Local Network Interconnect (DELNI)

An electronic device which permits workstations to exchange information over a 20 meter (65 foot) square area. You can connect up to eight workstations to one DELNI, or you can connect two DELNIs to each other. (One unit can connect to eight other DELNI LANS.)

device

A specific name for a disk or diskette that is currently storing data files.

device control unit

A hardware device that controls the reading, writing, or display of data at one or more input/output devices or terminals.

device drivers

Instructions the computer follows to reformat data for transfer to and from a particular peripheral device.

directory

A specific name assigned to a collection of files stored on disk or a diskette. A directory can provide a list of the file names that it contains.

end node

A DECnet term referring to a member of the network that can do everything except route packets through on behalf of other nodes. ULTRIX, MS–DOS, and third-party implementations of DECnet are all end nodes.

Ethernet

A communications concept for local communication networks that employ coaxial cable as a passive communications medium to interconnect computer systems.

executive

The controlling program or set of routines in an operating system.

Executor node

A node that runs Network Control Program (NCP) commands. For a personal computer, this is the local node.

FAL

See File Access Listener

File Access Listener (FAL)

A utility that runs as a background process to enable file transfers and accesses for utilities such as Network Device Utility (NDU) and Network File Transfer (NFT).

file specification

A complete file identification including an optional drive name and path name. The file specification must include at least a file name followed by an optional file type and version number.

foreign file

A file with a format not recognizable to DECnet–DOS or DECnet for OS/2. For the purpose of this documentation foreign file specifications are enclosed in quotation marks (" ") in a command line.

host node

A node on the network that your node can access for the purpose of sharing resources and information.

IEEE

Institute of Electronic and Electrical Engineers. A leading standard-setting group in the United States. IEEE 488 is a popular standard for real-time data collection. IEEE 802 is the standard for various types of local area networks.

I/O

Input/output. Term for transfer of data from main memory to either a disk drive, terminal, printer, or other device.

identifier

A user-supplied name of from 1 to 31 characters that denotes the name of a variable, statement label, entry, or file constant. The first character of an identifier must be a letter.

Job Spawner

Runs as a utility while awaiting incoming connection requests from other nodes in the network. The Job Spawner activates DECnet servers as background processes on your node (such as FAL). The Job Spawner also searches for connection request object names or numbers in its database, and then runs the program.

line

The physical line connecting the local node with the adjacent node.

link

To produce executable code from object code.

local area network (LAN)

A privately owned data communications system that offers high-speed communications channels optimized for connecting information processing equipment. The geographical area is usually limited to a section of a building, an entire building, or a group of buildings. Refers to Ethernet or token ring networks with DEC systems. See also **Ethernet**.

Local Area Transport (LAT)

A proprietary Digital architecture for terminal servers on Ethernet networks designed to conserve bandwidth and off-load processing from hosts.

local node

The node that you are currently using to communicate with other nodes.

logging

The process of recording information from an occurrence in the network. Logging is the process that generates a record of the event.

loop

To repeat the same sequence of instructions, or continuously repeat a computer operation.

loopback connector

A hardware device that temporarily connects the output portion of a circuit to the input portion of a circuit for the purpose of testing communication lines.

Loopback Mirror (MIRROR)

A utility on your node that allows any node on the DECnet network to send test messages to verify its network connection. The mirror runs as a background process to return those messages.

loopback test

A test that sends an electronic signal over a hardware or software connection to verify the operation of devices along the communication link. Certain loopback tests require a loopback connector. See also **loopback connector**.

mirroring

Pertains to the display or creation of graphic data that portrays an image in exactly the reverse orientation it originally had. Many computer graphics systems will automatically create a mirror image of a graphic entity.

modem

A hardware device capable of changing communications signals from digital to analog and back to digital. The term *modem* is a blend of the words *mod*ulation and *de*modulation.

modulation

The process of varying an original computer communication (digital) signal into a telephone communication (analog) signal. See also **demodulation**.

Microsoft Disk Operating System (MS–DOS)

The standard operating system used by the IBM Personal Computer and compatible computers. The manufacturer of MS–DOS is Microsoft Corporation.

MS–DOS

See Microsoft Disk Operating System.

network

A group of interconnected computers or systems that communicate with each other to share resources and information.

Network Control Program (NCP)

The block that contains the necessary information to set up a virtual circuit or to accept or reject a request to set up a virtual circuit. For DOS and OS/2, NCP allows you to perform Network Management tasks and to test the network hardware and software.

network coordinator

A person who assigns and updates node names and node addresses. The network coordinator provides administrative assistance to network users.

Network Device Utility (NDU)

A DECnet utility which defines virtual disk drives and virtual printers on remote systems. Capabilities include definition of files on remote systems as virtual disks and using them as if they were directly connected to your computer.

network driver

A set of instructions or software code that performs most of the functions that involve direct interfacing to the operating system. These functions include buffer management, interprocess communication management, and handling the interface between device drivers. The network driver and the network process appear as a single device driver from the user perspective.

Network File Transfer Utility (NFT)

A program that allows users to access or delete files residing on DECnet hosts that provide network file access capabilities. NFT initiates the service requests that will be carried by the FAL program. See also **FAL**.

network kernel

A minimum configuration or set of components to run DECnet software. It operates as a background task on a personnel computer.

network link

A temporary connection that establishes communication between programs running on different nodes. For example, the SETHOST utility with network virtual terminal services is a program that establishes a network link with another node, enabling you to log in to a remote host system.

network manager

A person who assigns and updates node names and node addresses. The network manager also provides administrative assistance to persons using the network.

Network Management Listener (NML)

A process that communicates with a network management interface on another node to provide information about the local node. Nodes in the network can send NCP commands to remote nodes that are running NML. NML executes those commands and sends the result(s) back to the source node.

network restart

A process that occurs when you reboot a running DECnet node or when you restart stopped network software.

node

Any terminal, station, or communications computer in a computer network. An individual computer system in a network that can communicate with other computer systems in the network.

node address

A unique numeric character string that identifies a node to other systems in the network.

node definition

A character string that identifies a particular node. The definition includes both the node address and the node name.

node name

A one to six character alphanumeric string (containing at least one letter) that identifies a node to other users in the network.

non-blocking mode

When a process/application issues a call in non-blocking mode, the application does not block while waiting for a buffer to be filled or emptied.

nonstream systems

In file specifications, file data is in specific record formats. Some examples of these formats are fixed length, variable, and variable with fixed length control (FLC). See also **stream systems**.

object

The basic named data definition stored in the CDD (Common Data Dictionary) that forms the end points of the CDD hierarchy branches. A system resource such as a file, device, or directory.

octal

A number system with a radix of 8. Octal numerals are frequently used to represent binary numerals, with each octal digit representing a group of 3 binary digits.

operating system

Software that controls the execution of computer programs and that may provide scheduling debugging, input/output control, and related services. Abbreviated O.S.

packet switched data network (PSDN)

A set of equipment and interconnecting links that provides a packet switching communications service to subscribers.

parameter

A variable that is given a specific value that is passed to a program before execution.

password

A character string that uniquely confirms your identity to the system.

process name

A character string that identifies one particular user in a group of users sharing the same system.

prompt

Text displayed that requests information from the user; also the symbol representing specific information that the user should supply, such as the $ command prompt.

protocols

Rules or formats that operating systems must follow to conduct effective communications with other computers in a network.

purge

To erase a file.

reachable node

A node to which the local node has a usable communications path.

read

To transfer information from a peripheral device into core memory or into a register in the CPU.

record

A collection of related data items treated as a unit. A record contains one or more fields. A receiving task receives bytes in one record.

remote

Physically distant from a local computer such as a video display terminal or printer.

remote node

Any node on the network other than the node you are currently using.

remote node list

A list containing node definitions of other nodes that are on the same network.

router

Dedicated hardware used to route traffic on a network. The alternative is to use a portion of a general purpose system such as a VAX.

routing node

A DECnet node that can receive and forward information from one node to another. A routing node can perform other functions that are not limited strictly to routing information.

server

A hardware and software device designed to perform a specific function for many users.

set

In a relational data base model, a collection of things. In a network /hierarchical data base model, a one-to-many relationship, the path by which one record type is connected to another.

substates

In DECnet, an intermediate line state appears as a tag on a line-state display.

synchronous mode

A data transmission method which have related events change simultaneously or in definite timed intervals. See also **asynchronous node**.

topology

The physical arrangement of nodes and connecting hardware that make up the network.

translator

A program or series of programs that changes statements from one machine language into another.

user name

A string consisting of 1 to 39 alphabetic characters identifying a user at a remote node.

utility

A program that helps the user run other programs, operating systems, and equipment.

virtual

Pertaining to occurring in effect, or simulated, but not true in actual fact or form.

virtual disks

Single remote files that appear to be entire DOS volumes.

virtual printers

Printers, accessed remotely, in network resource sharing.

wild card

A symbol used to specify multiple files with related names, without specifying each file by its full name.

Index

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-DEC-DEMO (800-332-3366) using a 1200- or 2400-baud modem. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	————	Local Digital subsidiary or approved distributor
Internal[1]	————	USASSB Order Processing - WMO/E15 *or* U.S. Area Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

[1] For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

PATHWORKS for DOS
DECnet Network Management Guide

AA–PAFHC-TK

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	□	□	□	□
Completeness (enough information)	□	□	□	□
Clarity (easy to understand)	□	□	□	□
Organization (structure of subject matter)	□	□	□	□
Figures (useful)	□	□	□	□
Examples (useful)	□	□	□	□
Index (ability to find topic)	□	□	□	□
Page layout (easy to find information)	□	□	□	□

I would like to see more/less ______________________________

____

What I like best about this manual is ______________________________

____

What I like least about this manual is ______________________________

____

I found the following errors in this manual:
Page Description

_____ ______________________________

_____ ______________________________

_____ ______________________________

Additional comments or suggestions to improve this manual:

____

____

____

I am using **Version** _____ of the software this manual describes.

Name/Title ______________________ Dept. __________

Company ______________________ Date __________

Mailing Address ______________________________

______________________ Phone __________

— Do Not Tear – Fold Here and Tape

digital™

No Postage
Necessary
If Mailed
in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Telecommunications and Networks
LKG1–3/L12
550 KING STREET
LITTLETON, MA 01460–1289

–– Do Not Tear – Fold Here

digital

Order Number: AA-PAFHC-TK
Printed in the U.S.A.