Networks · Communications

DECnet–DOS

User's Guide

AA–EB45B–TV

digital

User's Guide

DECnet–DOS

User's Guide

Order No. AA–EB45B–TV

April 1986

This manual describes how to setup, use, and exit from DECnet–DOS. It details the DECnet–DOS commands used for performing networking tasks.

Supersession/Update Information:	This is a revised manual.
Operating System and Version:	MS–DOS V2.11 PC DOS V2.10 PC DOS V3.10
Software Version:	DECnet–DOS V1.1 DECnet–Rainbow V1.1

digital

AA–EB45B–TV
First Printing, April 1986

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 only be used or copied 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 or its affiliated companies.

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

The following are trademarks of Digital Equipment Corporation:

DEC	PDP	ULTRIX–32
DECmate	P/OS	ULTRIX–32M
DECnet	Professional	UNIBUS
DECUS	Rainbow	VAX
DECwriter	RSTS	VAXcluster
DIBOL	RSX	VMS
digital	RT	VT
MASSBUS	ULTRIX	Work Processor

MS™ and XENIX™ are trademarks of Microsoft Corporation.
IBM is a registered trademark of International Business Machines Corporation.
PC/XT and Personal Computer AT are trademarks of International Business Machines Corporation.

This manual was produced by Networks and Communications Publications.

Contents

Preface

1 Overview of DECnet–DOS

1.1 DECnet Terms and Concepts 1–1
1.2 DECnet–DOS Capabilities 1–3
1.3 DECnet–DOS Utilities 1–4

2 Using the Network Control Program (NCP)

2.1 Running the NCP Utility 2–2
2.1.1 Exiting from NCP 2–3
2.2 General Command Format 2–4
2.3 Typing NCP Commands 2–5
2.4 Using the HELP Command 2–5
2.5 Defining Your Node Name and Address 2–6
2.6 Naming Remote Nodes 2–8
2.6.1 Specifying Default Access Control Information 2–8
2.6.2 Changing and Overriding Default Access Information 2–9
2.6.3 Displaying Default Access Control Information 2–10
2.7 Obtaining Network Information 2–11
2.7.1 Executor Node Information 2–11
2.7.2 Remote Node Information 2–11
2.7.3 Line Information 2–12
2.7.4 Circuit Information 2–12
2.8 Changing Network Information 2–13
2.9 NCP Command Summary 2–15
2.9.1 Standard Definitions 2–17
2.9.2 CLEAR ACCESS 2–18
2.9.3 CLEAR KNOWN ACCESS 2–19
2.9.4 CLEAR EXECUTOR 2–20
2.9.5 CLEAR NODE 2–21
2.9.6 COPY KNOWN NODES 2–22
2.9.7 DEFINE CIRCUIT 2–23

2.9.8 DEFINE EXECUTOR 2–24
2.9.9 DEFINE LINE 2–25
2.9.10 EXIT 2–26
2.9.11 HELP 2–27
2.9.12 MONITOR LOGGING 2–28
2.9.13 QUIT 2–30
2.9.14 READ LOGGING 2–31
2.9.15 SET ACCESS 2–32
2.9.16 SET CIRCUIT 2–33
2.9.17 SET ECHO 2–34
2.9.18 SET EXECUTOR 2–35
2.9.19 SET KNOWN (or ACTIVE) LINKS 2–38
2.9.20 SET LINE 2–39
2.9.21 SET LINK 2–41
2.9.22 SET NODE 2–42
2.9.23 SHOW CIRCUIT 2–44
2.9.24 SHOW EXECUTOR 2–45
2.9.25 SHOW KNOWN ACCESS 2–46
2.9.26 SHOW LAT 2–47
2.9.27 SHOW LINE 2–48
2.9.28 SHOW LINK 2–49
2.9.29 SHOW NODE 2–50
2.9.30 ZERO CIRCUIT 2–51
2.9.31 ZERO EXECUTOR 2–52
2.9.32 ZERO LAT 2–53
2.9.33 ZERO LINE 2–54

3 Accessing Local and Remote Files

3.1 Running the NFT Utility 3–2
3.2 Defining Remote Node Access Information 3–2
3.2.1 Defining Access Control Information with NFT 3–3
3.2.2 Using Default Access Control Information 3–4
3.2.3 Changing the Default Access Control Information 3–4
3.2.4 Displaying Access Control Information 3–5
3.3 File Specifications 3–6
3.3.1 Specifying Remote Files 3–6
3.3.2 Using Wildcards 3–7
3.3.3 Using Question Marks, Percent Signs, and Asterisks as Wildcards 3–8
3.4 Types of Files and Systems 3–8
3.5 Typing NFT Commands 3–10
3.5.1 Command Prompts 3–11
3.5.2 Command Switches 3–11
3.6 Using the HELP Command 3–16
3.7 File Operations 3–16
3.7.1 Appending Files 3–17
3.7.2 Listing Files 3–17

3.7.3 Copying Files 3–18
3.7.4 Deleting Files 3–18
3.7.5 Displaying Directory Information 3–19
3.7.6 Printing Files 3–19
3.7.7 Typing a File's Contents 3–19
3.7.8 Running Command Files 3–20
3.7.9 Logging File Operations 3–20
3.8 NFT Command Summary 3–21
3.8.1 APPEND 3–22
3.8.2 COPY 3–24
3.8.3 DELETE 3–27
3.8.4 DIRECTORY 3–28
3.8.5 EXIT 3–30
3.8.6 HELP 3–31
3.8.7 PRINT 3–32
3.8.8 SET 3–33
3.8.9 SHOW 3–34
3.8.10 SUBMIT 3–35
3.8.11 TYPE 3–36
3.9 Providing Remote File Access to Your Node Using FAL 3–37
3.9.1 /ASCII 3–38
3.9.2 /BINARY 3–39
3.9.3 /ERROR 3–40
3.9.4 /LOG 3–41
3.10 Using Your Computer as a Server 3–42

4 Logging On to a Remote Host

4.1 Using the SETHOST Command 4–2
4.1.1 /ABORT 4–3
4.1.2 /CTERM 4–4
4.1.3 /EDIT 4–5
4.1.4 /EXIT 4–6
4.1.5 /HELP 4–7
4.1.6 latservicename or nodespec 4–8
4.1.7 /LOG 4–9
4.1.8 /SAVEDEFAULT 4–10
4.1.9 /SESSION 4–11
4.1.10 /SHOW 4–12
4.1.11 /TAKE 4–13
4.2 Interrupting a Session 4–14
4.2.1 <RETURN> 4–15
4.2.2 <H>ELP 4–15
4.2.3 <L>OG 4–15
4.2.4 <R>ECONNECT 4–15
4.2.5 <E>DIT 4–15
4.2.6 <S>ESSION 4–16
4.2.7 <A>BORT SESSION 4–16

4.2.8 <N>EW SESSION 4–16
4.3 Your Rainbow 100 and IBM PC/XT or PC AT System as a VT102 Terminal 4–16
4.4 Using Your IBM System Keyboard during Terminal Emulation 4–20
4.4.1 Letters, Numbers and Punctuation Marks 4–22
4.4.2 Function Keys 4–22
4.4.3 Arrow Keys 4–24
4.4.4 Numeric Keypad 4–25
4.5 Setting VT102 Class Terminal Features Using SET-Up on the IBM PCs 4–26
4.5.1 Entering SET-Up Mode 4–26
4.5.2 Selecting a Parameter SET-Up Menu 4–27
4.5.3 Saving Set-Up Values 4–28
4.5.4 Using the Set-Up Values 4–28
4.5.5 Recalling the SET-Up Values 4–28
4.5.6 Exiting from SET-Up Mode 4–28
4.5.7 Editing Text During a Remote Session 4–28
4.6 Disconnecting from a Host 4–29

5 Using a Remote Printer and Disk

5.1 Installing the Virtual Device Drivers 5–2
5.2 Using NDU to Control Virtual Disks 5–3
5.3 Typing NDU Commands 5–4
5.3.1 Command Verbs 5–4
5.3.2 Entities 5–4
5.4 Using the HELP Command 5–6
5.5 NDU Command Summary for Virtual Disks 5–6
5.5.1 CLOSE 5–7
5.5.2 CREATE 5–8
5.5.3 DELETE 5–10
5.5.4 HELP 5–11
5.5.5 OPEN 5–12
5.5.6 SHOW 5–13
5.6 Using the NDU to Control Remote Printers 5–14
5.7 Typing NDU Commands 5–14
5.7.1 Command Verbs 5–15
5.7.2 Entities 5–15
5.8 Using the HELP Command 5–16
5.9 NDU Command Summary for Remote Printers 5–16
5.9.1 CLOSE 5–17
5.9.2 CREATE and OPEN 5–18
5.9.3 HELP 5–19
5.9.4 SHOW 5–20

6 Using the DECnet–DOS Mail Utility

6.1 Running the Mail Utility 6–1
6.1.1 REPLY_ADDRESS 6–2
6.1.2 DEFAULT_NODE 6–2
6.1.3 CARBON_COPY 6–3
6.1.4 EDITOR 6–4
6.1.5 PERSONAL_MESSAGE 6–4
6.2 Starting the Mail Utility 6–4
6.2.1 Entering Mail Commands from the Command Line 6–4
6.2.2 Entering Mail Commands from the Mail Prompt 6–5
6.3 Using Distribution Lists 6–5
6.4 Using Qualifiers 6–7
6.5 Getting Help for Mail 6–7
6.6 Mail Command Summary 6–8
6.6.1 EXIT 6–8
6.6.2 HELP 6–8
6.6.3 MAIL and SEND 6–9

7 Testing the Network

7.1 Overview of the Network Test Utility 7–2
7.2 Running the Network Test Utility 7–5
7.2.1 Starting the NTU 7–6
7.2.2 Leaving the NTU 7–7
7.3 Getting HELP for NTU 7–7
7.4 Using the NTU Commands 7–7
7.5 NTU Command Summary 7–9
7.5.1 CLEAR ACCESS 7–13
7.5.2 CLEAR ACCOUNT 7–14
7.5.3 CLEAR COUNT 7–15
7.5.4 CLEAR DEFAULTS 7–16
7.5.5 CLEAR LENGTH 7–17
7.5.6 CLEAR PASSWORD 7–18
7.5.7 CLEAR USER 7–19
7.5.8 CLEAR WITH 7–20
7.5.9 EXIT 7–21
7.5.10 HELP 7–22
7.5.11 LOOP CIRCUIT 7–23
7.5.12 LOOP EXECUTOR 7–26
7.5.13 LOOP LINE CONTROLLER 7–27
7.5.14 LOOP NODE 7–28
7.5.15 MIRROR 7–30
7.5.16 MONITOR LOGGING 7–31
7.5.17 READ LOGGING 7–33
7.5.18 SET ACCOUNT 7–34
7.5.19 SET CIRCUIT 7–35

7.5.20 SET COUNT 7–36
7.5.21 SET ECHO 7–37
7.5.22 SET LENGTH 7–38
7.5.23 SET LINE STATE 7–39
7.5.24 SET LINK 7–40
7.5.25 SET PASSWORD 7–41
7.5.26 SET USER 7–42
7.5.27 SET WITH 7–43
7.5.28 SHOW ACTIVE LINKS 7–44
7.5.29 SHOW ACTIVE NODES 7–45
7.5.30 SHOW ADJACENT NODE 7–46
7.5.31 SHOW CIRCUIT 7–47
7.5.32 SHOW DEFAULTS 7–48
7.5.33 SHOW EXECUTOR 7–49
7.5.34 SHOW KNOWN LINKS 7–50
7.5.35 SHOW KNOWN NODES 7–51
7.5.36 SHOW LINE 7–52
7.5.37 SHOW NODE 7–53
7.5.38 ZERO CIRCUIT 7–54
7.5.39 ZERO EXECUTOR 7–55
7.5.40 ZERO LINE 7–56

A NCP Error Messages

A.1 Introduction A–1
A.2 Types of NCP Error Messages A–1
A.3 Command Line Errors A–2
A.4 File Input/Output (I/O) Errors A–5
A.5 Network Error Messages A–6
A.6 NTU Errors A–8

B NFT Error Messages

B.1 Types of NFT Error Messages B–2
B.2 Command Line Error Messages B–3
B.3 File I/O Error Messages B–6
B.4 Network Error Messages B–8
B.5 Secondary Error Messages B–9

C SETHOST Error Messages

C.1 Network Errors C–1
C.2 Execution Error Messages C–5

D NDU Error Messages

D.1 Virtual Disk Error Messages D–2
D.1.1 Network Errors D–2
D.1.2 Command Line Error Messages D–4
D.1.3 Execution Error Messages D–6
D.2 Virtual Printer Error Messages D–10
D.2.1 Network Errors D–10
D.2.2 Command Line Error Messages D–12
D.2.3 Execution Error Messages D–13

Glossary

Examples

3–1 Copying a File to a Remote Node 3–6
3–2 An NFT Command 3–10

Figures

1–1 A Sample DECnet Network 1–2
2–1 Defining Your Node Name and Address 2–7
2–2 Remote Node Names 2–10
4–1 Rainbow 100 System Keyboard 4–17
4–2 IBM PC/XT System Keyboard 4–18
4–3 IBM PC AT System Keyboard 4–19
5–1 An NDU Command 5–4
5–2 An NDU Command 5–14
7–1 LOOP CIRCUIT Test (to the Loopback Plug on the Controller) 7–3
7–2 LOOP CIRCUIT Test (to the Loopback Plug on the Cable) 7–3
7–3 LOOP CIRCUIT Test (to the Local Modem) 7–3
7–4 LOOP CIRCUIT Test (to the Remote Modem) 7–4

Tables

2–1 NCP Commands 2–15
2–2 NCP Command Parameters 2–17
3–1 Operating Systems and File Specifications 3–7
3–2 Copying Files from a Remote Node 3–9
3–3 Copying Files to a Remote Node 3–10
3–4 Valid Switches for Copying Files from a Remote Node 3–12
3–5 Valid Switches for Copying Files to a Remote Node 3–13
3–6 Valid Switches for the DIRECTORY Command 3–15
3–7 Valid NFT Commands 3–21
4–1 VMS Commands and Keys 4–21

4–2 Commands and Keys for Systems Other than VMS 4–22
5–1 NDU Command Entities for Virtual Disks 5–4
5–2 NDU Commands for Virtual Disks 5–6
5–3 NDU Command Entities for Remote Printers 5–15
5–4 NDU Commands for Remote Printers 5–16
7–1 NTU SHOW Command Parameters 7–8
7–2 Common NTU Command Variables 7–9
7–3 NTU Command Summary 7–10

Preface

DECnet–DOS is a set of software communication products that enables individual computer systems, such as your personal computer, to communicate with one another in a **network**. The individual systems within a network are called **nodes**.

The term DECnet–DOS refers to the following two products:

- DECnet–DOS, Version 1.1 for the IBM PC, PC/XT, and Personal Computer AT (running the IBM PC DOS Version 2.10 or 3.10 operating systems).
- DECnet–Rainbow, Version 1.1 for the Rainbow 100 computers running the Rainbow MS–DOS Version 2.11 operating system. (DECnet–Rainbow V1.1 does not support Ethernet configurations.)

The term DOS refers to the following two operating systems:

- MS–DOS running on the Rainbow personal computer.
- IBM Personal Computer DOS (PC DOS) running on the IBM PC, the IBM PC/XT, and the IBM PC AT.

Manual Objectives

The *DECnet–DOS User's Guide* describes the DECnet–DOS software product and the eight utilities you can use with DECnet–DOS.

The guide assumes that you are familiar with the use of the Rainbow or IBM personal computers, and the MS–DOS or PC DOS operating system. Throughout this guide, the term computer refers to the Rainbow personal computer as well as the IBM PC, the IBM PC/XT, and the IBM PC AT.

Intended Audience

This manual is intended for users who want to expand the capabilities of their Rainbow and/or IBM personal computers in order to share data and resources with other DECnet systems.

Structure of this Manual

This manual consists of 7 chapters, 4 appendices and one glossary:

Chapter 1 Introduces the capabilities of DECnet–DOS. It also defines some basic concepts within a DECnet environment and introduces the eight DECnet–DOS utilities.

Chapter 2 Describes how to use the Network Control Program (NCP) 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 the network.

Chapter 3 Describes how to use the Network File Transfer utility (NFT) to access files on other nodes in the network. This chapter also discusses the File Access Listener (FAL) and the Job Spawner.

Chapter 4 Describes how to use the SETHOST utility to connect to another node in the network so you can access that node's resources.

Chapter 5 Describes how to use the Network Device Utility (NDU) to set up disk drives and use printers on remote nodes.

Chapter 6 Describes how to use the DECnet–DOS Mail Utility for sending mail messages and text files to other nodes in the network.

Chapter 7 Describes how to use the Network Test Utility (NTU) to test the network hardware and software.

Appendix A Lists the error messages you can receive while using NCP.

Appendix B Lists the error messages you can receive while using NFT.

Appendix C Lists the error messages you can receive while using SETHOST.

Appendix D Lists the error messages you can receive while using NDU.

Glossary Contains a list of DECnet–DOS terms.

Graphic Conventions Used in This Document

The following graphic conventions are used in this manual:

Convention	Meaning
`monospaced type`	Monospaced type indicates examples of system output or user input. System output is in black; user input is in red.
UPPER CASE	Represents acceptable abbreviations, for example **DEL**ETE. The abbreviations are displayed as bold characters.
UPPERCASE	Uppercase in commands and examples indicates that you should enter the characters as shown (enter either uppercase or lowercase).
italics	Italics in commands and examples indicate that either the system supplies or you should supply a value.
{ }	Braces 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.
[ ]	Square brackets 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.
( )	Parentheses 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 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 CTRL key while you press the *x* key, where *x* is a letter.

Note that unless otherwise specified, you should end every command line by pressing the RET key. On the Rainbow personal computer, this key is labeled Return. On the IBM personal computer, this key is labeled ←┘.

Associated Documents

The following documents are included in the documentation set if you are using a Rainbow computer.

- *DECnet–Rainbow Installation Guide*
- *DECnet–DOS Getting Started*
- *DECnet–DOS User's Guide*, which you are reading
- *DECnet–DOS Programmer's Reference Manual*
- *DECnet–DOS Mini–Reference Guide*
- *DECnet–Rainbow Release Notes*

You should also have the following manuals available for reference:

The installation guide and introductory manuals for your computer.

The following documents are included in the documentation set if you are using an IBM personal computer.

- *DECnet–DOS Installation Guide*
- *DECnet–DOS Getting Started*
- *DECnet–DOS User's Guide*, which you are reading
- *DECnet–DOS Programmer's Reference Manual*
- *DECnet–DOS Mini–Reference Guide*
- *DECnet–DOS Release Notes*

You should also have the following manuals available for reference:

The installation guide and introductory manuals for your computer.

1 Overview of DECnet–DOS

This chapter introduces:

- DECnet terms and concepts.
- Network capabilities provided by DECnet–DOS.
- The eight DECnet–DOS utilities you can use.

1.1 DECnet Terms and Concepts

Digital's DECnet products connect individual computer systems, such as your personal computer, together in flexible configurations called **networks**. Individual systems in a network, called **nodes**, share resources and exchange information, files, and programs.

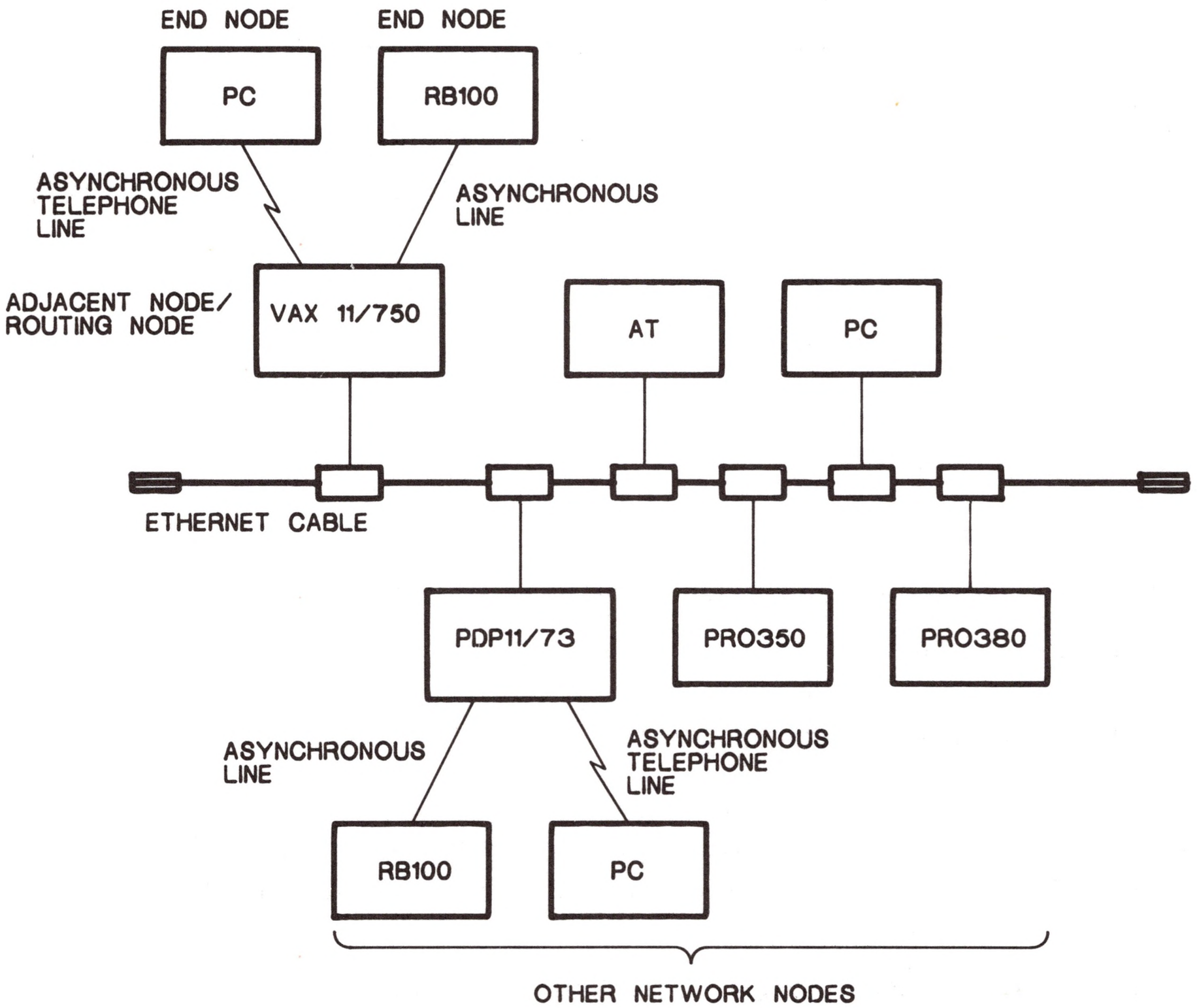
There is a different DECnet product for each Digital operating system. For example:

- DECnet–VAX is used on VAX computers running the VAX/VMS operating system.
- PRO/DECnet is used on Professional computers running the P/OS operating system.
- DECnet–Rainbow is used on the Rainbow personal computer running the MS–DOS operating system.
- DECnet–DOS is used on the IBM PC, the IBM PC/XT or the IBM Personal Computer AT running the PC DOS operating system.

NOTE

In this document, the term DECnet–DOS refers to the software on both the Rainbow and IBM personal computers.

Figure 1–1 shows DECnet–DOS in a DECnet network.



TWO357

Figure 1–1: A Sample DECnet Network

Figure 1–1 shows an **end node** connected to a VAX computer. In a DECnet–DOS environment, an end node can also be called the **local node** and the **executor node**.

- An **end node** can receive and transmit information for its own use. It cannot receive and then forward information intended for other nodes.
- A **local node** is the node you are working on when you enter commands at the keyboard.
- An **executor node** is a node that performs network management functions. This node enables you to obtain information about the network.

From the vantage point of the personal computer shown in Figure 1–1, the other network nodes in the diagram are called **remote nodes**. Two other terms define the VAX node in particular:

- **Adjacent node**

  An adjacent node is a node that is physically connected to your local node by a single line.

- **Routing node**

  The adjacent node shown in Figure 1–1 is a routing node. A routing node automatically forwards (or routes) information, files, and programs between nodes in the network. This allows the end node to copy data to and from any other node in the network. An end node cannot automatically receive and then forward data intended for another node.

In a DECnet–DOS environment, you can set up your node as either:

1. An asynchronous DDCMP DECnet end node, using the communications port.

   OR

2. An Ethernet end node, using a vendor-supplied Ethernet communications board and Ethernet hardware to connect to a Local Area Network (LAN).

Note that DECnet–Rainbow only supports asynchronous communication. DECnet–DOS supports both asynchronous and Ethernet communication.

1.2 DECnet–DOS Capabilities

DECnet–DOS provides the following capabilities:

- **Network management**, which allows you to control, monitor, and test DECnet–DOS software. You can reconfigure your network as the need arises.
- **Remote file access**, which allows you to access files on remote nodes. You can store and retrieve information on remote nodes. By storing the most current version of a file on a remote node, you eliminate the need for keeping a version of the file on each node.

- **File transfer**, which allows you to exchange files with other nodes. This speeds up the information flow between organizations and decreases the amount of paperwork.
- **Resource sharing**, which eliminates the need for duplicating resources at each node. With resource sharing, many nodes can use the same printers, storage facilities, and processing capabilities.
- **Resource server functions**, which provide services to other nodes in the network (for example: file storage).
- **Mail**, which allows you to send messages to other nodes in the network.

1.3 DECnet–DOS Utilities

The following list is a summary of the DECnet–DOS utilities you can use.

- **Network Control Program (NCP)**

  NCP allows you to perform network management. NCP is run as a part of the installation procedure before you can use your pc as a node in the DECnet network. NCP is discussed in Chapter 2 of this guide.

- **Network File Transfer (NFT) utility**

  NFT allows you to transfer files between your local node and remote nodes within the DECnet network. NFT also provides other file-related services. NFT is discussed in Chapter 3 of this guide.

- **File Access Listener (FAL)**

  FAL provides access to your computer from other nodes in the network. FAL is discussed in Chapter 3 of this guide.

- **Job Spawner**

  The job spawner allows your computer to act as a server for multiple service functions. The spawner activates FAL and DTR to run on your node. The job spawner is discussed in Chapter 3 of this guide.

- **SETHOST utility**

  SETHOST, with network virtual terminal services, allows you to log on to a host node. You can use your personal computer as if it were directly connected to the host node, and use the host's resources. SETHOST is discussed in Chapter 4 of this guide.

- **Network Device Utility (NDU)**

  NDU allows you to define virtual disk drives and virtual printers on remote systems. NDU capabilities include:

  – Defining disks on a remote system and using them as if they were directly connected to your computer.

  – Allows you to direct text to a remote node to be queued later for printing.

  NDU is discussed in Chapter 5 of this guide.

- **DECnet–DOS Mail Utility**

  The Mail utility lets you send mail messages and text files across the network. The Mail utility is described in Chapter 6 of this guide.

- **Network Test Utility (NTU)**

  NTU tests the network hardware and software. NTU provides a subset of procedures associated with NCP. NTU is discussed in Chapter 7 of this guide.

2
Using the Network Control Program (NCP)

The Network Control Program (NCP) is the DECnet management utility that accepts commands from your terminal to perform the following set of functions:

- Identify your personal computer to the rest of the network.
- Establish node names and default access control information for other nodes in the network.
- Establish access control information for allowing local file access.
- Change parameters which control how the hardware and software provide network services.
- Monitor network activity.
- Monitor network error counters.

This chapter describes:

- Running NCP (see Section 2.1).
- General NCP command formats (see Section 2.2).
- Typing NCP commands (see Section 2.3).
- Getting help (see Section 2.4).
- Identifying your node to other nodes in the network (see Section 2.5).
- Establishing node names and access control information for other network nodes (see Section 2.6).
- Obtaining network information (see Section 2.7).

- Changing network information (see Section 2.8).
- Each NCP command, its format and usage (see Section 2.9).

It is required that you know the person responsible for your network. If you have questions regarding your node or the network, this person should be able to provide you with extra information or assistance.

2.1 Running the NCP Utility

To run the NCP utility, you need to use NCP commands. You can supply NCP commands by using one of two methods:

- Single command method:

  Enter a command string that includes NCP followed by the name of the specific NCP command. Press the RET key. After each NCP command is entered, the MS–DOS prompt is returned.

```
E>NCP command(RET)
E>
```

This method can also be used for entering commands from redirected input files. To enter commands from a file, enter a command string that includes NCP followed by the @ sign and the name of the file. Then press the RET key.

```
E>NCP @file-name(RET)
```

NOTE

The SET ECHO command allows you to specify whether a redirected input file is displayed on your screen as the file is executed. (See Section 2.9.17 on how to turn echo on and off using the SET ECHO command.)

When NCP has finished, an error level value is returned that can be used by the IF ERRORLEVEL statement in batch files. If NCP has successfully run, a value of 0 is returned. If the execution was unsuccessful, a value greater than 0 is returned.

- Multiple command method:

  You supply the name of the utility (NCP) and press the RET key. NCP then responds with its own prompt. (You cannot use this method for submitting NCP commands in batch mode).

```
E>NCP(RET)
NCP>
```

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

```
NCP>command(RET)
NCP>command(RET)
NCP>command(RET)
```

2.1.1 Exiting from NCP

To exit from NCP, you can use one of several NCP commands. The EXIT and CTRL/Z commands have the same effect, whereas, the QUIT command has a different result.

- Enter the EXIT command and press the RET key. The system returns with the MS–DOS prompt. You can also use CTRL/Z to exit from NCP. While holding down the CTRL key, press the letter Z key. Then press the RET key.

  If you exit from NCP using either EXIT or CTRL/Z, any permanent parameter changes that you made with SET and/or DEFINE commands are saved in the DECPARM.DAT file.

  ```
  NCP>EXIT RET
  ```

  The MS–DOS prompt is returned.

  ```
  E>
  ```

- Enter the QUIT command and press the RET key.

  ```
  NCP>QUIT RET
  ```

  If you have made any changes, the system responds with:

  ```
  Do you want to save changes to permanent databases (Y or N) ?
  ```

  If you answer **Y**, all current changes to executor, circuit and line parameters are saved in DECPARM.DAT. If you answer **N**, the changes will not be saved. When you reboot your system, the network process checks to see if there is a DECPARM.DAT file. If the file does not exist, the process initializes using its own set of default parameters.

  The system responds with the MS–DOS prompt:

  ```
  E>
  ```

- Use CTRL/C

  Typing CTRL/C will also cause NCP to exit, without saving any of the changes you made.

2.2 General 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 qualifier. Generally, the order in which you specify parameters does not matter.

Examples of the NCP command format follow:

Command Verbs	Entities	Parameters
CLEAR	NODE *node-address*	All
SET	LINE *line-id*	STATE ON

It is not necessary to understand the details of the command format. However, you should be able to recognize the parts of each command. The following example shows parts of an NCP command:

```
NCP>SHOW NODE PARIS CHARACTERISTICS TO NODE.LIS(RET)
```

verb entity parameter qualifier

Eleven command verbs are used with the NCP utility. Each verb causes NCP to take a different action:

- **CLEAR** removes or clears parameters for a specific node or the executor.
- **COPY** copies the node database to another file (without the access information). This is useful for making master copies of the DECNODE.DAT file for distribution.
- **DEFINE** establishes specific parameters which take effect after you reboot your system.
- **EXIT** allows you to exit from NCP and save parameter changes.
- **HELP** provides on–line information about the NCP utility and specific commands.
- **MONITOR** continuously displays event logging information on the terminal screen.
- **QUIT** allows you to exit from NCP, and gives you the choice of saving parameter changes.
- **READ** displays event logging information on your terminal screen. You can also redirect it to a specified output file.
- **SET** establishes specific parameters for a network entity.

- **SHOW** displays parameters that you defined with the SET or DEFINE command. It also displays counters and other parameters that cannot be set, and may be redirected to a specific output file.
- **ZERO** zeroes the counters associated with a specific network entity.

If you exit from NCP using EXIT or CTRL/Z, the permanent executor, circuit, and line parameters set with the SET or the DEFINE command are saved in the file DECPARM.DAT (for use at next reboot).

If you exit from NCP using the QUIT command, you choose whether to save or ignore permanent parameter changes for next reboot. Note that changes made using the SET command always become effective immediately. (See Section 2.1.1 on how parameter changes are affected when you use the QUIT command.)

Although you can use both DEFINE and SET to modify network parameters, each command has a different result when you issue it. The DEFINE command establishes specific parameters, but they will not take effect until the next time you reboot your system. DEFINE lets you restore any default parameters that you may have temporarily changed with the SET command. SET also establishes specific parameters, but the changes take effect as soon as the command is issued. (Some parameters changed by the SET command retain their values after the next reboot while some do not. Refer to Sections 2.9.15 through 2.9.22 for more information about SET commands.)

2.3 Typing NCP Commands

To type an NCP command, enter the command at the NCP command prompt. You can abbreviate command keywords to three or more unique characters. (An exception to this is the use of the commands LINK and LINE. The first three characters for each command are the same, so that you must use all four characters to properly identify the command you want to use.)

For example, you can type **SHO** for SHOW. However, for clarity and consistency, all examples in this chapter show the full command format.

2.4 Using the HELP Command

In order to use the HELP command, the file NCPHELP.BIN must be in the same directory as DECPARM.DAT. If you need assistance in selecting NCP commands, type HELP, then press the RET key. For example:

```
NCP>HELP(RET)
```

The system responds with:

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

HELP is available for the following commands:

@   ; CLEAR    COPY     DEFINE      EXIT    MONITOR
QUIT  READ     SET    SHOW    ZERO
```

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

```
NCP>HELP CLEAR(RET)
```

The system responds with:

```
HELP is available for the following commands:

CLEAR ACCESS      CLEAR KNOWN ACCESS      CLEAR EXECUTOR      CLEAR NODE
```

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

```
NCP>HELP CLEAR EXECUTOR(RET)
```

The system responds with:

```
Removes specified executor node parameters from the local node's
database (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.
```

2.5 Defining Your Node Name and Address

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

During software installation, your local node address and name are set when you perform the installation procedure. (It is not necessary for you to repeat this step once installation is completed.) The *DECnet–DOS Installation Guide*and the *DECnet–Rainbow Installation Guide*provide a step-by-step explanation of the installation procedure.

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 executor's node name and address become effective after the system has been rebooted.

- A **node name** is an identification string consisting of one to six alphanumeric characters, including at least one alphabetic character. For example:

  PARIS3

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

  area.number

  where

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

  number can be in the range of 1 to 1023. It is separated from the area number with a period.

  For DECnet–DOS Version 1.1, your area number must match the area number of the node that is routing information for you. For asynchronous DECnet–DOS, your area number must match the area number of your adjacent node.

  For example:

  2.975

Figure 2–1 illustrates the use of the DEFINE EXECUTOR command. A node name RAINBO is assigned to an area number 2.975.

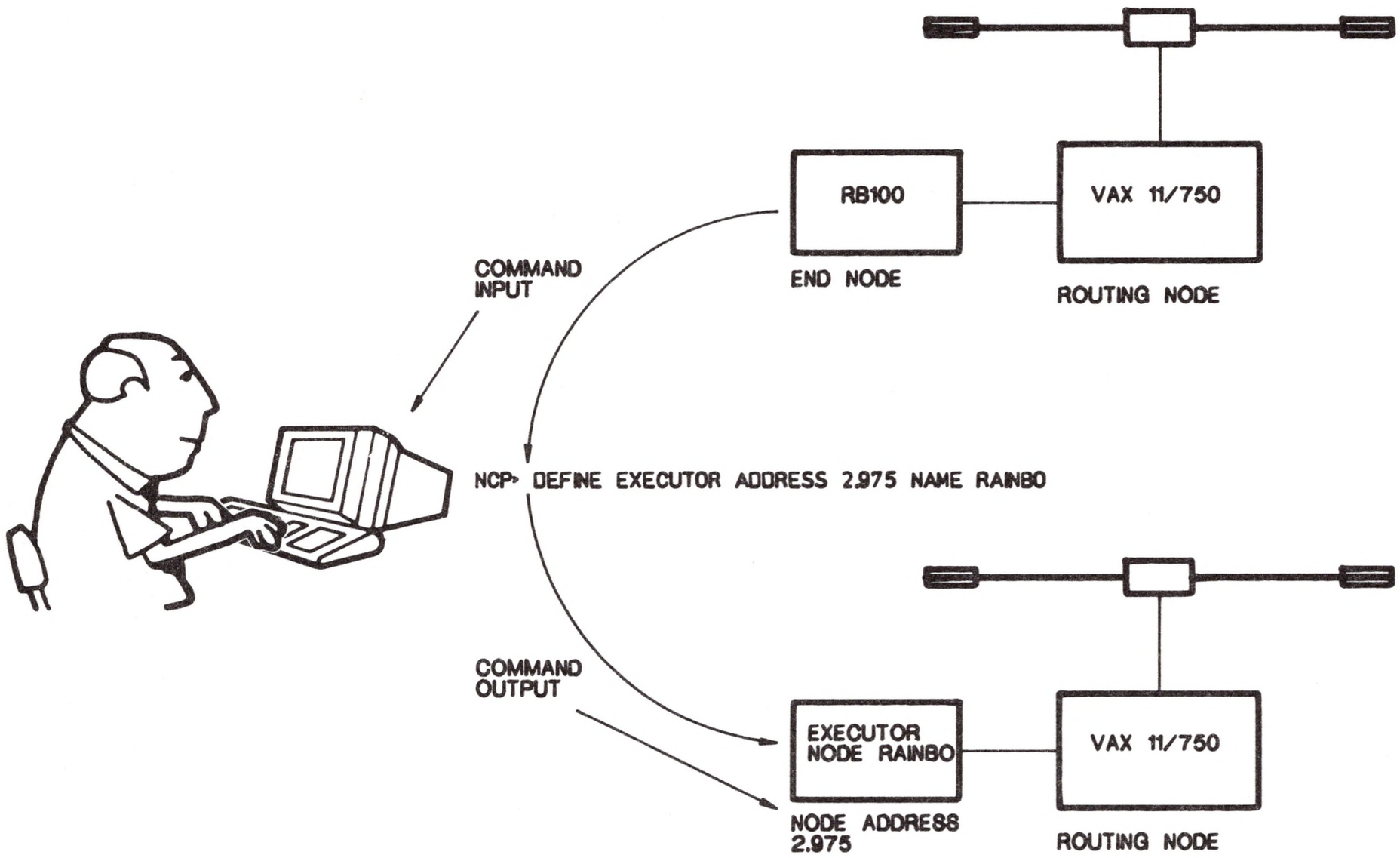


Figure 2–1: Defining Your Node and Address

2.6 Naming Remote Nodes

At the local node, you create a list of the remote nodes that you want to access by name. This list can include access control information that allows you to access a specific node. To specify remote node information, use the SET NODE command.

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

- **Node address**

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

- **Node name**

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

2.6.1 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:

- **Username**

  A character string consisting 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 may 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 not used on most systems, including DECnet–DOS and DECnet–Rainbow.)

You can specify default access control information using NCP, or you can explicitly specify the information during a DECnet–DOS operation. For example, when you specify access control data during an NFT session, this information is used by any following command dealing with the same node. Once you exit from NFT, the access control information is not saved. If you run NFT at another time, you cannot use the access control information that was set in a previous NFT session. However, you can use access control information previously specified with NCP. If NCP provides defaults and you decide to specify different values with NFT commands, these new values are used in place of the defaults. Refer to Chapter 3 of this manual for setting access control information during an NFT session.

You can also specify that the node you want to access is a LAT host node. Use the following designation to specify this node:

LAT–HOST Designate this node as a LAT host.

If you specify default access control information using NCP, you can use the node name alone in network operations during a session involving a DECnet–DOS utility.

To specify this information, use the SET NODE command. The SET NODE command assigns a node name plus access control information to a unique node address. (See Section 2.9.22 for more details on this command.) For example:

```
NCP>SET NODE 2.375 NAME LONDON USER SMITH PASSWORD OPEN(RET)
```

This command assigns node name LONDON to node address 2.375. It also specifies the user's name as SMITH and the password as OPEN.

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

1. The utility sees if the data was specified with NCP. The utility checks the permanent database file, DECALIAS.DAT. Refer to the *DECnet–DOS Programmer's Reference Manual* for a description of the *dnet_getalias()* function call.
2. If no access control data was specified using NCP, it is assumed that the remote node does not require access control information, and the requested operation is attempted.

NOTE

If you specify access control data with NFT, it is not necessary to specify the information again for use within the same NFT utility session.

2.6.2 Changing and Overriding Default Access Information

You can override the default access control information by specifying different information during a DECnet–DOS utility session. The process of overriding default access control information during an NFT session is explained in Chapter 3 of this manual.

If you want to change access defaults with NCP, you can use the SET NODE command. This command resets node parameters stored on the local node for the specified remote node. For example:

```
NCP>SET NODE 2.31 NAME BOSTON USER SAM PASSWORD CAT(RET)
```

This command sets different access control information for the node BOSTON.

2.6.3 Displaying Default Access Control Information

The NCP SHOW KNOWN NODES command displays default access control information for a set of remote nodes. Figure 2–2 shows remote nodes in a sample DECnet–DOS network. Each node has a specific node name.

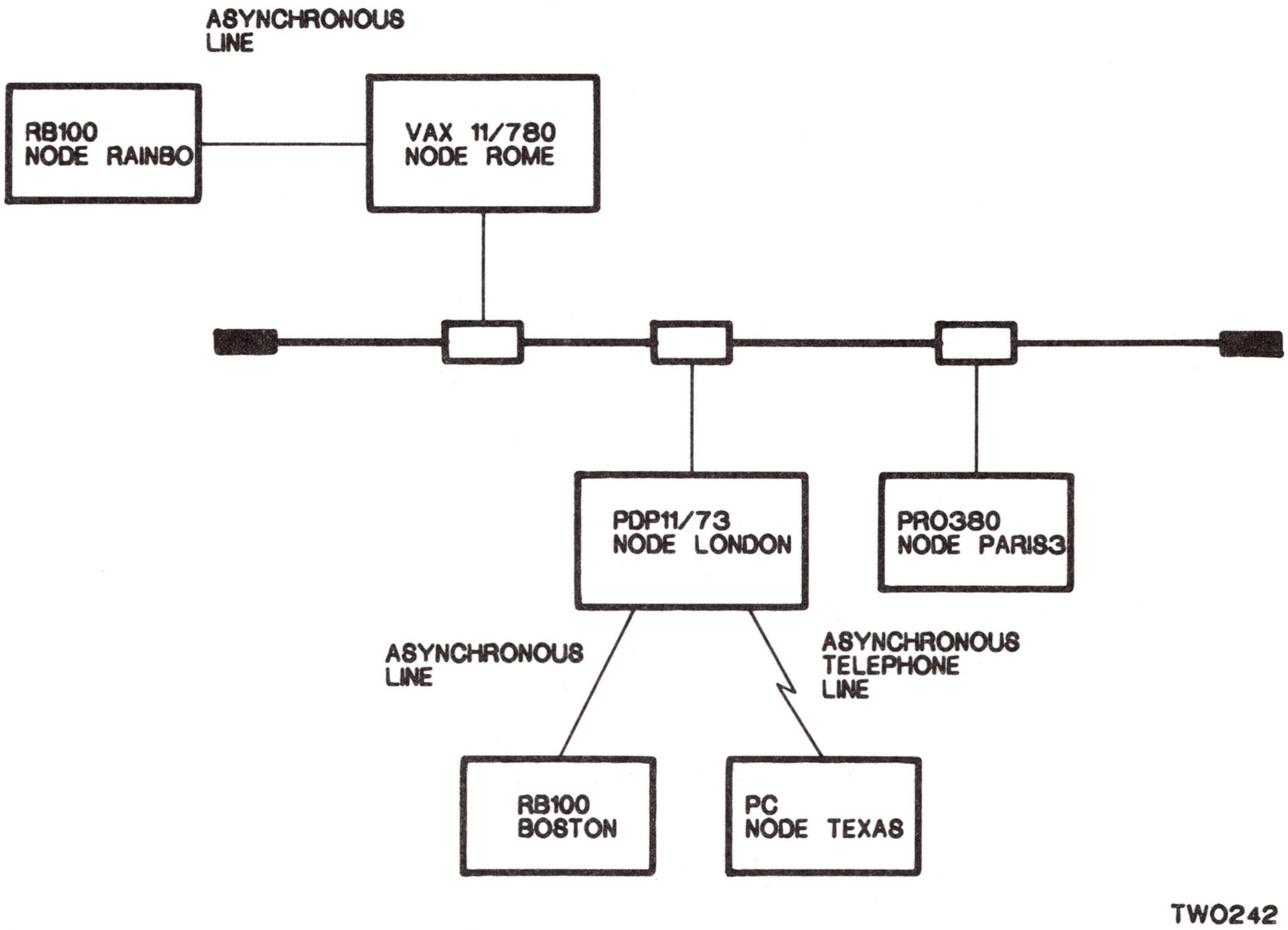


Figure 2–2: Remote Node Names

To display information for the remote nodes that appear in Figure 2–2, type the NCP command, SHOW KNOWN NODES. This command displays the following information for nodes that are **known** to the local node RAINBO:

- Node address
- Node name
- Number of active logical links
- Access control information (output as account information)

```
NCP>SHOW KNOWN NODES(RET)
```

The system responds with:

```
Node Summary as of 1-SEP-1985 1:31:17

Executor node                 = 2.975 (RAINBO)

State                         = On
Identification                = DECnet-DOS Office Group

Node     Node     Active  LAT     Account
Address  Name     Links           Information

2.25     TEXAS    0               /DALLAS/...

2.47     PARIS3   0       L       /JOHN/...

2.48     ROME     0       M       /SMITH/.../2345

2.375    LONDON   0       LM      /KEVIN/...
```

2.7 Obtaining Network Information

Using NCP commands, you can display the following network information on your screen:

- Node, line and circuit parameters can be displayed by specifying CHARACTERISTICS with a specific network entity.
- Node names, node addresses, and access control information for remote nodes can be displayed if they have been defined by the executor node.
- Counters display statistics about the flow of network messages to and from your node. The counters also record error conditions and accumulate their totals for you.
- A network event logger records network events. See READ LOGGING (Section 2.9.14).
- Network status (state) and other related information can be displayed for a specific node, line, or circuit.

2.7.1 Executor Node Information

To display information about the executor node, type:

```
NCP>SHOW EXECUTOR CHARACTERISTICS(RET)
```

This command summarizes the network parameters that you specified for the executor node.

2.7.2 Remote Node Information

To determine if you defined a specific node with the SET NODE command, type:

```
NCP>SHOW NODE node-id(RET)
```

This command displays the node name, node address, access control information, and the number of active logical links for node *node-id* from this node.

If you want information only for nodes with active logical links, type:

```
NCP>SHOW ACTIVE NODES(RET)
```

This command displays the node name and address, the access control information, and the number of active links from this node to the adjacent node and any other nodes with active logical links to your node. The executor node is always displayed.

2.7.3 Line Information

In the context of DECnet–DOS, a line is the physical line connecting the local node with the adjacent node. There are two possible line states:

- **ON** – The line is ready for use. To determine if the line is working properly, use the SHOW LINE CHARACTERISTICS command. (See Section 2.9.27 for an explanation.)
- **OFF** – The line's database and parameters are present, but the line is not available for any type of network activity.

There are three substates controlled through network management. They can only be observed when the LINE STATE is ON. Use the SHOW LINE CHARACTERISTICS command to display line substates. The substates are:

- **SYNCHRONIZING** – The line is to be or is being initialized.
- **RUNNING** – The line is in normal running state.
- **SERVICE** – The line is reserved for loopback testing.

To display status information about the physical line, type:

```
NCP>SHOW LINE STATUS(RET)
```

This command displays the current status of the line.

2.7.4 Circuit Information

In the context of DECnet–DOS, a circuit is the communications path established between the local node and its adjacent node. There are two possible circuit states:

- **ON** – The circuit is ready for use. To determine if the circuit is working properly, use the SHOW CIRCUIT CHARACTERISTICS command. (See Section 2.9.23 for an explanation.)
- **OFF** – The circuit's database and parameters are present, but the circuit is not available for any type of network activity.

There are three substates controlled through network management. Use the SHOW CIRCUIT CHARACTERISTICS command to display circuit substates. The substates are:

- **SYNCHRONIZING** – The circuit is being initialized.
- **STARTING** – The circuit is initialized.
- **RUNNING** – The circuit is in running state.

To display status information about the circuit, type:

```
NCP>SHOW CIRCUIT STATUS(RET)
```

This command displays the current status of the line.

2.8 Changing Network Information

Using NCP commands, you can change the executor node, line, and circuit parameters according to the needs of your network. It is recommended that you do not change the settings for these parameters unless you are familiar with DECnet concepts.

The following network parameters are some of the features that you can change:

At Installation

Executor node name and address

The node name is a one to six alphanumeric character string associated with a specific node address. The string must contain at least one alphabetic character. The node address is a numeric string that identifies the location of the node in the network.

Line speed (applies to asynchronous DDCMP lines only)

The speed of the communications line can be set to a specific baud rate. Line speed is the rate at which signals are transmitted over a communications line.

During Use

Line state

The state of the communications line can be either ON or OFF. See Section 2.7.3 for an explanation of states and substates.

Seldom Changed

Segment 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 segment buffers are used for transmitting DECnet messages. The segment buffer size defines the amount of information that a node can process from the buffer at one time. Do not change the segment buffer size unless you are very familiar with DECnet concepts.

Maximum number of buffers

This parameter defines the maximum number of buffers available for each node. Increasing the number can increase performance. However, it also uses more memory space.

Access control information

Access control information is security information that allows you to access a specific remote node. This information includes node name, user name, password, and accounting information.

2.9 NCP Command Summary

The remainder of this chapter is devoted to descriptions of the individual NCP commands and their parameters. The commands are discussed in alphabetical order. Table 2–1 lists the NCP commands.

Table 2–1: NCP Commands

Command	Function
CLEAR ACCESS	Clears the incoming access information for the designated user.
CLEAR EXECUTOR	Removes executor node parameters stored on the local node.
CLEAR KNOWN ACCESS	Clears the entire access database.
CLEAR NODE	Removes remote node parameters stored on the local node.
COPY KNOWN NODES	Copies the node database (in DECNODE.DAT) to the specified file. Access information is not copied.
DEFINE CIRCUIT	Defines the circuit characteristics which will take effect the next time you reboot your system.
DEFINE EXECUTOR	Defines executor node characteristics which will take effect with the next system reboot.
DEFINE LINE	Defines the state of the line. It becomes effective with the next system reboot.
EXIT	Allows you to exit from NCP and save parameter changes.
HELP	Displays a summary of NCP commands on your screen.
MONITOR LOGGING	Displays event logging information on the screen in front of you.
QUIT	Allows you to exit from NCP, and gives you the choice of saving parameter changes.
READ LOGGING	Displays the contents of the event logging buffer on the screen in front of you.
SET ACCESS	Sets incoming access information for the designated user.
SET CIRCUIT	Defines or modifies circuit information.
SET ECHO	Displays the contents of a batch file or a redirected file on your screen as the file is executed.

(continued on next page)

Table 2–1 (cont.): NCP Commands

Command	Function
SET EXECUTOR	Defines or modifies the executor node's parameters or characteristics.
SET KNOWN LINKS	Breaks all links and frees the associated sockets.
SET LINE	Defines or modifies the line's parameters or characteristics.
SET LINK	Terminates the logical link connections that are associated with a specific socket. The socket is also detached.
SET NODE	Assigns a node name plus access control information to a unique node address.
SHOW CIRCUIT	Displays circuit information.
SHOW EXECUTOR	Displays executor node information.
SHOW KNOWN ACCESS	Displays incoming access information.
SHOW KNOWN LINKS	Displays all the current logical links.
SHOW LAT	Displays counters for LAT (applies to LAT line only).
SHOW LINE	Displays line information.
SHOW LINK	Displays logical link status information.
SHOW NODE	Displays remote node information.
ZERO CIRCUIT	Zeroes the counters for the circuit.
ZERO EXECUTOR	Zeroes the counters for the executor node.
ZERO LAT	Zeroes counters for LAT (applies to LAT line only).
ZERO LINE	Zeroes the counters for the line.

2.9.1 Standard Definitions

Table 2–2 lists variables and parameters that are frequently used in NCP commands. They are described here to avoid needless repetition in the command descriptions on the following pages.

Table 2–2: NCP Command Parameters

Parameter	Function
circuit-id	specifies circuit ASYNC–1 or ETHER–1. Since there is only one circuit to use for each configuration in both DECnet–DOS V1.0 and DECnet–DOS V1.1, the *circuit-id* is optional.
device-id	specifies the device for controlling the DECnet line (if you are using asynchronous DECnet). For the Rainbow, the device is COM–1. For the IBM PCs, the device can be either COM–1 or COM–2.
line-id	specifies line ASYNC–1 or ETHER–1. Since there is only one line to use for each configuration in both DECnet–DOS V1.0 and DECnet–DOS V1.1, the *line-id* is optional.
node-id	specifies a node address (see *node address*) or a node name (see *node name*).
node address	specifies a unique decimal string consisting of an area number in the range of 1 to 63, followed by a period and a node number in the range of 1 to 1023.
node name	specifies a string of 1 to 6 alphanumeric characters, including at least one alphabetic character.
CHARACTERISTICS	displays parameters that are currently set for the executor, line, or circuit.
COUNTERS	displays error and traffic information for the executor, line, or circuit. A **counter** displays statistics about the flow of network messages for your node. The counters also tabulate recorded error conditions. **Traffic** refers to messages traveling over the line between nodes.
STATUS	displays the availability of the executor, line, or circuit for network activity.
SUMMARY	displays only the most useful information for the executor, line, circuit, or links. SUMMARY is the default display type.
TO *file-id*	specifies the name of an output file to which SHOW or READ information is to be directed.

2.9.2 CLEAR ACCESS

The CLEAR ACCESS command removes incoming access information for the designated user. The access information is stored on a per user basis, and it includes the password and access type.

Format

CLEAR ACCESS USER *username*

where

username is a 1 to 39 alphabetic character string that defines the user.

Example

The following example removes the entire entry for the user BARKER.

```
CLEAR ACCESS USER BARKER(RET)
```

2.9.3 CLEAR KNOWN ACCESS

The CLEAR KNOWN ACCESS command clears all of the incoming access information in the database. This allows you to delete all access information for all users, rather than clearing it for one designated user (as with CLEAR ACCESS).

Format

CLEAR KNOWN ACCESS

Example

```
NCP>CLEAR KNOWN ACCESS(RET)
```

2.9.4 CLEAR EXECUTOR

The CLEAR EXECUTOR command removes specified executor node parameters from the local node's database.

Format

CLEAR EXECUTOR { RECEIVE PASSWORD / TRANSMIT PASSWORD }

where

RECEIVE PASSWORD — removes the password the executor node must receive from the adjacent node in order to exchange messages with the adjacent node.

TRANSMIT PASSWORD — removes the password the executor node must transmit to the adjacent node in order to exchange messages with the adjacent node.

NOTE

These passwords are used for asynchronous DDCMP connections only.

Remarks

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 communications line is restarted (from either node) due to errors.

Example

```
NCP>CLEAR EXECUTOR TRANSMIT PASSWORD(RET)
```

This 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.

2.9.5 CLEAR NODE

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

Format

CLEAR KNOWN NODES

or

CLEAR NODE *node-id* [ALL | USER | PASSWORD | ACCOUNT | LAT–HOST]

where

KNOWN NODES	removes node parameters (node name, node address, user ID, password and account) for all nodes known to the local node.
NODE *node-id*	specifies the node whose parameters are to be removed.
ALL	removes all parameters (node name/address and access control information) for the specified node. If you specify ALL, you cannot include any other parameters.
USER	removes the username, password, and accounting information for the specified node.
PASSWORD	removes the password associated with the username.
ACCOUNT	removes the accounting information for the specified node.
LAT–HOST	removes the indicator that identifies the node as a LAT host.

Example

```
NCP>CLEAR NODE PARIS ALL(RET)
```

This command removes all parameters for node PARIS from the local node's database.

2.9.6 COPY KNOWN NODES

The COPY KNOWN NODES command copies the node database to a file. If the database contains access information, it is not copied. This command is useful if you want to copy your node database for another user, but you do not want to include any of the access information.

Format

COPY KNOWN NODES TO *file-id*

where

TO *file-id* see definition in Section 2.9.1.

Example

```
NCP>COPY KNOWN NODES TO NODEDATA.DAT(RET)
```

This example copies the node database to the file NODEDATA.DAT.

2.9.7 DEFINE CIRCUIT

The DEFINE CIRCUIT command defines the characteristics for the circuit state. The state can either be ON or OFF. The characteristic you specify does not take effect until the next time you reboot your system.

Format

DEFINE CIRCUIT STATE {ON / OFF}

Example

```
NCP>DEFINE CIRCUIT STATE ON[RET]
```

This example sets the circuit state to be ON for the next reboot.

2.9.8 DEFINE EXECUTOR

The DEFINE EXECUTOR command sets executor characteristics that become effective after the next system reboot.

Format

DEFINE EXECUTOR

ADDRESS *node-address*
MAXIMUM BUFFERS *number*
MAXIMUM LINKS *number*
NAME *node-name*
SEGMENT BUFFER SIZE *number*

where

ADDRESS *node-address*
: specifies the address assigned to the executor node.

MAXIMUM BUFFERS *number*
: specifies the maximum number of large network communication buffers available for use by DECnet–DOS. The default is six times (6X) the number of links. The actual number of buffers is restricted by available memory. This affects the network response time and memory usage.

MAXIMUM LINKS *number*
: specifies the maximum number of active logical links for the executor node. The valid range is 1 to 32.

NAME *node-name*
: specifies the node name for the executor node.

SEGMENT BUFFER SIZE *number*
: specifies the size of the segment buffers in bytes. These buffers are used for transmitting DECnet messages off the local area network only. (The segment size on the local area network is fixed.) The size is a decimal integer in the range of 1 to 65535.

Examples

```
NCP>DEFINE EXECUTOR ADDRESS 2.197(RET)
```

This command sets the executor address to 2.197.

```
NCP>DEFINE EXECUTOR ADDRESS 2.197 NAME MOTHER(RET)
```

This command sets the executor address to 2.197 and assigns the executor name to be MOTHER.

```
NCP>DEFINE EXECUTOR MAXIMUM LINKS 6(RET)
```

This command sets the maximum number of logical links for the executor node to 6.

2.9.9 DEFINE LINE

The DEFINE LINE command defines the state of the line at DECnet startup. It becomes effective with the next system reboot.

Format

DEFINE LINE STATE {ON / OFF}
RECEIVE BUFFERS *n*

where

STATE — defines the line's reboot state to ON or OFF. When the line state is ON, the line is available for normal use. When the line state is OFF, the line's database and parameters are present, but the line is not available for any type of network activity.

RECEIVE BUFFERS *n* — specifies the number of buffers to be used for receiving information. Increasing the number of buffers affects the network response time and memory usage (for use only in Ethernet communications). The valid range is 1 to 32. The default is 8.

Examples

```
NCP>DEFINE LINE STATE ON(RET)
```

This example defines the line state to be ON. This causes the line to be ready for use at next reboot.

```
NCP>DEFINE LINE RECEIVE BUFFERS 50(RET)
```

This example sets the maximum number of receive buffers to use for Ethernet communication to 50.

2.9.10 EXIT

The EXIT command lets you exit from NCP and save any parameter changes that you made with the SET and/or DEFINE commands. These changes are saved in the DECPARM.DAT file.

Format

EXIT

Example

```
NCP>EXIT RET

E>
```

This example exits you from NCP and returns you to the prompt or system you were using previously.

2.9.11 HELP

The HELP command displays information about NCP commands on your screen. See Section 2.4 for full details and examples.

Format

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

where

command-verb specifies the NCP command for which you would like more information.

entity specifies that you would like information about the specified command as it relates to this component.

Example

```
NCP>HELP CLEAR EXECUTOR RET
```

This command causes HELP information to be displayed for the CLEAR EXECUTOR command.

2.9.12 MONITOR LOGGING

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 a network support person if network problems occur. Refer to READ LOGGING for more information on event recording.

You can stop the display of information logging by pressing any key on the keyboard. (The logging will also stop when there are no more events to be displayed.) If you want to stop the display and also exit from NCP, type CTRL/C. The system requests verification with the prompt, "Terminating process?". If you answer YES, NCP exits and returns you to the prompt or system you were using previously. If you answer NO, monitoring continues.

For DECnet–DOS systems, event logging is always turned on, and you cannot select only certain events to be displayed. After the events are displayed or redirected, they are then discarded.

If an event is lost, NCP displays an error message before displaying the rest of the logged events:

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

Each event can display the following information:

```
Event type nn.nn     Name of event type
Occurred dd-mmm-yyyy   hh:mm:ss
Other event-dependent text
.
.
.
```

Format

MONITOR LOGGING [TO *file-id*]

where

TO *file-id* see definition in Section 2.9.1.

Example

NCP>MONITOR LOGGING (RET)

The system responds with:

```
Events logged as of 17-Jan-1986 11:17:41
Event type 4.10 Circuit up
Occurred 17-Jan-1986 11:18:00
Adjacent Node: 55.2 (LOWEND)

Event type 4.7 Circuit down -circuit fault
Occurred 17-Jan-1986 11:18:09
Adjacent Node: 55.2 (LOWEND)
Reason: Line synchronization lost

Event type 4.10 Circuit up
Occurred 17-Jan-1986 11:18:15
Adjacent Node: 55.2 (LOWEND)
.
.
.
```

2.9.13 QUIT

The QUIT command allows you to exit from NCP. At the same time, you have the option of either saving or deleting any parameter changes you made.

Format

QUIT

Remarks

If you issue the QUIT command and you have made any changes that may affect the database, NCP requests verification. For example:

```
NCP>QUIT(RET)

Do you want to save changes to permanent database   (Y or N) ?
```

If you answer YES, NCP saves the changes and stores them in the database file. NCP then exits. If you answer NO, the changes are not saved and NCP simply exits.

Example

```
NCP>QUIT(RET)

C>
```

In this example, no changes were made or saved, so NCP exits and returns you to the prompt or system you were using previously.

2.9.14 READ LOGGING

The READ LOGGING command displays the contents of the event logging buffer. It may be redirected to a specified output file.

Format

READ LOGGING [TO *file-id*]

where

TO *file-id* see definition in Section 2.9.1.

Example

```
NCP>READ LOGGING(RET)
```

The system responds with:

```
Events logged as of 17-Jan-1986 11:17:41
Event type 4.10 Circuit up
Occurred 17-Jan-1986 11:18:00
Adjacent Node: 55.2 (LOWEND)

Event type 4.7 Circuit down -circuit fault
Occurred 17-Jan-1986 11:18:09
Adjacent Node: 55.2 (LOWEND)
Reason: Line synchronization lost

Event type 4.10 Circuit up
Occurred 17-Jan-1986 11:18:15
Adjacent Node: 55.2 (LOWEND)
End of log
```

2.9.15 SET ACCESS

The SET ACCESS command sets incoming access information for the designated user. The access information is stored on a per user basis, and it includes the password and access type.

If you want the characters representing the access information to be stored in uppercase letters, simply type the characters. If you do not want the characters to be stored as uppercase, place the string in quotation marks. For example:

abc will be stored in the database as **ABC**

"Aab" will be stored in the database as **Aab**

Format

SET ACCESS USER *username*
[PASSWORD *password*
access-type]

where

username is a 1 to 39 alphabetic character string that defines the user.

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

access-type describes the type of privilege the user has to access information on this node. There are four types of access. They are:

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

Remarks

The information you define with SET ACCESS is stored in the database DECACC.DAT. The DECACC.DAT file is used by FAL and other DECnet–DOS utilities to limit access from remote systems. This file is also used by the *dnet_getacc ( )* function. (Refer to the *DECnet–DOS Programmer's Reference Manual* for information on using the *dnet* functions.)

Example

The following example sets access information for the user BARKER. In this case, the password is JUMPER and the access type is READ.

```
NCP>SET ACCESS USER BARKER PASSWORD JUMPER READ
```

2.9.16 SET CIRCUIT

The SET CIRCUIT command defines or modifies circuit information.

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

Format

SET CIRCUIT [*circuit-id*] { HELLO TIMER *seconds* | OWNER {DECnet / MOP} | SERVICE {DISABLED / ENABLED} | STATE {ON / OFF} }

where

CIRCUIT [*circuit-id*]	specifies the circuit (either ASYNC–1 or ETHER–1) for which parameters are to be defined or modified.
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.
OWNER	specifies the circuit owner.
DECnet	is the default setting.
MOP	is required for running loopback tests via the Network Test Utility. (Refer to Chapter 7 for details on NTU.)
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.
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

```
NCP>SET CIRCUIT ASYNC-1 OWNER DECnet RET
```

This command identifies the owner of circuit ASYNC–1 as DECnet.

2.9.17 SET ECHO

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

Format

SET ECHO {ON / OFF}

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.

Example

```
NCP>SET ECHO ON(RET)
```

This command causes the redirected input from a file to be displayed on the screen.

2.9.18 SET EXECUTOR

The SET EXECUTOR command sets the executor node's parameters or characteristics.

Format

SET EXECUTOR

ADDRESS *node-address*
DELAY FACTOR *number*
DELAY WEIGHT *number*
IDENTIFICATION *id-string*
INACTIVITY TIMER *seconds*
INCOMING TIMER *seconds*
NAME *node-name*
OUTGOING TIMER *seconds*
RECEIVE PASSWORD *password*
RECEIVE PIPE QUOTA *number*
RETRANSMIT FACTOR *number*
SEGMENT BUFFER SIZE *number*
STATE ON
TRANSMIT PASSWORD *password*
TRANSMIT PIPE QUOTA *number*

where

ADDRESS *node-address*

specifies the address assigned to the executor node. You can set ADDRESS only when 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.

DELAY FACTOR *number*

determines the amount of time to wait before a message is retransmitted to a node. The number is decimal in the range of 1 to 65535. The default is 2.

DELAY WEIGHT *number*

is 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 number is decimal in the range 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 quotation marks (") to delimit any string containing blanks or tabs. You cannot use a single quotation mark within the identification string.

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

INACTIVITY TIMER *seconds*

is the maximum duration of inactivity (no data in either direction) on a logical link before the node checks to see if the logical link still works. If no activity occurs within the minimum number of seconds, the node generates traffic to test the link. The value range is 1 to 65535. The default is 30.

INCOMING TIMER *seconds*

is 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 default is 45.

NAME *node-name*

specifies the node name for the executor node. You can set NAME only when 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.

OUTGOING TIMER *seconds*

is 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 default is 60.

RECEIVE PASSWORD *password*

is the password the local node must receive from the adjacent node in order to exchange messages with the adjacent node.

RECEIVE PIPE QUOTA *number*

the number of segments the driver will allow to be received for a link by flow control. The valid range is 1 to 32767. The default is 3.

RETRANSMIT FACTOR *number*

is the maximum number of times that the executor node will restart 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 3.

SEGMENT BUFFER SIZE *number*

specifies the size of the segment buffers in bytes. These buffers are used for transmitting DECnet messages off the local area network only. (The segment size on the local area network is fixed.) The size is a decimal integer in the range of 1 to 65535.

STATE ON

allows 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. You must SET EXECUTOR ADDRESS (and NAME, if desired) before you can SET EXECUTOR STATE ON. If you specify STATE ON, you cannot include any other parameters.

TRANSMIT PASSWORD *password*

is the password the local node must transmit to the adjacent node in order to exchange messages with the adjacent node.

TRANSMIT PIPE QUOTA *number*

the number of segments the driver will buffer into NSP before it blocks the user on a transmission. The valid range is 1 to 32767. The default is 3.

NOTE

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

Examples

```
NCP>SET EXECUTOR STATE ON(RET)
```

This 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 OUTGOING TIMER 10(RET)
```

This command indicates that the executor node will wait 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 ADDRESS 4.4 NAME FOO(RET)
```

This command sets the executor node's address to 4.4 and its name to FOO.

2.9.19 SET KNOWN (or ACTIVE) LINKS

The SET KNOWN (or ACTIVE) LINKS command breaks all link connections that are not set with the socket option, SO__KEEPALIVE, and frees the associated sockets.

Format

SET {KNOWN LINKS / ACTIVE LINKS} STATE OFF

where

KNOWN LINKS	breaks all current links whose sockets are not set with SO__KEEPALIVE. The sockets can be reassigned to other tasks.
ACTIVE LINKS	performs the same function as KNOWN LINKS.
STATE OFF	turns the logical link off.

Remarks

Use this command only if you are familiar with the socket level programming interface. Refer to the *DECnet–DOS Programmer's Reference Manual* for additional information.

Example

```
NCP>SET ACTIVE LINKS STATE OFF(RET)
```

This command aborts all current links and frees the associated sockets.

2.9.20 SET LINE

The SET LINE command defines or modifies line information.

Most parameters, such as the baud rate, are set by the person who coordinates activity on your network. Therefore, you must obtain this information from that person.

Restrictions

The executor state must be ON in order for you to define or modify any line parameters. In addition, the line state must be OFF if you wish 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.

Format

SET LINE [*line-id*] { DEVICE {COM–1 / COM–2} / MODEM {NULL / FULL} / SPEED *baud-rate* / STATE {OFF / ON} }

where

LINE [*line-id*] specifies either ASYNC–1 or ETHER–1 as the line for which parameters are to be created or modified.

SPEED *baud-rate* sets the DDCMP line speed to one of the following:

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

A line speed of 7200 is not supported by the Rainbow 100A and B computer systems.

STATE sets the line's operational state to ON or OFF.

DEVICE *device-id* specifies the device for controlling the line. The line's circuit must be in the OFF state. For the Rainbow personal computers, the device is COM–1. For the IBM personal computers, the device can be either COM–1 or COM–2.

MODEM — specifies the type of modem control to be used. Note that modem control is used only 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.

NOTE

Information about modem and line speed applies to asynchronous DDCMP communications only.

Example

```
NCP>SET LINE ASYNC-1 SPEED 9600 RET
```

This command sets the speed of line ASYNC–1 to 9600. Note that line speed is set only if you are using asynchronous DDCMP connections. (Since there is only one line to use for each configuration in both DECnet–DOS V1.0 and DECnet–DOS V1.1, the *line-id* is optional.)

2.9.21 SET LINK

The SET LINK command terminates logical link connections that are associated with a specific socket. The socket is also detached.

Format

SET LINK *socket* STATE OFF

where

LINK *socket* specifies the socket to be detached.

STATE OFF turns the logical link off.

Remarks

Use this command only if you are familiar with the socket level programming interface. If the socket option SO__KEEPALIVE has been set, the socket cannot be terminated. Refer to the *DECnet–DOS Programmer's Reference Manual* for additional information.

Example

```
NCP>SET LINK 3 STATE OFF(RET)
```

This command detaches socket 3 and breaks its logical links. To determine the socket number, use the SHOW KNOWN LINKS command.

2.9.22 SET NODE

The SET NODE command assigns a node name plus access control information to a unique node address. If you want the characters representing the access information to be stored in uppercase letters, simply type the characters. If you do not want the characters to be stored as uppercase, place the string in quotation marks. For example:

abc will be stored in the database as **ABC**

"Aab" will be stored in the database as **Aab**

Format

SET { NODE *node-address* NAME *node-name* [*access info*][LAT–HOST] / KNOWN NODES FROM *node-id* / NODE *node-id* FROM *node-id* }

where

NODE *node-address* — specifies the address of the node to be defined.

NAME *node-name* — specifies the name to be associated with the node address.

access info — specifies access control information for the remote node in the following format:

USER *username* [PASSWORD *password*] [ACCOUNT *account*]

where

username 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 may be required to access files or programs on the remote node.

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

See Sections 2.6 through 2.6.3 for more details on specifying access control information.

LAT–HOST — indicates that the node is a LAT host.

KNOWN NODES FROM *node-id*

specifies a remote node from which to copy the complete database of node names and node addresses to the local node. This set of node names will not include any access control information.

NOTE

If the database on the remote node is a large one (for example, in excess of 100 nodes), this can significantly slow down the access time to the database.

NODE *node-id*
FROM *node-id*

specifies a single node whose name and address are to be copied from the specified remote node into the local database. No access control information will be copied.

Examples

```
NCP>SET NODE 2.65 NAME MADRID USER MAGIC PASSWORD WAND(RET)
```

This command assigns the node name MADRID to the node at address 2.65. It also specifies the username as MAGIC and the password as WAND.

```
NCP>SET NODE YODA FROM R2D2(RET)
```

This command allows you to copy the node name YODA and its node address from the specified remote node R2D2.

2.9.23 SHOW CIRCUIT

The SHOW CIRCUIT command displays all the information you set with the SET CIRCUIT command, as well as certain system information. You can either display the information on your screen or redirect it to an output file at the local node.

Format

SHOW CIRCUIT [*circuit–id*] [CHARACTERISTICS / COUNTERS / STATUS / SUMMARY] [TO *file-id*]

where

CIRCUIT [*circuit-id*] — specifies the circuit (either ASYNC–1 or ETHER–1) for which information is to be displayed.

CHARACTERISTICS
COUNTERS
STATUS
SUMMARY
TO *file-id*

— see definitions in Section 2.9.1.

Remarks

Circuit counters monitor traffic and errors on individual circuits. The network coordinator periodically checks each circuit's counters to assess circuit performance and determine potential problems.

Examples

```
NCP>SHOW CIRCUIT ASYNC-1 STATUS RET
```

This command displays the current status of circuit ASYNC–1.

```
NCP>SHOW CIRCUIT COUNTERS RET
```

This command displays the current counters for the circuit.

```
NCP>SHOW CIRCUIT ETHER-1 STATUS TO CIR.DAT RET
```

This command redirects the current status information about circuit ETHER–1 to an output file named CIR.DAT. (Since there is only one circuit to use for each configuration in both DECnet–DOS V1.0 and DECnet–DOS V1.1, the *circuit-id* is optional.)

2.9.24 SHOW EXECUTOR

The SHOW EXECUTOR command displays the executor node information. You can either display the information on your screen, or redirect it to an output file at the local node.

Format

SHOW EXECUTOR [CHARACTERISTICS | COUNTERS | STATUS | SUMMARY] [TO *file-id*]

where

CHARACTERISTICS — see definitions in Section 2.9.1.
COUNTERS
STATUS
SUMMARY
TO *file-id*

Examples

```
NCP>SHOW EXECUTOR STATUS TO STAT.NOD
```

This command redirects the status information for the executor node to an output file named STAT.NOD.

```
NCP>SHOW EXECUTOR COUNTERS
```

This command displays the counters for the executor node.

2.9.25 SHOW KNOWN ACCESS

The SHOW KNOWN ACCESS command displays the contents of the incoming access information database, which is stored in the file DECACC.DAT. If a password is part of the database entry, it only appears on the screen as three dots (...).

Format

SHOW KNOWN ACCESS

Example

The following example displays the incoming access information for the users BENCE, FOO, and MATAHARI.

```
NCP>SHOW KNOWN ACCESS(RET)
Access      User/
Type        Password

RO          BENCE/...
ALL         FOO/...
NONE        MATAHARI
```

2.9.26 SHOW LAT

The SHOW LAT command displays information about the LAT line you may be using. The information may be directed to a file. This command only works if you have LAT installed on your node.

Format

SHOW LAT [COUNTERS] [TO *file-id*]

where

COUNTERS — see definitions in Section 2.9.1.
TO *file-id*

Example

The following command displays the counter information for the LAT line.

```
NCP>SHOW LAT COUNTERS(RET)
```

2.9.27 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.

Format

SHOW LINE [*line-id*] [CHARACTERISTICS / COUNTERS / STATUS / SUMMARY] [TO *file-id*]

where

LINE *line-id* — specifies either ASYNC–1 or ETHER–1 as the line for which information is to be displayed.

CHARACTERISTICS
COUNTERS
STATUS
SUMMARY — see definitions in Section 2.9.1.

TO *file-id*

Remarks

Line counters monitor procedures on individual lines. The network coordinator periodically checks each line's counters to assess line performance and determine potential problems.

Example

```
NCP>SHOW LINE ASYNC-1 COUNTERS TO STORE.LIN(RET)
```

This command redirects counter information about line ASYNC–1 to an output file named STORE.LIN.

2.9.28 SHOW LINK

The SHOW LINK command displays summary type information (see definition in Section 2.9.1) for the specified logical links.

Format

SHOW {KNOWN LINKS / ACTIVE LINKS}

where

KNOWN LINKS — displays information for all logical links that are known to the local node.

ACTIVE LINKS — displays information for all logical links that are known to the local node.

Examples

```
NCP>SHOW KNOWN LINKS(RET)

Known Link Status as of 06-Jan-1986 12:44:35

State      Socket      Node      Local      Remote      Local          Remote
                               Addr       Addr        # Name         # Name

Running    1           4.333     25824      34          0 LLA25824     42
Running    2           4.24      25857      17420       0 LLA25847     42
Idle       3           4.333     25762      25          0 LLA25762     42
Running    4           4.209     25891      2080        0 LLA25891     42
```

This example displays summary information (by default) for all logical links known to the local node.

```
NCP>SHOW ACTIVE LINKS(RET)

Active Links Status as of 06-Jan-1986 12:44:43

State      Socket      Node      Local      Remote      Local          Remote
                               Addr       Addr        # Name         # Name

Running    1           4.333     25824      34          0 LLA25824     42
Running    2           4.24      25857      17420       0 LLA25857     42
Running    3           4.209     25891      2080        0 LLA25891     42
```

This example displays summary information (by default) for all active logical links known to the local node.

2.9.29 SHOW NODE

The SHOW NODE command displays remote node information. You can either display the information on your screen, or redirect it to an output file at the local node.

Format

SHOW NODE *node-id* [STATUS | SUMMARY] [TO *file-id*]

or

SHOW {KNOWN NODES | ACTIVE NEEDS | ADJACENT NODE}

where

KNOWN NODES	displays information for all nodes that you defined with the SET NODE command, in other words, nodes that are **known** to the local node.
ACTIVE NODES	displays information for all active nodes. Information is displayed for the executor node, your adjacent node, and all nodes connected to the executor node by logical link(s).
ADJACENT NODE	displays information for the adjacent DDCMP node only.
STATUS SUMMARY	see definitions in Section 2.9.1. Can be specified for SHOW NODE *node-id* only; defaults to a summary display for other SHOW NODE commands.
TO *file-id*	see definition in Section 2.9.1.

Examples

```
NCP>SHOW NODE SOLO STATUS(RET)
```

This command displays status information for node SOLO.

```
NCP>SHOW ACTIVE NODES(RET)
```

This command displays summary information (by default) for all nodes that are reachable to your node.

```
NCP>SHOW ADJACENT NODE TO ADJNOD.STA(RET)
```

This command redirects summary information about the adjacent node to an output file named ADJNOD.STA.

2.9.30 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 or ETHER–1 as the circuit for which counters are to be zeroed.

Example

```
NCP>ZERO CIRCUIT ETHER-1(RET)
```

This command zeroes the counters for circuit ETHER–1.

2.9.31 ZERO EXECUTOR

The ZERO EXECUTOR command zeroes the counters associated with and maintained on the executor node.

Format

ZERO EXECUTOR [COUNTERS]

2.9.32 ZERO LAT

The ZERO LAT command zeroes the counters for the LAT line. This command only works if you have LAT installed on your node.

Format

ZERO LAT [COUNTERS]

Example

```
NCP>ZERO LAT COUNTERS(RET)
```

2.9.33 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 or ETHER–1 as the line for which counters are to be zeroed.

Example

```
NCP>ZERO LINE ASYNC-1 RET
```

This command zeroes the counters for line ASYNC–1.

3
Accessing Local and Remote Files

The Network File Transfer (NFT) utility is a network program that allows you to access remote files. After you install the DECnet–DOS software, you can use NFT to:

- Append two or more files.
- Copy files between the local and remote nodes.
- Delete local and remote files.
- List files located on a local or remote directory.
- Define and display access control information.
- Run batch (command) files on remote nodes.
- Display the contents of a local or remote file on your screen.
- Print files on remote printers.

This chapter describes:

- Running NFT (see Section 3.1).
- Specifying access control information (see Section 3.2).
- File Specifications (see Section 3.3).
- Types of file systems (see Section 3.4).
- Typing NFT commands (see Section 3.5).
- Getting help (see section 3.6).
- File operations (see Section 3.7).

- The NFT commands and their syntax in alphabetical order (see Section 3.8).
- How to provide remote file access to your node (see Section 3.9).
- How to set up your computer to act as a server for multiple service functions (see Section 3.10).

3.1 Running the NFT Utility

You perform NFT functions by using one of the following methods:

- Single command method:

```
E>NFT command(RET)
E>
```

  You can also use a batch file to start NFT. This command file can contain the NFT commands. For example:

```
E>NFT <COMMANDS.DAT(RET)
```

- Multiple command method:

```
E>NFT(RET)
NFT>command(RET)
NFT>command(RET)
NFT>EXIT(RET)
E>
```

You can also use (CTRL/Z) (RET) to exit from NFT. To do this:

1. Type the letter Z while you hold down the (CTRL) key.
2. Press the (RET) key.

NOTE

When you run NFT in batch mode, an ERRORLEVEL of 0 is returned for success. A 1 is returned for failure.

3.2 Defining Remote Node Access Information

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

- **Username**

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

- **Password**

  A character string consisting of 1 to 39 alphabetic characters.

- **Account**

  A character string consisting of 1 to 39 alphanumeric characters.

You must follow the node name or node address (area.number) with the username, password and account enclosed in quotation marks (''). You must then follow the entire string with a double colon (::). The following is an example of a node name with access control information:

LONDON''SMITH OPEN''::

You can specify default access control information using NCP (see Chapter 2 of this guide), or you can specify the information directly to NFT during an NFT operation. When you specify the information during an NFT operation, NFT temporarily stores the information in a table for use in any following command dealing with the same node.

The access control information you specify during an NFT session is not saved when you exit from NFT. Therefore, if you run NFT at another time, you cannot use any access control information previously specified in any NFT command. However, you can use any access control information you previously specified using NCP. If NCP provides defaults and you specify your own values with NFT, the values you specify are used.

3.2.1 Defining Access Control Information with NFT

There are two ways to specify access control information during an NFT operation:

- By accessing a remote node for the first time.
- By using the SET command.

NFT temporarily stores access control information for up to ten nodes.

In the following example, you access a remote node by requesting a directory listing of files located on the VMS node LONDON, username SMITH, password OPEN.

```
NFT>DIRECTORY LONDON"SMITH OPEN"::(RET)
```

NFT displays the list of files located in the requested directory, including the block size of each file, and the time and date the file was last modified or created.

```
Directory of: LONDON"SMITH password"::SYS$SYSROOT:[SMITH]

FILE1.TXT;1       12        01-MAR-85        16:01:51
FILE2.TXT;1       34        02-SEP-83        14:20:35
FILE3.TXT;1       22        09-JUL-77        12:15:22

NFT>
```

NFT stores the access control information for node LONDON in its temporary table.

You use the SET command to specify and then save a username, password and account as well as specific disk and/or directory information. In the following example, you specify the username and password for node MADRID. In addition, you specify the name of the subdirectory ARENA.FILES.

```
NFT>SET MADRID"ARENA BULL"::[ARENA.FILES]RET
```

NFT stores the access information for node MADRID in its temporary table.

3.2.2 Using Default Access Control Information

Once you specify access control information using NFT, you can use the node name alone in subsequent file operations during the same NFT session. NFT uses the access control information you previously specified for the node name. This is the **default** access control information.

For example, if you request a directory listing of node LONDON using only the node name, NFT displays the same list of files as the first time you accessed the node. When you type a node name alone, you must follow the name with a double colon. For example:

```
NFT>DIRECTORY LONDON::RET

Directory of: LONDON"SMITH password"::SYS$SYSROOT:[SMITH]

FILE1.TXT;1      12       01-MAR-85        16:01:51
FILE2.TXT;1      34       02-SEP-83        14:20:35
FILE3.TXT;1      22       09-JUL-77        12:15:22

NFT>
```

When you type a node name alone, NFT follows this procedure to determine the access control information:

1. First NFT checks to determine if you specified the access control information earlier in the same NFT session. If you did, that access control information is used.
2. If you did not specify it earlier in the same session, NFT checks to determine if you specified it using NCP. If you did, that access control information is used.
3. If you did not specify access control information either using NCP or earlier in the same NFT session, NFT assumes that the remote node does not require the information and attempts the current operation. (For information about access control, refer to Chapter 2.)

3.2.3 Changing the Default Access Control Information

You can change the default access control information by typing different access control information for the same node name in an NFT operation. In this case, NFT replaces the original access control information in its temporary table with the new information.

For example, if you request a directory listing of files on remote node LONDON, but specify a different username and password, NFT displays a list of files for the new user:

```
NFT>DIRECTORY LONDON"DOC PEN"::RET

Directory of: LONDON"DOC password"::SYS$SYSROOT:[DOC]

ABC.TXT;1          10          07-SEP-84          12:10:49
DEF.TXT;1          29          12-JAN-84          15:20:32
GHI.TXT;1          47          03-MAR-84          14:10:05

NFT>
```

Now if you request a directory listing using the node name alone, NFT displays this new list of files associated with node LONDON.

```
NFT>DIRECTORY LONDON::RET

Directory of: LONDON"DOC password"::SYS$SYSROOT:[DOC]

ABC.TXT;1          10          07-SEP-84          12:10:49
DEF.TXT;1          29          12-JAN-84          15:20:32
GHI.TXT;1          47          03-MAR-84          14:10:05

NFT>
```

You can also change the access control information using the SET command. In fact, when you use the SET command to specify default access control information, you can only replace the information in the NFT table using another SET command. For example, to change the access control information for node MADRID, type:

```
NFT>SET MADRID"SPANISH DANCE"::RET
```

Now if you request a directory listing using the node name alone, NFT displays the new list of files associated with node MADRID.

```
NFT>DIRECTORY/BRIEF MADRID::RET

Directory of: MADRID"SPANISH password"::SYS$SYSROOT:[SPANISH]

LAW.DOC;1          RULE.DOC;2

NFT>
```

3.2.4 Displaying Access Control Information

The SHOW command displays access control information that you specify during an NFT session. For example:

```
NFT>SHOWRET

LONDON"DOC password"::
MADRID"SPANISH password"::
```

3.3 File Specifications

A complete file name is called a **file specification**. A file specification provides the computer system with all the information it requires to identify a unique file. Each operating system in the network has its own set of rules for naming files. For example, when you name a local file, you must follow the standard DOS operating system format:

- A **drive name**. (This is optional if you are using the default drive name.)
- A **path name**. (This is optional.)
- A **file name** of up to eight alphanumeric characters.
- A **file type** of up to three alphabetic characters, separated from the file name with a period. (This is optional.)

The example below specifies a file named SURVEY.CRD. The file is located on drive A in the subdirectory SUE, which is located in the subdirectory USERS:

```
A:\USERS\SUE\SURVEY.CRD
```

When you access a file on a remote node, you must use a file specification that conforms to the conventions required by the remote node. For example, some operating systems require a version number as part of the file specification:

```
LONDON::CHAP1.DOC;2
```

Example 3–1 shows how to copy a local DOS file called EMPLOY.LST to a remote VMS node named BATH. The file is stored on the remote node as NAMES.EMP;1 (Version 1).

```
NFT>COPY A:EMPLOY.LST BATH``IRON BOATS''::SCRB:[IRON]NAMES.EMP;1(RET)
```

Local DOS file specification	Node name with access control information	Remote VMS file specification

Example 3–1: Copying a File to a Remote Node

3.3.1 Specifying Remote Files

DECnet–DOS can copy files to and from nodes running different operating systems. Refer to Table 3–1 for a list of file specifications that you can use with different operating systems.

Table 3–1: Operating Systems and File Specifications

Operating System	File Specification
VAX/VMS	dev:[directory]filename.typ;ver
TOPS–20	dev:[directory]filename.typ.ver
TOPS–10	dev:[directory]filename.typ
RSX	dev:[DIRECTORY]filename.typ;ver
P/OS	dev:[DIRECTORY]filename.typ;ver
DOS	dev:dir\dir\..filename.typ
ULTRIX–32	dir/dir/.../filename.typ
RSTS/E	dev:[DIRECTORY]filename.typ

NOTE

Most systems accept square brackets [ ] or angle brackets < > to delimit a directory name, and a period (.) or a semi-colon (;) to delimit version. NFT accepts all of these delimiters.

If any portion of the above file specification formats is omitted, NFT assumes a default. File specifications of any other format are **foreign** to DECnet–DOS NFT. When you type a foreign file specification, you must enclose it in quotation marks (''). For example:

```
NFT>COPY BOSTON::"DK2:[100,100]NAMES.DAT"RET
```

This directs NFT to copy the RSTS/E file from the remote node BOSTON to the local node, allowing unacceptable characters such as a comma in the directory name.

Most file specifications are not foreign.

3.3.2 Using Wildcards

Wildcards allow you to specify more than one file at a time. There are three wildcards you can use:

- A question mark (?) or a percent sign (%) matches any single character in the same position that the question mark or the percent sign occupies. For example, you can use TEST?.DOC to specify the following files:

  TEST1.DOC
  TEST2.DOC
  TESTA.DOC
  TESTB.DOC

- An asterisk (*) matches part or all of a file specification. For example, you can use *.DOC to specify the same list of TEST files. In this example, the asterisk replaces any file name with a file type of DOC.

You can also use a combination of wildcards. For example, you can use TEST?.* to specify the following files:

TEST1.DOC
TEST2.DOC
TESTX.TXT
TESTY.TXT
TESTZ.TRY

Note that some remote systems do not support all three wildcards. Also, there may be wildcards which are supported on some remote systems but not supported by NFT.

3.3.3 Using Question Marks, Percent Signs, and Asterisks as Wildcards

Local and remote files can contain question marks, percent signs, or asterisks as wildcards. NFT cannot change the names of the files when wildcards are used. As an example, you can type:

```
COPY TEST?.* BOSTON::*.*
```

If you want to rename the files included in the remote file specification, you cannot use wildcards. For example, you cannot type:

```
COPY TEST?.* BOSTON::FOO?.*
```

Wildcards can also be used with the DELETE, TYPE, and DIRECTORY commands. For example:

```
DELETE LONDON::*.*.*
```

This command deletes all files in the user's default directory on the node LONDON.

3.4 Types of Files and Systems

Files can be one of two data types:

- IMAGE

  An image file is a file whose data is copied without any interpretation or data change. In other words, the file is copied and received as a carbon copy of itself. The most common example of this type of file is an executable file.

- ASCII

  An ASCII file is a file whose records end with a carriage return/line feed (CR/LF) pair.

Systems can be one of two types:

- Stream

  With a stream system, a file is a series of continuous characters.

  Most stream systems including the DOS operating system do not support such attributes as fixed or variable length records. If a file with these attributes is copied to the local system, the attributes are lost.

- Nonstream

  With a nonstream system, file data is in specific record formats. For example, fixed length, stream, variable length, and variable with fixed length control (VFC).

  Fixed length records are all the same size. The size is fixed when you create the file, and you cannot change it.

  Variable length records can be of different lengths, up to a maximum size that you specify. The maximum size is fixed when you create the file, and you cannot change it.

  VFC records include a fixed length control field that precedes the variable length data. This format allows you to add data that labels the contents of the variable length portion of the record.

Refer to Table 3–2 for the changes that occur when you copy a file from a remote node.

Table 3–2: Copying Files from a Remote Node

Data Type	System Type	Record Attributes	How Stored
IMAGE	Not applicable	Ignored	As received from the remote node.
ASCII	Stream	Ignored	As received from the remote node with embedded carriage control.
ASCII	Nonstream	Other than implied CR/LF, PRN or FTN	As received from the remote node.
ASCII	Nonstream	Implied CR/LF pair	CR/LF added to each record.
ASCII	Nonstream	PRN or FTN	Data converted correctly.

NFT converts embedded carriage control characters by default. You can cancel this conversion with the /NOCONVERT switch.

Although NFT can usually determine whether a file is ASCII, you should use the /IMAGE switch when you copy a non-ASCII file to a remote node. Refer to Table 3–3 for the changes that occur when you copy a file to a remote node.

Table 3–3: Copying Files to a Remote Node

Data Type	System Type	Record Attributes	How Sent
ASCII	Stream	None	Unchanged. Records are determined by LFs.
ASCII	Nonstream	Variable, Implied CR/LF	Carriage return/line feed pair is dropped. Records are determined by LFs.
IMAGE	Stream	None	Unchanged.
IMAGE	Nonstream	Fixed: 128 bytes	Unchanged.

3.5 Typing NFT Commands

To type an NFT command, enter it at the NFT command prompt. The NFT commands consist of three parts:

- The command verb and switch
- The source file specification
- The destination file specification

A file specification may be local (no node name included) or remote (node name included).

Example 3–2 defines the parts of an NFT command.

```
NFT>COPY/PRINT FILE1.TXT LONDON::FILE1.TXT;1(RET)
```

verb and switch	source file specification (local)	destination file specification

Example 3–2: An NFT Command

Except for the node name and file specification (or the command verb and switch), you must separate one part of the command from another with a space or tab.

You can abbreviate command verbs and switches to a character string that is unique to that verb. For example, you can type C for the COPY command, or /VA for the /VARIABLE switch. However, for clarity and consistency, all examples in this chapter show the full command syntax.

When using the LF key to continue a command on another line and a space is needed to separate the last word on the first line from the first word on the second line, you must enter the space immediately before or after the LF key.

If you need help with a command or switch, remember to type the HELP command followed by an optional command name or switch.

3.5.1 Command Prompts

In many cases, if you type a command verb alone, the command prompts you for further information. For example, if you want to append one or more files to the end of an existing file and you type the APPEND command verb alone, APPEND prompts you for the required information as shown in the following example:

```
NFT>APPEND RET
File(s)? LONDON``SMITH OPEN''::WRIT:[SMITH]FILB.TXT;3 RET
To? FILA.TXT RET
NFT>
```

Command prompts are discussed in the applicable command descriptions.

3.5.2 Command Switches

Some operating systems store files with attributes that indicate the type of file and the format of the file's records. A DOS file does not include attributes. Therefore, when you copy a file from the local system to another system that stores attributes (such as VAX/VMS), you can use switches to indicate the type of file you are copying and the type of system it is coming from. NFT provides defaults for file attributes. You can use switches to change these defaults.

You can also use switches to modify command verbs. Switches are global in effect and must immediately follow the command verb in the command line. The same switch can have a different effect, depending on whether you are copying a file to or from a remote node.

A valid switch for several NFT commands is /NOLOG. This switch requests that NFT not print a notification line after the APPEND, COPY, DELETE, PRINT, and SUBMIT operations. It is ignored for all other commands.

If you use a valid switch that has no effect on certain commands, such as /NOLOG, NFT does not display a message. However, if you use a switch that NFT does not accept, such as /PRINT with a remote input file, NFT displays a message.

Tables 3–4, 3–5, and 3–6 list valid NFT switches.

Table 3–4: Valid Switches for Copying Files from a Remote Node

Switch	Function
/ASCII	No data interpretation is performed on ASCII file coming from a stream file system. For ASCII files being copied from a nonstream file system, data handling is determined by the remote file system and by the remote file's record format and record attributes. This switch is valid only with the COPY and APPEND commands.
/IMAGE	Requests that NFT copy the file from the remote system as it is (binary data), with no conversion of any kind. This switch is valid only with the COPY and APPEND commands. **NOTE** If neither /ASCII nor /IMAGE is indicated, the information that was passed from the remote node will be used to copy the file. This is also the default.
/BLOCK	Copies files as image mode files, with no regard to record structure. This allows you to move files that have undefined formats or files with very long record lengths. When you indicate /BLOCK, it forces the following attributes: /IMAGE mode, /FIXED length, and /MRS of 512. **NOTE** When you use this switch, do not include any other switches
/NOCONVERT	If the file is copied from a node with an RMS file system, and the record attributes for the file are either FTN (FORTRAN carriage control) or PRN (print file carriage control), the carriage control characters within each logical record are normally expanded before the record is written to the local file. This switch turns that feature off. (For details on /NOCONVERT, refer to the *VAX–11 RMS Reference Manual* or other appropriate system manuals.) This switch is valid only with the COPY and APPEND commands.
/NOLOG	Requests that NFT not print a notification line after the APPEND, COPY, DELETE, PRINT, and SUBMIT operations.

Table 3–5: Valid Switches for Copying Files to a Remote Node

Switch	Function
/ASCII	Tells NFT that the file has records that end, by default, with a carriage return/line feed pair. This switch is valid only with the COPY and APPEND commands. If the first record does not end with a CR/LF pair, an image copy of the file is made.
/IMAGE	Requests that NFT copy the file to the remote system as it is, (binary data) with no conversion of any kind. The default record format is FIXED and the default Maximum Record Size (MRS) is 128 bytes. You can change these defaults with the /VARIABLE (or /VFC) and /MRS switches respectively. The last record can be shorter than the previous records. This switch is valid only with the COPY and APPEND commands. **NOTE** If neither /ASCII nor /IMAGE is indicated, the file is examined. If the first 512 bytes contain CR/LF, the file is copied as ASCII. If the first bytes do not contain CR/LF, the file is copied as an image file.
/FIXED	Indicates that the records within the file are all the same length. By default, the length is 128 bytes, but you can change the length with the /MRS switch. This switch is valid only with the COPY command.
/VARIABLE	If you are copying a file to an RMS file system, this switch resets the record format (RFM) to Variable length records, with a Maximum Record Size of 0 (no maximum). You can change the Maximum Record Size with the /MRS switch. This switch is valid only with the COPY command.
/VFC = *nnn*	If you are copying a file to an RMS file system, this switch sets the record format (RFM) to Variable length with fixed control headers of length *nnn*, where *nnn* = 0 to 255 bytes. The default header size is 2. The first *n* bytes become the header record. If you are copying the file to a stream system, this switch has no effect. **NOTE** If neither /FIXED nor /VFC is indicated, the default is /VARIABLE.
/ALLOCATION = *nnn*	If you are creating a new file on a remote system, this switch requests that the remote system set the allocation quantity (in blocks) for the new file to *nnn*. The default is 0, which causes allocation as needed. This switch is valid only with the COPY command. The valid range is 0 to 2147483647.

(continued on next page)

Table 3–5 (cont.): Valid Switches for Copying Files to a Remote Node

Switch	Function
/BLOCK	Copies files as image mode files, with no regard to record structure. This allows you to move files that have undefined formats or files with very long record lengths. When you indicate /BLOCK, it forces the following attributes: IMAGE mode, FIXED length, and a MRS of 512. **NOTE** When you use this switch, do not include any other switches.
/CC = *xxx*	Sets record attributes for files you are copying to RMS file systems only. If you are copying files to a stream system, such as TOPS–20, the record attributes are ignored. *xxx* represents one of the following: None – No record attributes. FTN – Records contain FORTRAN carriage control. Records that contain FORTRAN carriage control are not native to DOS systems. For details on FTN, refer to the *VAX–11 RMS Reference Manual* or other appropriate system manuals. CR – (Default) Records have an implied carriage return/line feed. PRN – Records contain a fixed header with print carriage control. This is used with the /VFC switch. Records that contain print carriage control are not native to DOS systems. For details on PRN, refer to the *VAX–11 RMS Reference Manual* or other appropriate system manuals. It is assumed that if you specify an explicit record attribute for the output file, the file data already conforms to that specification. In this case, NFT does no conversion. This switch is valid only with the COPY and APPEND commands.
/DELETE	Is valid only with the /PRINT switch. It requests that NFT delete the copied file from the remote node after the file is copied and printed. This switch is valid only with the COPY and APPEND commands.
/LSA	Indicates to a remote node that records are line sequenced ASCII. LSA files are valid for remote systems. They are not native to DOS systems. For details on LSA record attributes, refer to the appropriate TOPS–10 and TOPS–20 DECnet documentation.
/MACY11	Indicates that the file is to be written on the remote node in MACY11 format. This switch is valid only with the APPEND and COPY commands. MACY11 files are only valid for remote systems. They are not native to DOS systems. For details on MACY11 record attributes, refer to the appropriate TOPS–10 and TOPS–20 DECnet documentation.

(continued on next page)

Table 3–5 (cont.): Valid Switches for Copying Files to a Remote Node

Switch	Function
/MRS = *nnn*	Sets the Maximum Record Size, where the *nnn* is 0 to 32767. The default MRS for VARIABLE is 0. This means no maximum for VARIABLE file copies. The default MRS for FIXED files is 128. This switch is valid only with the COPY command.
/NOLOG	Requests that NFT not print a notification line after the APPEND, COPY, DELETE, PRINT, and SUBMIT operations.
/NOSPAN	Requests that records not span blocks. It is valid for RMS file systems only. This switch is valid only with the COPY command.
/PRINT	Allows you to print a file on the remote node's default printer after the file copy is complete. This switch is valid only with the COPY and APPEND commands.
/SUBMIT	Allows you to queue a batch file to be executed at the remote node after the file copy operation is complete. This switch is valid only with the COPY and APPEND commands.

Table 3–6: Valid Switches for the DIRECTORY Command

Switch	Function
/BRIEF	Requests that NFT display file names only. The names are displayed on the screen four to a line.
/FULL	Requests that NFT display complete file information for one or more files. The information for each file includes: the file name; the file size (in blocks); the owner of the file; the date and time the file was last modified or created; the type of file organization, record format, and record attributes; and the type of protection assigned to the file.

3.6 Using the HELP Command

If you need assistance in selecting NFT commands and switches, use the HELP command. Type:

```
NFT>HELP(RET)
```

The system responds with:

```
Help is available on the following commands:

APPEND          COPY          DELETE      DIRECTORY
EXIT            HELP          PRINT       SET
SHOW            SUBMIT        TYPE

switches:

/ALLOCATION     /ASCII        /BLOCK      /BRIEF
/CC             /DELETE       /FIXED      /FULL
/IMAGE          /LSA          /MACY11     /MRS
/NOCONVERT      /NOLOG        /NOSPAN     /PRINT
/SUBMIT         /VARIABLE     /VFC
```

To obtain information about the SHOW command, for example, type:

```
NFT>HELP SHOW(RET)
```

To obtain information about the /ASCII switch, type:

```
NFT>HELP /ASCII(RET)
```

3.7 File Operations

NFT allows you to manipulate files on accessible DECnet nodes. Using NFT, you can perform file operations, such as:

- Appending files.
- Listing files.
- Copying files.
- Deleting files.
- Displaying file names.
- Printing files.
- Typing the contents of a file.
- Submitting files as batch jobs.

3.7.1 Appending Files

To copy one or more files to the end of an existing file, use the APPEND command. You can append the following combination of files:

- A local file to a remote file.
- A remote file to a local file.

For example, to append the remote file FILB.TXT, located on the VAX node LONDON, to the local file FILA.TXT, type:

```
NFT>APPEND LONDON"SMITH OPEN"::WRIT:[SMITH]FILB.TXT;3 FILA.TXT(RET)
```

The two files are now:

- The local file, FILA.TXT, which includes the contents of FILA.TXT and FILB.TXT.
- The remote file, FILB.TXT;3, which is the original copy of FILB.TXT on node LONDON.

If you are appending more than one file to the end of another file, separate the file specifications with a comma. For example:

```
NFT>APPEND LONDON"SMITH OPEN"::WRIT:[SMITH]FILB.TXT;3,FILC.TXT;2 (LF)
FILA.TXT(RET)
```

Remember to use the (LF) key to continue a command line.

3.7.2 Listing Files

For the APPEND, COPY, and TYPE commands, you can list up to ten file specifications. For example, you can append one to ten input files to a single output file. When you list files, the following rules apply:

- You cannot list more than ten file specifications.
- You cannot use wildcards (asterisk or question mark) in a list of files.
- You must separate each item in the list by a comma.
- These commands use temporary defaults when you enter a command line that contains more than one input file specification. Temporary defaults are used for:
  - Node name.
  - Device name.
  - Directory name.
  - File name and file type.

If a file specification includes a device and/or directory name, these names are applied to subsequent file specifications within the list. For example:

```
NFT>COPY BOSTON::DBA1:[ADAMS]TEST1.DAT,(LF)
TEST2.DAT,TEST3.DAT(RET)
```

This command copies three test files all from directory ADAMS on the device DBA1.

3.7.3 Copying Files

To copy files between your node and a remote node, use the COPY command.

When you copy a remote file to the local node, you must use the correct remote file specification. The following example copies a remote file named NEWS.DOC;10 from a VMS node called LONDON to the local node. The remote file is located in the directory SMITH on device WRIT. The file is given the new name FLASH.DOC when it is copied to the local node.

```
NFT>COPY LONDON"SMITH OPEN"::WRIT:[SMITH]NEWS.DOC;10 FLASH.DOC(RET)
```

You can also copy a file from a remote node to a local node without specifying a local file name. For example:

```
NFT>COPY LONDON"SMITH OPEN"::WRIT:[SMITH]NEWS.DOC;10(RET)
```

By default, the local file is named NEWS.DOC, the same as the original file.

When you copy a local file to a remote node, you must specify the local file name and the remote node name. You can also specify the remote file name in a format supported by the remote operating system. For example, to copy a DOS file called FILEA.TXT from drive B on the local node to directory SMITH on the LONDON node, type:

```
NFT>COPY B:FILEA.TXT LONDON"SMITH OPEN"::WRIT:[SMITH]FILEA.TXT;3(RET)
```

By default, the remote file is FILEA.TXT, the same as the local file.

3.7.4 Deleting Files

To delete a file or a group of files, use the DELETE command. The following example deletes all versions of a file named TEST1.TST from a remote VAX node named GENEVA.

```
NFT>DELETE GENEVA::TEST1.TST;*(RET)
```

If you only want to delete one file, enter the name and the type for the file. If there is only one version of that file on the remote node, NFT will delete it. If there are multiple versions of the same file, NFT will delete only the latest (or highest) version of the file.

To delete a file which is not the latest version of that file, you must indicate the specific version number you want to delete. For example:

```
NFT>DELETE RAMONA::FINDER.TXT;3(RET)
```

This command deletes version 3 of the file FINDER.TXT on the remote node RAMONA.

3.7.5 Displaying Directory Information

To display a list of local or remote file specifications on the screen, use the DIRECTORY command. The names are displayed in the format used by the specified node. For example, to list file specifications from a directory on the remote VAX node LONDON, type:

```
NFT>DIRECTORY/BRIEF LONDON::(RET)
```

The system responds with:

```
Directory of: LONDON"SMITH password"::SYS$SYSROOT:[SMITH]

APNDXA.DOC;3      CHAP1.DOC;2       CHAP2.DOC;13      DATA.DAT;9
NEWS.DOC;10       MEMO.TXT;1        TEST2.TST;6       TEST3.TST;2
TEXT.DOC;8
```

You can also list either a single file specification or use wildcards to specify a group of file specifications. The following example lists the group of files with a file type of TST:

```
NFT>DIRECTORY/BRIEF LONDON::*.TST(RET)

Directory of: LONDON"SMITH password"::SYS$SYSROOT:[SMITH]

TEST2.TST;6 TEST3.TST;2
```

3.7.6 Printing Files

To print a local file on a remote printer, use the /PRINT switch with the COPY or APPEND command. For example, to copy the FILEA.TXT from the local node to the remote node LONDON and then remotely print the file, type:

```
NFT>COPY/PRINT FILEA.TXT LONDON::FILEA.TXT;1(RET)
```

Be aware that you cannot print a remote file on the local printer using the /PRINT switch. If you try, the file is copied to the local node, and NFT displays the following warning message:

```
Warning: Cannot print files on local printer.
```

However, you can print a remote file on the local printer by copying it to the printer device.

You can also print a remote file on a remote printer. To do this, use the PRINT command. You must specify the name of a file that already exists on the remote node. For more information about PRINT, refer to Section 3.8.7.

3.7.7 Typing a File's Contents

To display the contents of a file on your screen, use the TYPE command. For example:

```
NFT>TYPE TEXAS::PS:<SMITH>FILE1.DOC;2(RET)
```

3.7.8 Running Command Files

A command or batch file contains a list of command strings. To execute the commands within a remote command file on the remote node, use the NFT SUBMIT command followed by the name of the remote command file. By typing commands in one batch file, you can run the file many times without retyping the commands.

You can also use the /SUBMIT switch to execute commands in a local file on a remote node. This switch is only valid with the APPEND and COPY commands. When you use /SUBMIT with either of these commands, the copied file is queued to the remote node's batch system (once the copy operation is complete).

The following example is a batch file called DUMP.CTL.13 on the TOPS–20 remote node TEXAS. The batch file is located on the device PS: in the directory JONES.

```
@PRINT PS:<SMITH>FILE1.DOC;2
@PRINT PS:<SMITH>.BAK
```

This file:

1. Prints the file called FILE1.DOC;2 located in the SMITH directory.
2. Prints all files with a file type of .BAK in the same directory.

To run this file, type:

```
NFT>SUBMIT TEXAS::PS:<JONES>DUMP.CTL.13(RET)
```

This file is placed in the batch queue on node TEXAS. You cannot run a local command file using SUBMIT.

3.7.9 Logging File Operations

When you type an NFT copy command, such as COPY or APPEND, NFT displays a notification that it has opened the specified files. In the following example, NFT logs the copy operation, and notifies you as it opens the specified files.

```
NFT>COPY INTRO.DOC LONDON"SMITH OPEN"::(RET)
```

NFT responds with:

```
Copying file:INTRO.DOC to LONDON"SMITH OPEN"::INTRO.DOC;1[10 records]
NFT>
```

The notification line is displayed in two parts:

- The first part includes all of the text up to the record count in brackets. It is displayed as soon as both the remote and local files are successfully opened.
- The second part is the record count [10 records] at the end of the line. This part is displayed when the copy operation is complete.

You can suppress the logging message for the APPEND, COPY, DELETE, PRINT, and SUBMIT commands by using the /NOLOG switch.

3.8 NFT Command Summary

NFT provides you with command verbs that allow you to manipulate files located on local and remote nodes. Table 3–7 lists each NFT command and its function. The remainder of the chapter discusses each command in alphabetical order.

Table 3–7: Valid NFT Commands

Command	Function
APPEND	Appends files from the local node to an existing file on the remote node, or copies and appends files from the remote node to an existing file on the local node.
COPY	Copies files from the local node to the remote node or from the remote node to the local node.
DELETE	Deletes a local or remote file.
DIRECTORY	Lists files located in a specified directory.
EXIT	Exits from an NFT operation and returns control to the DOS operating system. (You can also exit by typing CTRL/Z RET.)
HELP	Displays information about NFT commands and switches on the screen.
PRINT	Allows you to queue a file that exists on a remote node to be printed at that remote node.
SET	Allows you to set remote file access defaults for the current NFT session.
SHOW	Displays access control information for remote files.
SUBMIT	Submits a batch file to be run on a remote node.
TYPE	Displays the contents of a local or remote file on the screen.

3.8.1 APPEND

The APPEND command adds the contents of one or more input files to the end of an existing output file. You can append either ASCII or binary files to or from remote nodes.

Depending on its position in a command, a file specification is either input (source) or output (destination).

Format

APPEND[*/switch*] *input-file output-file*

or

APPEND[*/switch*]
File(s)? *input-file*
To? *output-file*

where

/switch is a valid APPEND switch. The effect of a switch depends upon whether you are copying a file to or from a remote node. The valid APPEND switches are:

/ASCII
/DELETE
/IMAGE
/NOLOG
/PRINT
/SUBMIT

input-file specifies one or more input files to be copied. If you specify multiple input files, you must insert a comma between the file specifications. NFT then appends the multiple files to the output file.

output-file specifies the file to which the file(s) are to be appended.

Remarks

The local file specification has one of two forms. In its longest form, it consists of the drive name, directory path, file name, and file type. For example:

```
A:USERS\SUE\SURVEY.CRD
```

In its shortest form, it consists of the file name and file type. This form assumes that the file is in the current default DOS directory. For example:

```
SURVEY.CRD
```

The remote file specification has one of two forms. In its longest form, it consists of node name and access control information followed by a file specification appropriate to the remote node. For example:

```
LONDON"SMITH OPEN"::WRIT:[SMITH]FILEA.TXT;3
```

In its shortest form, it consists of a node name followed by a file specification appropriate to the remote node, for example:

```
LONDON::FILEA.TXT;3
```

If you append more than one input file to create a single output file, the attributes of the output file are determined by the attributes of the first input file that you specify in the command. If the attributes of the input files differ, the append operation seems to succeed, but the output file may be incorrect.

Example

```
NFT>APPEND/PRINT FILE1.DAT ESTHER::FILE2.DAT;1 RET
```

This command appends the local file FILE1.DAT to the remote file FILE2.DAT;1, located at node ESTHER. The resulting output file is printed.

You can also use the /SUBMIT switch with this command. /SUBMIT causes the file you just created to be queued to the remote node's batch system, once the append operation is complete.

3.8.2 COPY

The COPY command creates a new file or a new version of a file at the destination node. You can use the COPY command to copy files from the local node to the remote node, and from the remote node to the local node.

Format

COPY[*/switch*] *input-file output-file*

or

COPY[*/switch*]
File(s)? *input-file*
To? *output-file*

where

/switch is a valid COPY switch. The effect of a switch depends on whether you are copying a file to or from a remote node.

The valid switches when you are copying a file to a remote node are:

/ALLOCATION = *nnn*
/ASCII
/BLOCK
/CC = *xxx*
/DELETE
/FIXED = *nnn*
/IMAGE
/LSA
/MACY11
/MRS = *nnn*
/NOLOG
/NOSPAN
/PRINT
/SUBMIT
/VARIABLE
/VFC = *nnn*

The valid switches when you are copying files from a remote node are:

/ASCII
/BLOCK
/IMAGE
/NOCONVERT
/NOLOG

input-file specifies the input file(s) to be copied. If you specify more than one input file, you must separate them by commas.

output-file specifies the name(s) of the output file(s).

Remarks

The local file specification has one of two forms. In its longest form, it consists of the drive name, directory path, file name, and file type. For example:

```
A:\USERS\SUE\SURVEY.CRD
```

In its shortest form, it consists of the file name and file type. This form assumes that the file is in the current default DOS directory. For example:

```
SURVEY.CRD
```

The remote file specification has one of two forms. In its longest form, it consists of node name and access control information followed by a file specification appropriate to the remote node. For example:

```
LONDON"SMITH OPEN"::WRIT:[SMITH]FILEA.TXT;3
```

In its shortest form, it consists of a node name followed by a file specification appropriate to the remote node, for example:

```
LONDON::FILEA.TXT;3
```

You can use the COPY command to:

- Copy a single file.

  This operation copies a single file from one node to another. For example:

  ```
  E>COPY SAM.TXT LONDON::SAM.TXT(RET)
  ```

- Copy a list of files.

  This operation copies more than one input file to the same number of output files.

  NOTE

  If you specify multiple names, you must separate the file names by commas. There is a limit of 10 file names for each list.

  Remember that you cannot use wildcards when you use file name lists.

  For example:

  ```
  E>COPY LONDON::SAM.TXT,SHAMI.TXT SAM.TXT,SHAMI.TXT(RET)
  ```

- Copy multiple files using wildcards.

  This operation also copies more than one input file to the same number of output files. However, using this method, you can specify more than one file without typing more than one file specification. For example, COPY *.SUM LONDON::*.SUM copies all files with a file type of SUM from the local node to node LONDON. The files are copied with the same file names and file types.

If you are copying a local file to a remote node, you can include the /PRINT switch or the /SUBMIT switch. These switches cause the file you just copied to be queued to the remote node's printer or batch system, once the copy operation is complete.

Use caution when copying files from operating systems allowing long file names, and nonalphanumeric characters. NFT truncates all file names to eight characters, and all file types to three characters. For example, NFT copies a remote file named THISISMYFILENAME.RIGHTHERE;1 as THISISMY.RIG.

Furthermore, the DOS operating system overwrites existing files of the same name. Therefore, if you copy a file called THISISMYOTHERFILENAME.RIGHTHERE;1, NFT truncates it to THISISMY.RIG, and DOS overwrites the first file with the same name. This is also true when you are copying multiple versions of the same file.

Examples

```
NFT>COPY/PRINT/DELETE PAGE1.TXT LONDON::PAGE1.TXT;1(RET)
```

This command copies the local file PAGE1.TXT to the remote node LONDON. The file is printed on the remote printer, and then deleted from the remote node.

The output file specification can be omitted. In this case, the file specification is made the same as the input file specification. For example:

```
NFT>COPY/PRINT/DELETE PAGE1.TXT LONDON::(RET)
```

3.8.3 DELETE

The DELETE command deletes one or more specified local or remote files.

Format

DELETE[*/NOLOG*] *file-spec*

or

DELETE[*/NOLOG*]
File(s)? *file-spec*

where

file-spec is any valid local or remote file specification.

Examples

```
NFT>DELETE LONDON::TAX.LST;3(RET)
```

This command deletes the file TAX.LST;3 from the remote VMS node LONDON.

```
NFT>DELETE ROME::*.DOC;*(RET)
```

This command deletes all versions of all files with a file type of DOC located on node ROME.

You can only delete multiple files using wildcards. You cannot list files to delete them.

3.8.4 DIRECTORY

The DIRECTORY command displays a list of local or remote file names, including the size (in blocks) and the time and date the file was last modified or created.

Format

DIRECTORY[*/switch*] [*file-spec*]

where

/switch	is either /BRIEF or /FULL. The default is no switch.
file-spec	is any valid local or remote file specification. If you omit the file specification, NFT assumes the local disk and the current default directory.

Examples

Example 1.

```
NFT>DIRECTORY LONDON::TAX.LST;4(RET)

Directory of: LONDON"SMITH password"::SYS$SYSROOT:[SMITH]

TAX.LST;4   20   09-JUL-85   12:30:52
```

This command displays the file name TAX.LST;4 located on node LONDON. The display includes the file's size and the time and date the file was last modified or created.

Example 2.

```
NFT>DIRECTORY/FULL LONDON::REPORT.DAT;4(RET)

Directory of: LONDON"SMITH password"::SYS$SYSROOT:[SMITH]

REPORT.DAT;4
Size:                 8/9                          Owner:   [910,20]
Created:              22-MAR-85 15:38:36
File organization:    Sequential
Record Format:        Variable length, maximum 128 bytes
Record Attributes:    Carriage return carriage control
File protection:      System: RWED, Owner:RWED, Group:RW, World:R
```

This command displays complete information for the file REPORT.DAT on the remote node LONDON. The information includes the following:

- The size of the file (in blocks)
- The owner (the specific user whose account contains this file)
- The type of file organization
- The type of record format used in the file
- The type of record attributes contained in the file
- The type of protection assigned to the file

All of this information is displayed when you use the /FULL switch for files on a remote node. If you request a DIR/FULL listing for a file on the local node, you will only see the size of the file and the time and date the file was last modified or created (see Example 1, above).

Example 3.

```
NFT>DIRECTORY/BRIEF LONDON::(RET)

Directory of: LONDON"SMITH password"::SYS$SYSROOT:[SMITH]

APNDXA.DOC;3      CHAP1.DOC;2      CHAP2.DOC;13     DATA.DAT;9
NEWS.DOC;10       MEMO.TXT;1       TEST2.TST;6      TEST3.TST;2
TEXT.DOC;8
```

This command lists all file names on the NODE LONDON. The list does not include sizes, times, or dates. If a file name consists of more that 19 characters, NFT truncates the name to 8 characters in the directory listing.

3.8.5 EXIT

The EXIT command exits from an NFT operation and returns control to the DOS operating system.

Format

EXIT

Example

NFT>EXIT(RET)

This command exits you from the NFT utility. You can also enter (CTRL/Z) and then (RET) to exit from NFT.

3.8.6 HELP

The HELP command displays information about NFT commands and switches on your screen.

Format

HELP $\begin{bmatrix} \textit{command-verb} \\ \textit{/switch} \end{bmatrix}$

where

command-verb is any valid NFT command verb.

switch is any switch acceptable to NFT command verbs.

Examples

```
NFT>HELP(RET)
```

This command displays a summary of all NFT commands.

```
NFT>HELP SHOW(RET)
```

This command displays a summary of the SHOW command.

```
NFT>HELP /ASCII(RET)
```

This command displays a summary of the effects of the /ASCII switch.

3.8.7 PRINT

The PRINT command queues a remote file to be printed on a remote printer. You must indicate the file to be printed. The file will be printed on the system default printer.

Format

PRINT[*/switch*] *file-spec*

or

PRINT[*/switch*]
File(s)? *file-spec*

where

/switch is a valid PRINT switch. The valid PRINT switches for printing a file on a remote node are:

/DELETE
/NOLOG

file-spec is any valid remote file specification.

Examples

```
NFT>PRINT LONDON::TAX.LST;3(RET)
```

This command queues the file TAX.LST;3 (located on the remote VMS node LONDON) to be printed on the default system printer (which is also located on node LONDON).

```
NFT>PRINT ROME::CHAPTER1.MEM(RET)
```

This command queues the file CHAPTER1.MEM on node ROME to be printed on ROME's default system printer. Note that you cannot use wildcards or lists with the PRINT command.

3.8.8 SET

The SET command allows you to set default access control information as well as disk and directory information for up to ten nodes. After you set this information, you can type the node name only, followed by two colons. NFT checks its default access control information table for a record matching the specified node name. If it finds a match, NFT uses this information for network access.

There are no valid switches for the SET command.

Format

SET *node-spec*::[*file-spec*]

or

SET
Node? *node-spec*::[*file-spec*]

where

node-spec is a remote node name and access control information, followed by two colons.

file-spec is the file specification string. It includes the disk and/or directory information for the remote node. It does not include the file name and version number.

Examples

```
NFT>SET LONDON"SMITH OPEN"::SYS$USER_DISK:[USER.FILES]RET
```

This command sets default access control information for node LONDON. The information includes the disk that the file is on, SYS$USER_DISK, and the name of the subdirectory, USER.FILES.

```
NFT>SET PARIS"JONES NEW"::RET
```

This command sets default access control information for node PARIS. This information includes only the username JONES and the password NEW.

3.8.9 SHOW

The SHOW command displays the temporary default table of remote access information for the specified node. The password is not displayed on the screen. Instead, the character string *password* is displayed in its place.

There are no valid switches for the SHOW command.

Format

SHOW [*node-name*::]

where

node-name:: is a valid remote node name. If no node name is specified, NFT displays the contents of the access control information table for all nodes known to the NFT temporary default table.

Remarks

If NFT does not find the specified node name in the table, it checks to see if you typed default remote access control information using NCP. If you did, NFT displays the access control information. If NFT does not find the node name in its own table or with NCP, NFT displays:

```
No defaults match: node-name::
```

Examples

```
NFT>SHOW(RET)
```

This command displays the contents of the access control information table for all nodes known to the NFT temporary default table. This command does not display defaults you set using NCP.

```
NFT>SHOW LONDON::(RET)
```

This command displays the access control information for node LONDON.

3.8.10 SUBMIT

The SUBMIT command requests that the specified command file be run on the remote node. The specified remote node must support command file submission and execution.

Format

SUBMIT[*/NOLOG*] *remote-filespec*

or

SUBMIT[*/NOLOG*]
File(s)? *remote-filespec*

where

remote-filespec has two formats. In its longest form, it consists of node and access control information followed by a file specification required by the remote node. (See Section 3.2 for details on access control information.) In its shortest form, it consists of a node name followed by a file specification appropriate to the remote node, for example, NODE::FILE.TYP;3.

Remarks

Command files contain one or more commands that are recognized and run by the remote node's operating system. This means that the format of the commands within the file must conform to the standards of the remote system.

Example

```
NFT>SUBMIT PARIS::BACKUP.COM(RET)
```

This command requests that the file BACKUP.COM be run on the remote node PARIS. Note that you cannot use wildcards or lists with the SUBMIT command.

3.8.11 TYPE

The TYPE command displays the contents of a local or remote file on your screen. You should use this command for ASCII files only.

Format

TYPE *file-spec*

or

TYPE
File(s)? *file-space*

where

file-spec is any valid local or remote file specification.

Example

```
NFT>TYPE  LONDON: :PREFACE.DOC(RET)
```

This command displays the contents of the file PREFACE.DOC located on the remote node LONDON.

3.9 Providing Remote File Access to Your Node Using FAL

The File Access Listener (FAL) provides a way for you to let other nodes access files on your node. The purpose of the FAL utility is to listen for and receive remote access requests from the network. These requests are the results of file access routines from other nodes in the network (such as NFT on another pc node or a COPY command from a DECnet–VAX system). When FAL detects a request to copy a file to or from your node, it first checks the type of access privilege contained in the request. If it has the proper privilege for access to your node, FAL sends or receives the file data to or from the requesting node. (Refer to Chapter 2 for details about specifying and displaying access information, using the NCP commands SET ACCESS and SHOW ACCESS.)

FAL must be running on your system or must be run by the job spawner (see Section 3.10) before any exchanges of file data can take place. To run FAL, type the command at the system prompt. For example:

```
C>FAL(RET)
```

The utility will run until you cancel it with (CTRL/C) or any other character. If you use (CTRL/C), FAL will not finish the current task. Instead, it will exit immediately and close down all links with the remote node. If you use any other character, FAL will first finish its current task and then exit.

While FAL is running, file data is passed back and forth between your node and the requesting node. You can control the way FAL operates by using switches. The valid switches for FAL include:

/ASCII (/A)
/BINARY (/B)
/ERROR (/E)
/LOG (/L)

To use these switches, type them after the FAL command on the same line. There should be a space between the command and the switch. If you use more than one switch, be sure to separate them with a space.

The following sections describe the FAL switches in more detail.

3.9.1 /ASCII

MS–DOS systems do not retain file attributes with files. As a result, when FAL is asked by a remote node for the attributes of a file on the pc, it cannot determine what the attributes are.

When you use the /ASCII switch, all files that are copied from the pc are transferred as ASCII files. An ASCII file is a file whose records end with end-of-text or line-terminating characters, such as a carriage return/line feed CR/LF pair.

If you use the /ASCII switch, FAL will copy all files in ASCII format, including image files. Note that image files can lose data if they are copied in the ASCII format.

Format

/ASCII

Example

```
C>FAL /ASCII RET
```

NOTE

If you specify neither the /ASCII nor the /BINARY switch, the default is CHECK. When FAL is started without either switch, it automatically checks each file it is requested to send. FAL looks for any CR/LF characters in the first 512 bytes of the file. If there are any, it sends the file as ASCII. If the requested file is a binary file, FAL sends the file as binary.

3.9.2 /BINARY

MS–DOS systems do not retain file attributes with files. As a result, when FAL is asked by a remote node for the attributes of a file on the pc, it cannot determine what the attributes are.

When you use the /BINARY switch, all files that are copied from the pc are transferred as binary files. A BINARY (or image) file is a file whose data is copied exactly as it appears, with no interpretation of CR/LF characters.

Using this switch ensures that any file copied to a remote node and then back to your pc will not lose any data. However, ASCII files copied from the pc when the /BINARY switch is used may not be in the proper format for use on the remote node.

Format

/BINARY

Example

```
E>FAL /BINARY(RET)
```

NOTE

If you specify neither the /ASCII nor the /BINARY switch, the default is CHECK. When FAL is started without either switch, it automatically checks each file it is requested to send. FAL looks for any CR/LF characters in the first 512 bytes of the file. If there are, it sends the file as ASCII. If the requested file is a binary file, FAL sends the file as binary.

3.9.3 /ERROR

When you use the /ERROR switch, FAL will report an error to the remote node if the requesting node is attempting to overwrite an existing file. If you do not set this switch, an existing file will be deleted and a new one will be created with the same name. The default is NO ERROR.

Format

/ERROR

Example

```
E>FAL /ERROR(RET)
```

3.9.4 /LOG

When you use the /LOG switch, FAL logs the type of access for every request it receives. The type of access includes the following information:

- the command or request (such as an NFT DIRECTORY or DELETE command)
- the requested directory name and file name
- the node name
- the access control information for that node (username and account)

This information is displayed on the screen in front of you. If you do not want to see the information on the screen, you can redirect it to an output file when you first issue the FAL command.

Format

/LOG:*filename*

where

filename is the name of a file to use for storing the access information.

Example

```
C>FAL /LOG:ACCESSINFO.TXT
```

This command causes the logging information to be appended to the file ACCESSINFO.TXT.

3.10 Using Your Computer as a Server

The Job Spawner is a DECnet–DOS utility that allows your computer to act as a server for performing multiple service functions. When you turn on the job spawner, it listens for any connect requests from other nodes in the network. When a connection arrives, the spawner starts the program which will service that request. When the service is complete, the spawner again waits for connections. In this way, your pc can process different requests at different times without any need for user intervention.

In DECnet–DOS V1.1, the spawner supports both FAL and DTR.

As an example, if the job spawner is running and it detects a request for file transfer, the spawner initiates the FAL utility and causes FAL to run. FAL will continue to run until the requested activity is complete. The spawner then continues to listen for other requests. (Two service programs cannot be run simultaneously.)

To tell the spawner what program to run for each kind of connection that arrives, you must create the file DECSPAWN.DAT. You can use any editor that is installed on your system to create this file. It is an ASCII text file used as a database by the spawner.

The format of DECSPAWN.DAT can be any number of the following lines:

<object name or number> <program to run>

A sample DECSPAWN.DAT which supports FAL and DTR would be:

```
17      FAL
63      DTR
```

If you wish to write your own server programs, note the following:

1. The server program is run by the spawner as if this command were typed:

   program –use *n*

   where *n* is the number of a socket accepted in deferred mode. The program must complete the connection.

2. If a server program exits with a non–zero status, the spawner will detach the socket. If the program exits with a zero status, the socket should be detached by the server before exit.

To start the job spawner, simply type SPAWNER at the system prompt and press RET. To exit from the spawner, type an exclamation point (!).

Format

SPAWNER

Remarks

While you are running the job spawner utility, no other utilities should be running.

Examples

The following example illustrates the spawner running and detecting a request for DTR:

```
press "!" to abort
SPAWNER (V1.1.0) listening - Fri Feb 07 08:34:53 1986
Fri Feb 07 08:35:11 1986 - Connect request from node 55.49 for object
  63

Executing: dtr -use 2. . .

[Appending messages to log file: E:dECNETdTR.LOG]

DTR --I-- V1.1 (21-Jan-86) started on Fri Feb 07 1986 at 08:35:12
DTR --I-- Connect reject test initiated on Fri Feb 07 1986 at 08:35:12

DTR --I-- Terminated on Fri Feb 07 1986 at 08:35:13
SPAWNER (V1.1.0) listening - Fri Feb 07 08:35:13 1986
Fri Feb 07 08:35:23 1986 - Connect request from node 55.49 for object
  63

Executing: dtr -use 2. . .

[Appending messages to log file: E:dECNETdTR.LOG]

DTR --I-- V1.1 (21-Jan-86) started on Fri Feb 07 1986 at 08:35:23
DTR --I-- Data echo test initiated on Fri Feb 07 1986 at 08:35:24
DTR --I-- Data test complete on Fri Feb 07 1986 at 08:35:32

DTR --I-- Terminated on Fri Feb 07 1986 at 08:35:32
SPAWNER (V1.1.0) listening - Fri Feb 07 08:35:33 1986
Fri Feb 07 08:35:44 1986 - Connect request from node 4.30 for object
  19

Could not map object to image - rejecting connect
SPAWNER (V1.1.0) listening - Fri Feb 07 08:35:44 1986

SPAWNER exiting - Fri Feb 07 08:36:07 1986
```

If the spawner cannot locate the necessary database to use for answering a request, the following information is displayed:

```
Could not open file E:\DECNET\decspawn.dat

. Abort job spawner by pressing ''!''
. Job spawner does not support multiple links.
. Database file for spawner in DECnet database path.
  . Name of database file is DECSPAWN.DAT.

    Example contents of a DECSPAWN.DAT:
          TELL          tell
          TIMESRV       timesrv
          SERVER        server
           63           dtr
           27           receiver
```

4
Logging On to a Remote Host

This chapter describes the SETHOST utility with network virtual terminal services. This utility connects your system to a **host** node and allows your system to act like (emulate) a terminal connected to the host node.

When your personal computer emulates a host terminal, you can perform many standard terminal functions of that host and gain access to the host's resources as if your personal computer was a terminal directly connected to the host node.

Each time you log onto a host, the connection you make is called a session. Using SETHOST, you can create and maintain multiple sessions. The maximum number of sessions depends on the maximum number of network links you have established with the DEFINE EXECUTOR MAXIMUM LINKS command (see Chapter 2). The number of sessions also depends on the number of links that are currently active for the virtual disk or the virtual printer.

There are two techniques for gaining access to a remote host system. Each technique has its own characteristics. The techniques are:

1. LAT

   This technique provides access to LAT services. LAT is a local area transport protocol that communicates over Ethernet. It is available for Ethernet configurations only. LAT supports TDMS applications, and can only be used for single sessions.

2. CTERM

   CTERM is a wide area virtual terminal protocol which is layered on top of DECnet. It provides access to the entire DECnet network. It is available for both Ethernet and asynchronous DDCMP configurations. CTERM does not provide support for TDMS applications, but it does allow you to use multiple sessions.

When SETHOST connects your node to another node, it checks to see if you have a LAT driver installed. If so, SETHOST first attempts to connect your node to a LAT line. This is the default. If LAT is not available, then SETHOST attempts to connect you using CTERM. You can select CTERM over LAT as the default by using the /CTERM switch.

To use SETHOST, the host node must support terminal emulation from remote nodes and run Phase IV of the DECnet software on one of the following operating systems:

- VAX/VMS, Version 4.0 or later
- RSX–11M–PLUS, Version 3.0 or later
- RSX–11M, Version 4.2 or later
- Micro–RSX, Version 3.0 or later
- TOPS–10, Version 7.03 or later
- TOPS–20, Version 6.1 or later
- ULTRIX–32, Version 1.1 or later

4.1 Using the SETHOST Command

The SETHOST command, its parameters, and its various switches allow you to create, record, and modify your remote sessions. You can use the parameters and the switches in the command line, or you can use them at the SETHOST prompt.

Format

SETHOST [*nodespec* or *latservicename*] [*/switch*]

where

nodespec specifies the DECnet node name or DECnet node number to use for establishing a remote CTERM session.

latservicename specifies the LAT service name to use for establishing a remote LAT session.

/switch is a valid SETHOST switch. The valid SETHOST switches are:

/ABORT = *n*	/LOG = *filespec*
/CTERM	/SAVEDEFAULT
/EDIT = *edit-mode*	/SESSION = *session__number*
/EXIT	/SHOW
/HELP = *topic*	/TAKE = *filespec*

The /EDIT, /SAVEDEFAULT, /SESSION and /SHOW switches are used with CTERM only.

The following sections describe the SETHOST switches and parameters in more detail.

4.1.1 /ABORT

The /ABORT switch lets you specify which session you want to abort. You can abort a single session or you can abort all the sessions. To view the current sessions, use the /SHOW switch.

Format

/ABORT = *n*
ALL

where

n	is the specific session number you want to abort.
ALL	indicates that you want to abort all of the current sessions.

Remarks

You can abort the current session, or you can abort a session that was saved previously. Note that if you abort the current session it will be deleted. Your link will be disconnected and SETHOST will exit.

Example

The following example aborts session number 3, leaving all other sessions intact.

```
SETHOST /ABORT=3(RET)
```

4.1.2 /CTERM

The /CTERM switch lets you specify that CTERM protocol be used for the current connection. The default is to attempt to use LAT protocol first. If LAT is unavailable or if the LAT attempt fails, then CTERM protocol is used to complete the connection. Only CTERM provides support for multiple sessions.

Format

/CTERM

Example

The following example forces the connection to use CTERM protocol rather than LAT for establishing a session with the remote node THRUSH.

```
SETHOST THRUSH /CTERM(RET)
```

4.1.3 /EDIT

The /EDIT switch controls the type of command line editing you can perform during your remote session. This switch only applies to CTERM hosts or sessions that support command line editing.

Format

/EDIT = *edit-mode*

where

edit-mode is one of the three modes you can use for editing a command line. The three editing modes are:

- OVERSTRIKE – causes the new character to replace a character at the current cursor position.
- INSERT – causes the new character to be inserted to the left of the character at the current cursor position.
- NONE – turns off command line editing.

Remarks

You can toggle between INSERT and OVERSTRIKE (within a single command line) by using CTRL/A. The default editing mode is reset at the beginning of each line. The default mode is OVERSTRIKE, unless you have changed it with a /EDIT switch (or if it is a saved default – refer to /SAVEDEFAULT for more information).

Example

The following example changes the editing mode to INSERT.

```
SETHOST /EDIT=INSERT RET
```

4.1.4 /EXIT

The /EXIT switch lets you exit from SETHOST. You can enter the switch after you have typed the SETHOST command.

Format

/EXIT

Example

The following example illustrates first starting the utility, then requesting help for NODESPEC, and finally exiting from SETHOST.

```
SETHOST RET

SETHOST /HELP=NODESPEC RET

.
.
.

SETHOST /EXIT RET
```

4.1.5 /HELP

The /HELP switch displays the main help text screen for SETHOST. You can also get help on specific topics.

Format

/HELP = *topic*

where

topic is one of the specific SETHOST options for which you can view help text. The SETHOST help topics are:

nodespec	@	*	?	
/SESSION	/TAKE	/LOG	/HELP	/SAVEDEFAULT
/SHOW	/EDIT	/ABORT	/EXIT	/CTERM
keyboard-layouts		context-switching		
syntax		changing-default-line-editing-mode		

Example

The following example displays help text for the syntax of SETHOST switches.

```
SETHOST /HELP=syntax
```

4.1.6 *latservicename* or *nodespec*

The *latservicename* parameter lets you specify the LAT service name to use for establishing a remote session. The *nodespec* parameter lets you specify the DECnet node name or the DECnet node address to use for establishing a remote session.

Format

SETHOST *nodespec* or *latservicename*

where

nodespec specifies the node name or node address to use for the remote session. The nodespec can also include access information, such as the user's password or account information. The format for the access information is [/USER/PASSWORD/ACCOUNT]. Note that this access control information is only used by ULTRIX hosts.

latservicename specifies the LAT service name to use for the remote session.

Remarks

LAT service names are defined in two ways:

1. They are accumulated over time by the LAT driver.
2. They can be set using the following NCP command:

   NCP SET NODE *address* NAME *name* LAT

If the LAT service name you specify with SETHOST is not yet known by the driver, or if you have not set it with the NCP SET command, the LAT connection can fail.

Example

The following example establishes a session on the node ROMAN, for the user WESTON and password KERNEL.

```
SETHOST ROMAN/WESTON/KERNEL(RET)
```

4.1.7 /LOG

The /LOG switch lets you record the terminal session in a log file.

Format

/LOG = *filespec*

where

filespec is the name of the file you want to use for recording your session. You can specify any file name with maximum of 8 characters for the name and 3 characters for the file extension. If you do not specify a name, the default file name is SESSION.LOG.

Example

The following example specifies SESSION4.LOG as the file to use for recording the session.

```
SETHOST /LOG=SESSION4.LOG[RET]
```

4.1.8 /SAVEDEFAULT

The /SAVEDEFAULT switch lets you save the switch settings to be used as the default for all future sessions. The /SAVEDEFAULT switch only works with the /EDIT and the /CTERM switches. If you use /SAVEDEFAULT and do not specify any switches to be saved, it has no effect.

When you use this switch, SETHOST places your specified switch settings in a file called SETHOST.DEF in the DECnet directory. SETHOST reads this file whenever you start SETHOST.

Format

/SAVEDEFAULT /SWITCH [/EDIT] [/CTERM]

Example

The following example saves the default edit mode of INSERT in the SETHOST.DEF file.

```
SETHOST /EDIT=INSERT/SAVEDEFAULT(RET)
```

4.1.9 /SESSION

The /SESSION switch specifies the session number to use for reconnection to a host. It only works if you are connected to the host using CTERM.

You can use the /SHOW switch to determine which multiple sessions are available. You can then use /SESSION to reestablish a connection to a particular session.

Format

/SESSION = *n*

where

n is the number of the session to use for reconnecting to a host.

Example

The following example reconnects you to session number 2.

```
SETHOST /SESSION=2RET
```

4.1.10 /SHOW

The /SHOW switch displays a list of the current sessions and the hosts that connect them to your node. The /SHOW switch only works if you are connected to the host using CTERM.

Format

/SHOW

Example

The following example displays 2 current sessions on hosts KL1026 and BOULET.

```
SETHOST /SHOW(RET)

Session      Host

    1        KL1026
    2        BOULET
```

4.1.11 /TAKE

The /TAKE switch sends text from a local file to the remote node as if it were typed at the remote node's keyboard.

Format

/TAKE = *filespec*

or

@filespec

where

filespec is the name of the file containing the text you want to send to the remote node. If you use the /TAKE switch but you do not supply a file name, the default file name is SESSION.BAT.

Example

The following example requests that text be taken from the file SESSCOM.BAT and sent to node WINTER. You must include a space after the name of the remote node.

```
SETHOST WINTER /TAKE=SESSCOM.BAT(RET)
```

NOTE

Some remote systems or services may not be able to process more than one line of text correctly.

4.2 Interrupting a Session

Whenever you interrupt a session by using CTRL-\ RET, the SETHOST menu is displayed. The SETHOST menu lists various options for recording or changing your remote session. The following example illustrates a sample SETHOST menu.

```
$ CTRL-\ RET
<Return> to exit SETHOST
<H>elp
<L>og session off
<R>econnect to session 1
<E>diting mode is overstrike
<S>ession
<A>bort session
<N>ew session

Command:
```

The angle brackets indicate the key or letter you need to press to request one of the menu functions. Type only the first letter of the command you want to use. You do not need to press the RET key after you enter the letter.

Note that the "Editing mode" option only applies to CTERM hosts or sessions that support command line editing.

The following sections describe each of the menu options in more detail.

4.2.1 <RETURN>

The <RETURN> key lets you save the current session and exit from SETHOST. To return to the current session or reestablish a previous session, type SETHOST and the appropriate session number. For example:

```
E:>SETHOST /session=2
```

This happens only if you are using CTERM. If you are using LAT, the <RETURN> key disconnects you from the host and does not save the session.

The previous example reconnects you to Session 2. (Note that logging off the remote host does **not** save the current session. You must use <RETURN> if you want to come back to the session at a later time.) If you want to review the established sessions, type SETHOST and press RET. The system displays a listing of available session numbers, as well as the host system to which they are connected.

4.2.2 <H>ELP

HELP displays the help text for the SETHOST menu.

4.2.3 <L>OG

LOG lets you record your entire session as a file. When you type ◻, it acts as a toggle between ON and OFF. If the menu reads "Log session off" and you type ◻, SETHOST changes logging to ON and prompts you for the filename to use for recording the session. When you redisplay the SETHOST menu, it should now indicate "Log session on" with the name of the file you indicated.

If the menu reads "Log session on *filename*" and you type ◻, SETHOST changes logging to OFF and closes the file it was using to record the session. When you redisplay the SETHOST menu, it should now indicate "Log session off".

4.2.4 <R>ECONNECT

RECONNECT lets you leave the SETHOST menu and return to the current session. The current session number always displays on the menu. In the preceding menu example, the session number is 1.

4.2.5 <E>DIT

EDIT lets you change the default mode for command line editing. (This option only applies to CTERM hosts or sessions that support command line editing.)

When you select this option, SETHOST displays a submenu with the following choices:

- <O>verstrike
- <I>nsert
- <N>one

"Overstrike" mode means any letters you type replace the letters currently on the screen. "Insert" mode means that letters are not replaced. Instead, they are inserted into the command line at the left of the cursor. "None" means that no command line editing is available.

4.2.6 <S>ESSION

SESSION lets you connect to another session that you have previously saved. (This option only applies to CTERM hosts or systems that support multiple sessions.)

When you select this option, SETHOST displays a list of the available sessions. You will be prompted for the session number you want to use.

4.2.7 <A>BORT SESSION

ABORT SESSION lets you abort a single session. You can abort the current session, or you can abort a session that you saved previously. When you select this option, SETHOST prompts you for the session number to be aborted. If you abort the current session (indicated by "<R>econnect to session *n*"), it will be deleted. Your link will be disconnected and SETHOST will exit.

4.2.8 <N>EW SESSION

NEW SESSION begins a new session. When you select this option, you are prompted for a node name or a LAT service name to use for starting the next session.

4.3 Your Rainbow 100 and IBM PC/XT or PC AT System as a VT102 Terminal

The Rainbow 100 computer systems support the keyboard functions performed by the VT102 class terminals. You should refer to the Rainbow documentation that describes VT102 terminal emulation.

The IBM PC/XT and Personal Computer AT also provide VT102 class terminal support but only during SETHOST sessions. To enter data on your IBM system as a VT102 class terminal, different keys or combinations of keys may be required to simulate the VT102 class terminal key functions. Refer to documentation specific to your program or application for information on VT102 class terminal emulation.

When **receiving data**, the IBM PC/XT and PC AT personal computers and the VT102 class terminals accept the same series of escape sequences. Refer to documentation on VT102 class terminals for more information.

Figure 4–1 shows the layout of the Rainbow 100 system keyboard. The special keys discussed in this chapter are highlighted on the illustration.

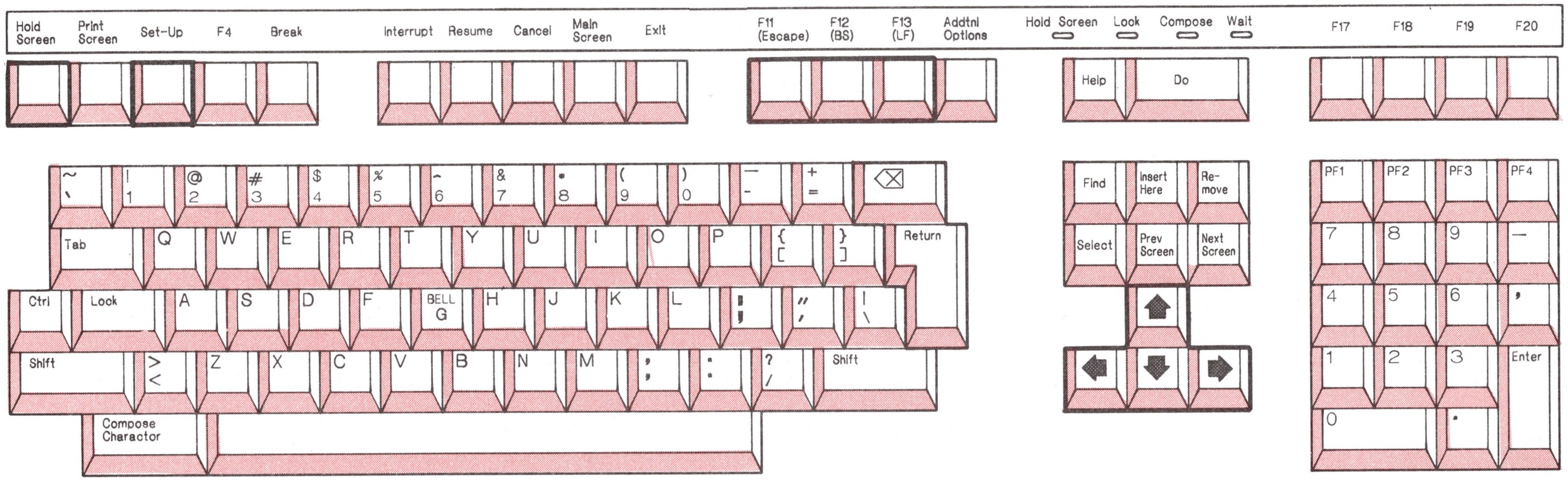


Figure 4–1: Rainbow 100 System Keyboard

Figure 4–2 shows the layout of the IBM PC/XT system keyboard. The special keys discussed in this chapter are highlighted on the illustration.

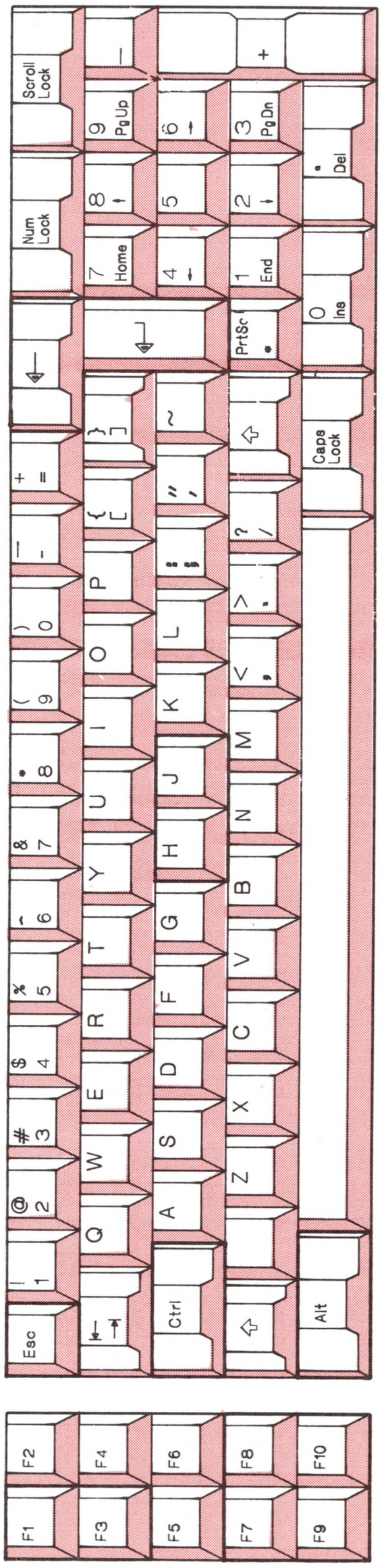


Figure 4–2: IBM PC/XT System Keyboard

Figure 4–3 shows the layout of the IBM PC AT system keyboard. The special keys discussed in this chapter are highlighted on the illustration.

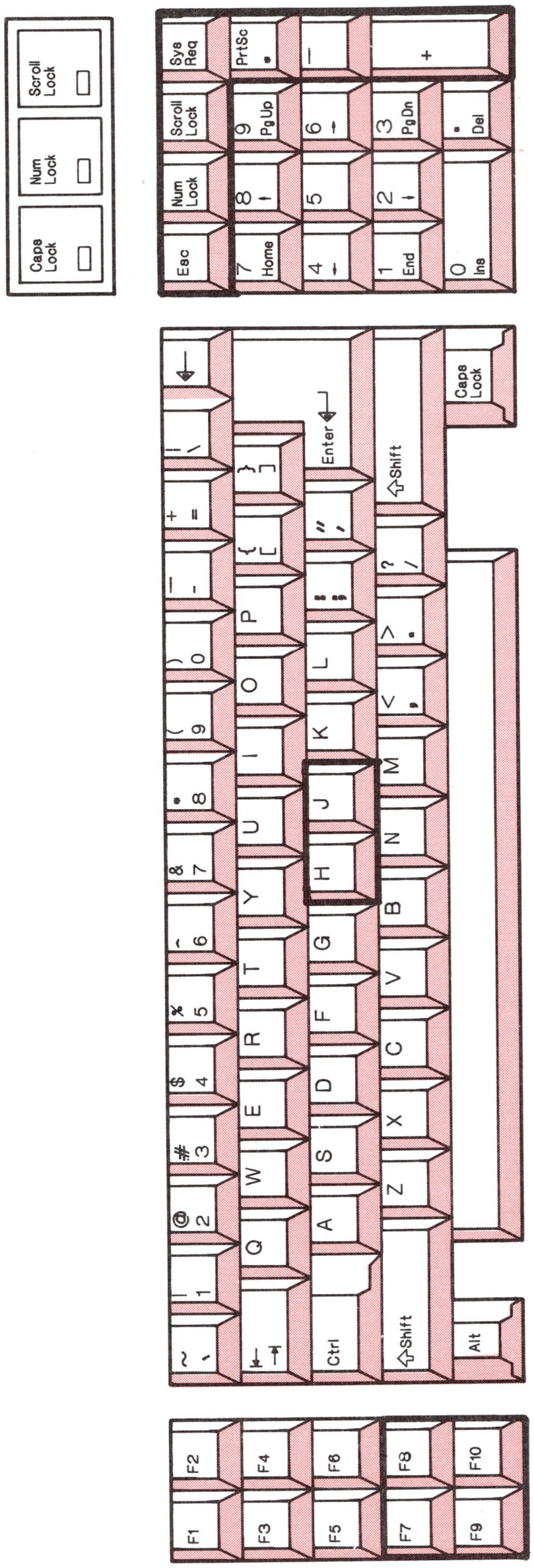


Figure 4–3: IBM PC AT System Keyboard

4.4 Using Your IBM System Keyboard during Terminal Emulation

This section describes the IBM keys that can be used with SETHOST. For easy reference, the functions are divided into four groups:

- Letters, numbers and punctuation marks (main keyboard)
- Function keys
- Arrow keys
- Numeric keypad

Certain functions are implemented on the IBM system by holding down one of the following special keys: the shift key (/\), the CTRL key, or the ALT key while you depress another key. In text, this action is indicated by the notation SHFT/*x*, CTRL/*x* and ALT/*x* where *x* represents the additional key.

Some of the commands you issue with the special keys can also be implemented using either the Rainbow keys or certain function keys. Table 4–1 lists the commands, Rainbow keys, and IBM function keys that are available when you emulate a VT102-class terminal on a VMS system.

Table 4–1: VMS Commands and Keys

Command	Rainbow Key	IBM PC/XT Key	IBM PC/AT Key	Function
CTRL/A				Switches between overstrike mode and insert mode. The default mode (overstrike) is reset at the beginning of each line.
CTRL/B	↑	F7	F7	Recalls previously entered commands.
CTRL/D	←	F9	F9	Moves the cursor one character to the left.
CTRL/E				Moves the cursor to the end of the line.
CTRL/F	→	F10	F10	Moves the cursor one character to the right.
CTRL/H	F12			Moves the cursor to the beginning of the line.
CTRL/J	F13			Deletes the word to the left of the cursor.
CTRL/U				Deletes characters from the beginning of the line to the cursor.
CTRL/W				Deletes the word to the left of the cursor.
	↓	F8	F8	Allows you to re-examine a command you previously recalled using CTRL/B, Up-arrow, or F7.

Table 4–2 lists the commands, the Rainbow keys, and the IBM function keys that you can use when emulating a non-VMS system.

Table 4–2: Commands and Keys for Systems Other than VMS

Command	Rainbow Key	IBM PC/XT Key	IBM PC/AT Key	Function
CTRL/A				Switches between overstrike mode and insert mode. The default mode is reset at the beginning of each line.
CTRL/U				Deletes characters from the beginning of the line to the cursor.
CTRL/W				Deletes the word to the left of the cursor.
↓	F8	F8		None. Down-arrow escape sequence will be sent to the host, with no effect.
←	F9	F9		Moves the cursor one character to the left.
→	F10	F10		Moves the cursor one character to the right.
↑	F7	F7		None. Up-arrow escape sequence will be sent to the host, with no effect.

4.4.1 Letters, Numbers and Punctuation Marks

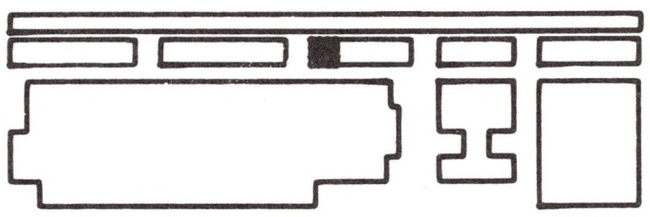
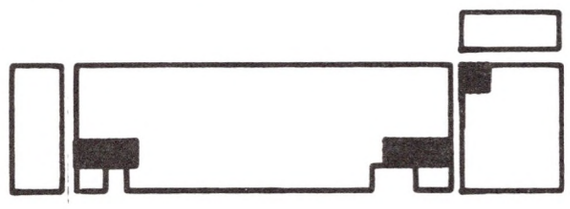
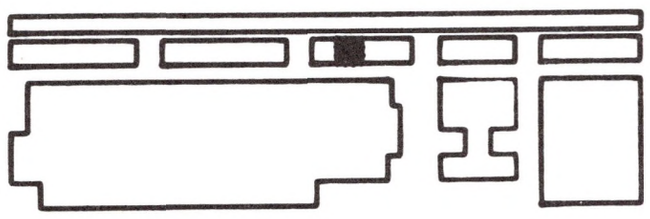
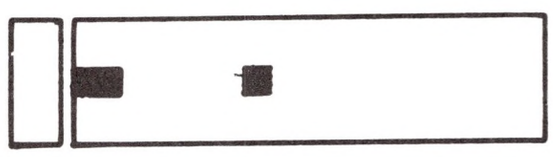
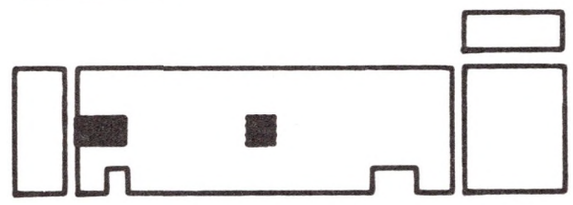
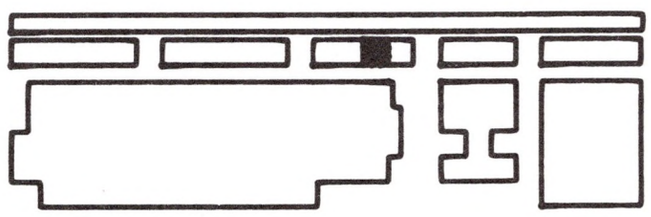
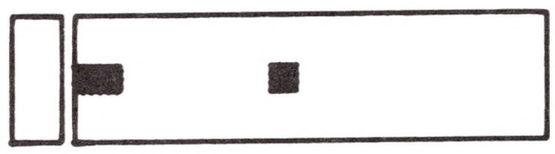
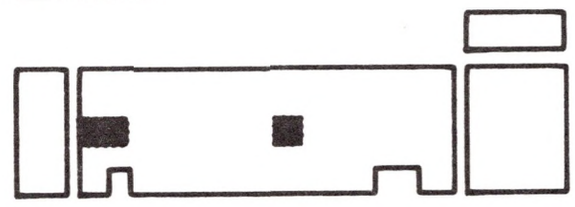
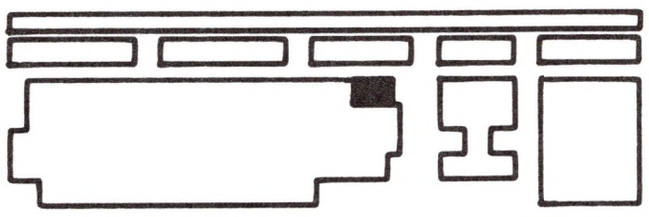
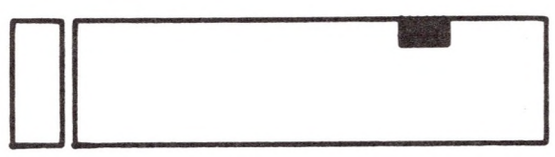
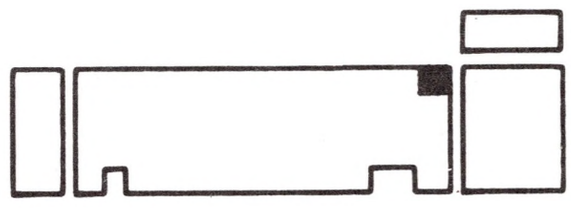
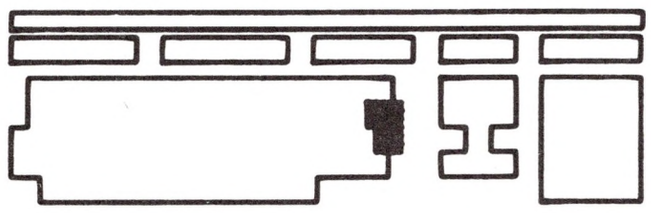
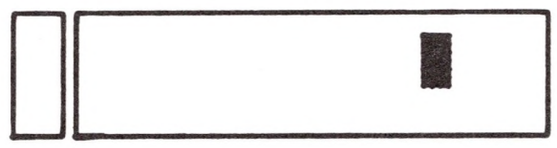
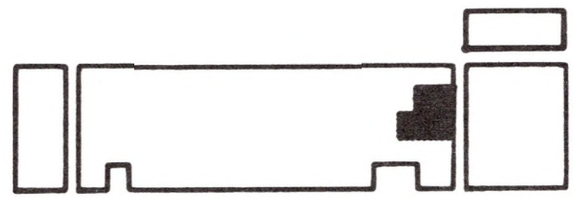
Letters, numbers and punctuation marks on the IBM system are equivalent to the same symbols on the VT102 class terminals. However, some keys are in different positions on the keyboard. They function in the same way.

4.4.2 Function Keys

Certain key combinations are required for the IBM system to simulate VT102 class terminal functions. They are:

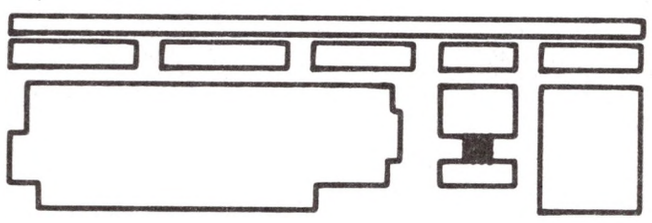
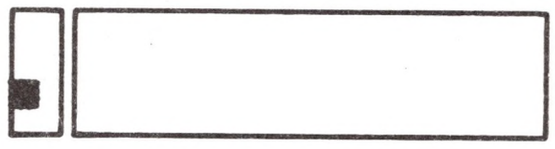
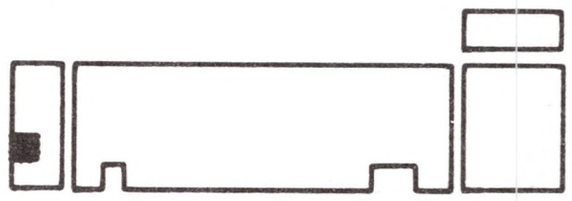
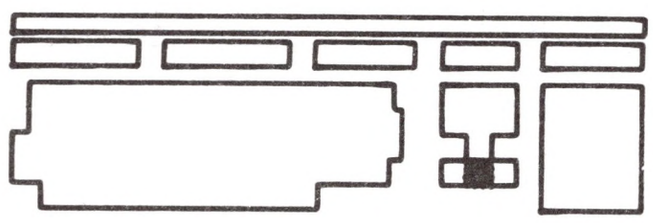
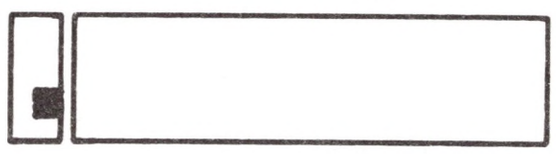
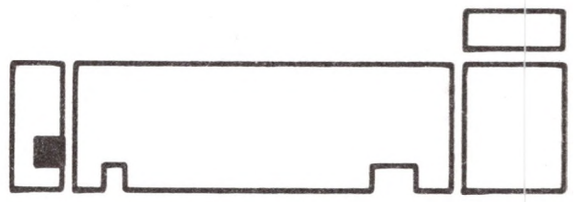
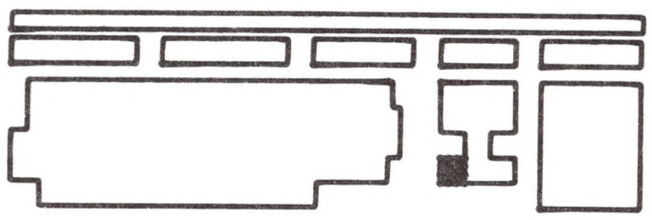
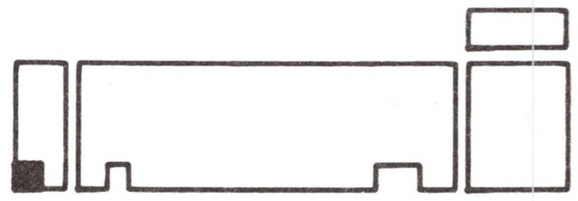
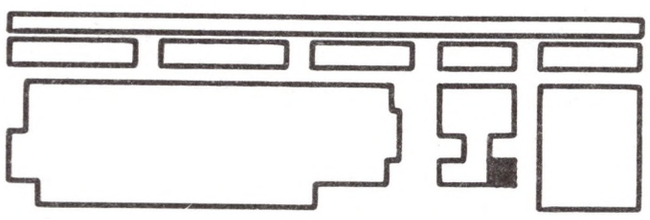
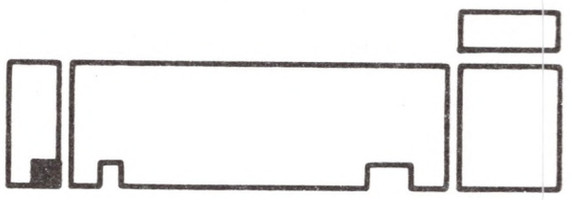
- The ALT/F1 key combination on the IBM system simulates the HOLD SCREEN key on the Rainbow.
- The ALT/F3 key combination on the IBM system simulates the SET-Up key on the Rainbow. This particular key combination enables you to enter and exit from the SET-Up display on your IBM system.

In addition, there are other sets of keys that function in the manner that you would expect when communicating with a host system. The specific keys, and how the IBM keys map to the Rainbow 100 keys appear in the following list. An illustration showing the location of the specific key(s) on each system appears immediately under the key pairs.

Rainbow 100 Key	IBM PC/XT Key	IBM PC AT Key
ESC	ESC	SHIFT ESC
		
BS	CTRL/H	CTRL/H
		
LF	CTRL/J	CTRL/J
		
X	←	←
		
Return	+	+
		

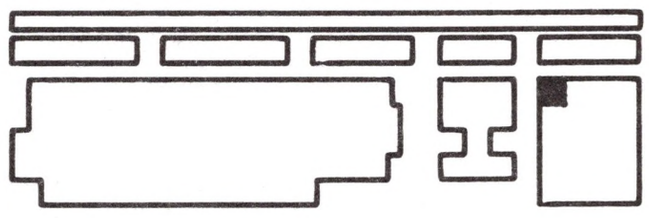
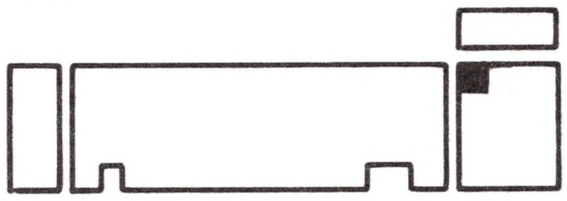
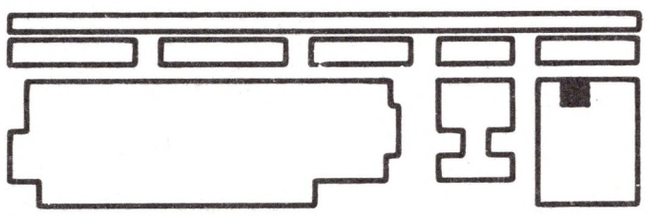
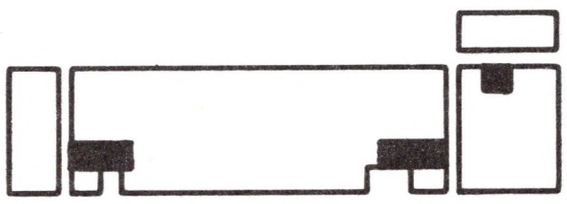
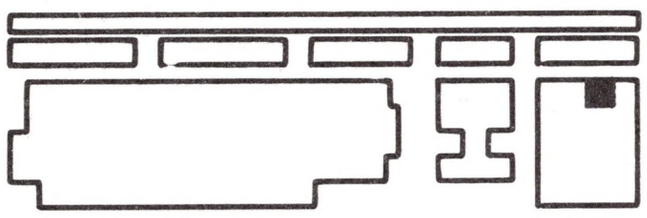
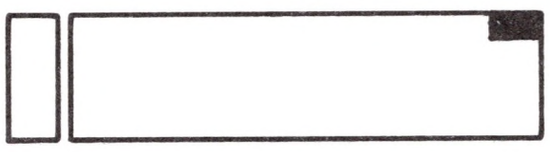
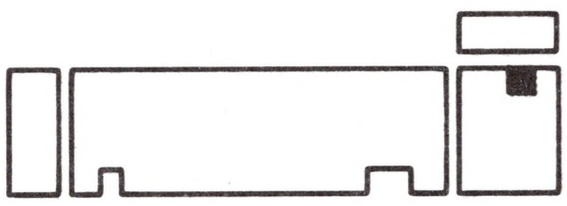
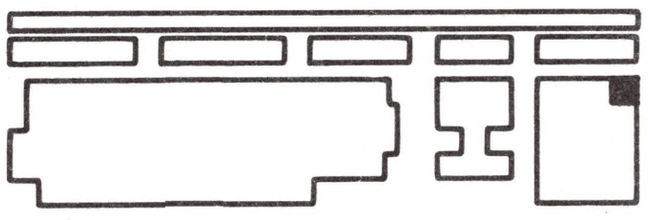
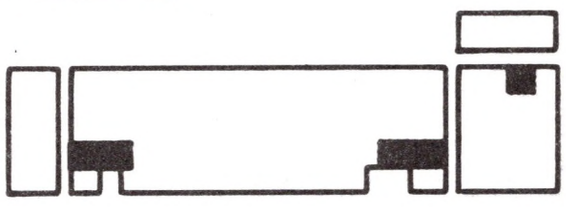
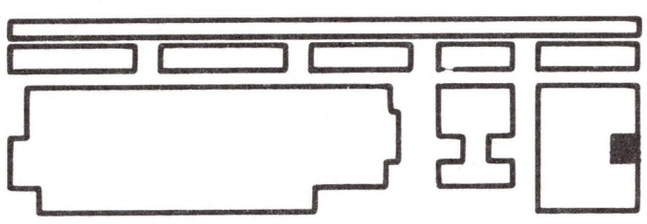
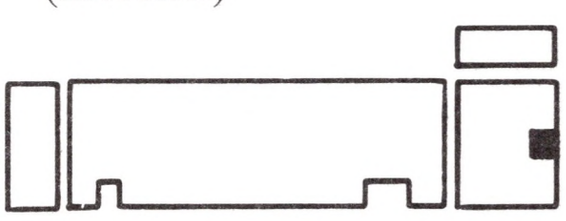
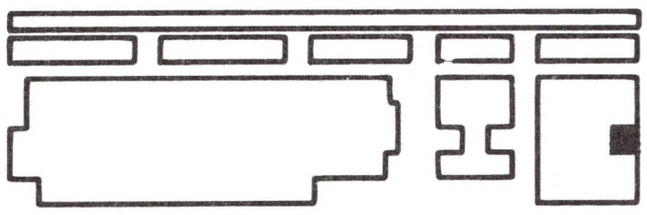
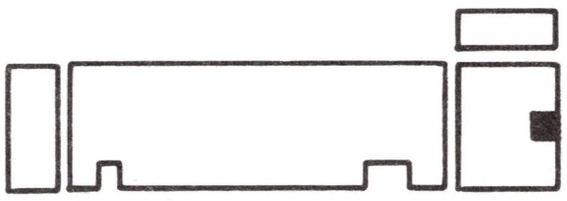
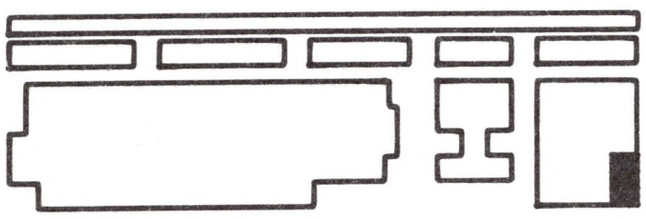
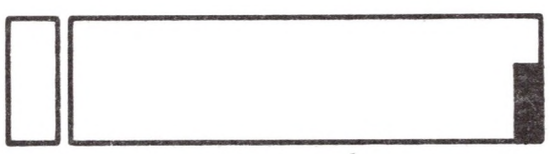
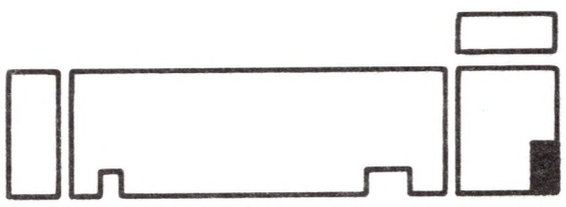
4.4.3 Arrow Keys

The functions performed by the arrow keys on the Rainbow system and the application keys F7 through F10 on the IBM system will depend on your particular application. The specific keys, and how the IBM keys map to the Rainbow 100 keys appear in the following list. An illustration showing the location of the specific key on each systems appears immediately under the key pairs.

Rainbow 100 Key	IBM PC/XT Key	IBM PC AT Key
↑	F7	F7
		
↓	F8	F8
		
←	F9	F9
		
→	F10	F10
		

4.4.4 Numeric Keypad

Specific keys, located on the numeric keypad, perform functions that depend on the type of program running on your host. This dependency applies to both the Rainbow 100 and the IBM personal computers. The SHFT/*x* combination key is used for some of the IBM functions. The particular keys (and how the IBM keys map to the Rainbow 100 keys) are shown in the following list. An illustration showing the location of the specific key(s) on both systems appears immediately under the key pairs.

Rainbow 100 Key	IBM PC/XT Key	IBM PC AT Key
PF1	Num Lock	ESC
		
PF2	SHFT/Num Lock	Num Lock
		
PF3	Scroll Lock	Scroll Lock
		
PF4	SHFT/Scroll Lock	Sys Req
		
- (dash)	- (dash)	* (asterisk)
		
, (comma)	SHFT/–	- (dash)
		
Enter	+ (plus)	+ (plus)
		

4.5 Setting VT102 Class Terminal Features Using SET-Up on the IBM PCs

SET-Up allows your IBM system to emulate features of the VT102 class terminal. Some of the features that you can set include:

- Cursor style (block or underline)
- Screen background (dark or light)
- Automatic repeating of keyboard keys

It is recommended that you do not change the settings for terminal features unless you are familiar with DECnet concepts.

4.5.1 Entering SET-Up Mode

Before you can enter SET-Up mode, a valid connection must be made to the remote system. To enter SET-Up mode, hold down the ALT key while you depress the function key F3. The computer displays the first SET-Up screen – SET-Up Directory. SET-Up features are divided into 5 areas of common use called major headings. Each major heading includes specific parameters SET-Up menus. The major headings (shown as bolded text) and their SET-Up menus appear in the following list:

Set-Up Directory

Action
Display
General
Screen
Exit

Action Set-Up

Main Directory
Clear Display
Factory Parameters
Save Parameters
Recall Parameters

Display Set-Up

Main Directory
No/Auto Wrap
Dark Text, Light Screen/Light Text, Dark Screen
No/Text Cursor
Block Cursor Style/Underline Cursor

General Set-Up

Main Directory
VT52 Mode/VT100 Mode
Return = CR/Return = Cr/Lf
Numeric Keypad/Application Keypad
Normal Cursor Keys/Application Cursor Keys

NOTE

Numeric keypad causes the auxiliary keypad to transmit ASCII character codes corresponding to the numeric characters on the key.

Application keypad causes the auxiliary keypad to transmit control codes used by an application program.

Normal cursor keys transmit ANSI cursor control sequences (up, down, left and right).

Application cursor Keys transmit application program control functions.

Screen Set-Up

Color enabled/black/white
Screen foreground color
Screen background color
Pulldown foreground color
Pulldown background color

When using SET-Up menus, be sure the screen option of either ''Color Enabled'' or ''Black/White'' is set properly for the display adapter that is installed on your pc.

4.5.2 Selecting a Parameter SET-Up Menu

Use the arrow keys (↑ up, ↓ down, ← left, and → right) to select a parameter SET-Up menu. The menu is displayed in reverse video. You can move to the specific parameter's SET-Up menu by pressing the (plus) key.

Use the appropriate arrow keys to move you to a particular parameter value. The value is displayed in reverse video. To change the value or perform the action, press the + (plus) key.

For example, to select a block cursor style, use the following steps:

1. Use the ALT/F3 key combination to enter the SET-Up Directory.
2. Select the Display Set-Up parameter with the → (right-arrow). The parameter is displayed in reverse video.
3. Press the + (plus) key to move to the Display SET-Up menu.

4. Use the appropriate arrow keys to move you to the block cursor style parameter. Press the + (plus) key to change this value. The selected value is displayed in reverse video.

5. Use the ALT/F3 key combination to exit from the SET-Up directory.

4.5.3 Saving Set-Up Values

To save the values you have selected in Set-Up mode, select the Save Parameters SET-UP menu from the Action SET-Up display. This action places the current parameters in the DECnet directory, in the file named VT102.DAT. By storing the parameters in the VT102.DAT file, SETHOST can restore saved parameters when it is run from any directory.

4.5.4 Using the Set-Up Values

If you do not save parameters after changing them, the computer will use the new values until you:

- Change the SET-Up features again.
- Exit from SETHOST or turn off the computer.
- Recall the original features by selecting the Factory Parameters from the Action SET-UP menu.
- Run an application program that changes the features.

4.5.5 Recalling the SET-Up Values

If you have saved parameters, they are automatically read when you run the SETHOST utility. To recall the values you previously stored in the VT102.DAT file in the DECnet directory, select the Recall Parameters SET-Up menu.

4.5.6 Exiting from SET-Up Mode

To exit from SET-Up mode, hold down the ALT key while you depress the function key F3. You can also use the Exit parameter from the SET-UP menu.

4.5.7 Editing Text During a Remote Session

You can edit text that is typed on the current line during a remote session by using one of the following procedures. CTRL/W and CTRL/U can be used as long as they are not required by the program you are running. The remote host may provide other special characters for editing text.

To edit text, use one of the following methods:

- Press either the <X] or the back space key to delete the last character you typed.
- Press this key to delete the last word you typed:

  CTRL/W

- Press this key to delete the line you are typing:

  CTRL/U

4.6 Disconnecting from a Host

To disconnect from the host and break the logical link connection, use one of the following methods:

1. Log off using the standard log-off procedure established for that operating system. This procedure also exits you from the SETHOST utility. For example, to log off the VAX host, type:

   $ LOGOUT RET

2. Use CTRL/\ or CTRL/4, then press the RET key. This exits from SETHOST, but allows you to save the session and retain a link to the host. If you then press the A key, the session is deleted and the link to the host is disconnected. If you are aborting the current session (or if you only have one session established), you will also exit from SETHOST.

5
Using a Remote Printer and Disk

The Network Device Utility (NDU) is a DECnet utility that controls your use of remote printers and disks as if they were directly connected to your personal computer. The disk or printer, resident on the remote node, appears to be a local device except for differences in access time. These differences depend on the type of communication services for the remote node you are using.

A remote disk file can be assigned a volume name (such as G:) and can be used by MS–DOS utilities or applications software as if it were a local hard disk volume. A remote printer is assigned the device name NPRN and can be used as if it were a local printer. Because the device only appears to be resident at the local node, it is called a virtual device.

Using NDU, you can:

- Assign a hard disk volume name to a new or existing remote file to be used as a virtual disk volume.
- Stop the use of a remote file as a virtual disk volume.
- Delete a remote disk file which had been used as a virtual disk volume.
- Assign the printer device name NPRN to a disk file at a remote node.
- Stop the use of the virtual printer at the remote node and ask that the saved text file be queued for printing.
- Obtain status information about all assigned hard disk drives, printer devices, and command default values.

The NDU controls the use of both virtual disks and virtual printers:

- Virtual Disk Volumes

  NDU allows you to access up to four virtual disk volumes at a time. You can access four volumes on the same remote node, one volume on each of four different remote nodes, or any other combination totaling four volumes. Each remote volume is a file that is treated as if it were a hard disk. The file is a binary file on the remote node which contains MS–DOS directories and files.

- Virtual Printers

  NDU allows you to access one virtual printer at a time. The remote node which provides the virtual printer saves any text you output to the printer device NPRN in a temporary file. When you stop your use of the virtual printer, the temporary file is queued to be printed at the default printer on the remote node.

This chapter is divided into two parts.

Part I describes:

- Installing the virtual device drivers (see Section 5.1).
- Controlling virtual disk volumes (see Section 5.2).
- Typing NDU commands to control virtual disk volumes (see Section 5.3).
- Getting help (see section 5.4).
- An NDU command summary (see Section 5.5), followed by a description of each NDU command for virtual disks and its syntax in alphabetical order.

Part II describes:

- Controlling remote printers (see Section 5.6).
- Typing NDU commands to control remote printers (see Section 5.7).
- An NDU command summary (see Section 5.9), followed by a description of each NDU command for controlling remote printers and its syntax in alphabetical order.

5.1 Installing the Virtual Device Drivers

Although NDU controls your use of virtual devices, two other programs actually perform the input and output for you. One program performs operations for virtual disks, while the other performs operations for virtual printers. You must install these programs (device drivers) on the DOS operating system before you can run NDU. If you do not, the following error message is displayed:

```
Disk (or Printer) driver not installed
```

You install the programs when you first install the DECnet–DOS software. The DECnet Installation Procedure (DIP) automatically installs the device drivers for you. Refer to the *DECnet–DOS Installation Guide* or the *DECnet–Rainbow Installation Guide* for more details.

5.2 Using NDU to Control Virtual Disks

Using NDU you can create a disk file on a remote node to be used as a virtual disk volume.

You can create an unlimited number of virtual disks, but only four can be used at any one time.

You enter the NDU utility by using one of the following methods:

- Method A: You enter a command string that includes NDU followed by the name of the specific NDU command. Press the RET key.

  ```
  E>NDU command(RET)
  ```

  After each NDU command is executed, the MS–DOS prompt is returned.

  ```
  E>
  ```

  This method can also be used for entering commands in batch mode. For instance, you may want to open virtual drives at startup time. To do this, include the NDU OPEN NODE *node-name* NDISK *file-name* DRIVE *drive* command in your AUTOEXEC.BAT file.

- Method B: You supply the name of the utility, NDU, and press the RET key. You cannot use this method for submitting NDU commands in batch mode.

  ```
  E>NDU(RET)
  ```

  NDU then responds with its own prompt.

  ```
  NDU>
  ```

  NOTE

  When you run NDU in batch mode, an ERRORLEVEL of 0 is returned for success. A 1 is returned for failure.

  Enter the NDU command after the NDU prompt, and press the RET key. Continue this procedure until you have entered all NDU commands.

  ```
  NDU>command(RET)
  NDU>command(RET)
  NDU>command(RET)
  ```

  To exit from NDU, enter the EXIT command and press the RET key. The system returns with the MS–DOS prompt.

  You can also use CTRL/Z RET to exit from NDU. While holding down the CTRL key, press the letter Z key. Then press the RET key.

  ```
  NDU>EXIT(RET)
  E>
  ```

5.3 Typing NDU Commands

Figure 5–1 shows an example of the parts of an NDU command:

```
NDU>CLOSE  DRIVE  G(RET)
     verb          value
           entity
```

Figure 5–1: An NDU Command

NDU commands contain a verb which specifies the action to perform and entities with values which describe the specifics of that action.

5.3.1 Command Verbs

You can abbreviate command verbs to a character string that is unique to that verb. For example, you can type **CRE** for CREATE. However, for clarity and consistency, all examples in this chapter show the full command format.

5.3.2 Entities

Table 5–1 lists the entities that NDU commands act upon when controlling virtual disks. When you specify more than one entity in a command line, use either a space or a tab to separate them.

Table 5–1: NDU Command Entities for Virtual Disks

Entity	Function
ACCESS *access_mode*	is used with the OPEN and CREATE commands to select the operating mode used to access the virtual hard disk. RW access indicates that the disk can be read from or written to, but it cannot be shared. RO access indicates that the disk can be shared, but only for the purpose of reading.
ALLOCATION *n*	specifies the initial disk file size of the virtual disk file (in blocks) on the remote node. The value of *n* is a decimal number. This entity is only valid with the CREATE command.
DRIVE *drive*	specifies a drive name to be treated as a hard disk. The name is one alphabetic character. **The drive name must be one of the names listed by the SHOW STATUS command.** When you start NDU, it displays the drives that are available by indicating they are CLOSED. You can then select G, for example, as the drive name to be used in the DOS commands and functions that address a drive.
MAX *n*	is the maximum size of the virtual disk in megabytes. The value of *n* can be 1, 10, 20, or 32. Note that 32 megabyte disks made under MS–DOS Version 2 are not compatible with 32 megabyte disks made under Version 3.

(continued on next page)

Table 5–1 (cont.): NDU Command Entities for Virtual Disks

Entity	Function
NDISK *file-name*	identifies the file name associated with a drive name. For example, NDISK MYDISK1.DSK associates the file name MYDISK1.DSK with the drive name that is addressed by a DOS command. The file name can be from 1 to 126 alphanumeric characters in length. The file name may include device and directory fields, but they must be in the proper format for the remote node's system type.
NODE *node-name* or *node-address*	identifies and locates the virtual hard disk's resident node. The node name can be from 1 to 6 alphanumeric characters in length and must contain at least one alphabetic character. You can also specify a node address for the remote node. The node's address includes a unique area number and a node number, separated by a period: *area.number* where *area* is a number in the range of 1 to 63. *number* can be in the range of 1 to 1023.
USER *access-info*	specifies access control information for a specific remote node. This is the identification information that you enter during log on at the hard disk's resident DECnet node. Each item in the identification string must be between 1 to 39 alphanumeric characters. The access control string has the following format: USER *user-name* [*/password*[*/account*]] where *user-name* is the name of the user on the specified remote node. If USER is not specified, the default access control information set by NCP will be used. *password* is the password you need to access files or programs on the specified remote node. *account* is normally not required by operating systems. Note that account information cannot be supplied unless you supply a specific password as well.

5.4 Using the HELP Command

If you need assistance in selecting NDU commands, use the HELP command. Enter the HELP command, then press RET. For example:

```
NDU>HELP(RET)
```

To obtain information about one of the subjects listed, enter HELP again plus a subject. Then press RET. For example:

```
NDU>HELP SHOW(RET)
```

5.5 NDU Command Summary for Virtual Disks

NDU provides you with command verbs that allow you to control the use of remote files as virtual disk volumes. Table 5–2 lists each NDU command and its function. The remainder of this section discusses each command in alphabetical order.

Table 5–2: NDU Commands for Virtual Disks

Command	Function
CLOSE	Requests that an existing connection to a virtual hard disk be terminated.
CREATE	Instructs NDU to create and open a new remote file and format it to be treated as a virtual hard disk.
DELETE	Deletes a remote file which had been used as a virtual hard disk.
HELP	Displays a message explaining the use of the NDU command and the meaning of the command parameters.
OPEN	Requests a connection to an existing hard disk.
SHOW	Displays the activity and current state of the virtual hard disk(s) and virtual printer. A status report is also displayed at the completion of other function requests.

5.5.1 CLOSE

The CLOSE command ends the connection between the specified drive and the virtual disk file on the remote node.

Format

CLOSE DRIVE *drive*

or

CLOSE NODE *node-name* NDISK *file-name*

where

drive is one alphabetic character that specifies the virtual drive to be acted upon.

node-name identifies and locates the virtual hard disk's resident node. The node name must be from 1 to 6 alphanumeric characters, including at least 1 alphabetic character.

You can also specify a *node-address* for the remote node. Refer to Table 5–1 to see how the node address is formatted.

file-name specifies the name of the file that is the virtual disk on the remote node. The file name can consist of 1 to 126 alphanumeric characters. It may include device and directory fields, provided it is properly formatted for the remote node's system type.

Example

```
NDU>CLOSE DRIVE G(RET)
```

This command ends the connection between the local node and the remote hard disk drive G. Any subsequent operations to drive G result in the following error message:

```
Not ready error reading drive G
Abort, Retry, Ignore?
```

5.5.2 CREATE

The CREATE command creates and formats a new remote data file, which is treated as a hard disk, and establishes a connection between a volume name at your local node and the file.

Format

CREATE NODE *node-name* NDISK *file-name* [DRIVE *drive* | USER *access-info* | ACCESS { *RW* / *RO* } | ALLOCATION *n* | MAX *n*]

where

node-name	identifies and locates the virtual hard disk's resident node.
file-name	identifies the file name associated with a drive name.
drive	specifies the name for a drive to be treated as a hard disk. The name consists of one alphabetic character. You should use the NDU SHOW command to list available names for the disk. Only the names for closed drives can be used. The default is the next available drive.
access-info	indicates access control information for a specific remote node. (See the entry under USER in Table 5–1 for a description of access control information.)
RW or *RO*	identifies the operating mode for accessing the virtual disk. RO specifies that the disk can be shared, but only for the purpose of reading. RW specifies that the disk can be read from or written to, but it cannot be shared. The default is RW.
ALLOCATION *n*	specifies the initial disk file size in blocks of the virtual disk file on the remote node. The value of *n* is a decimal number greater than the minimum virtual disk size. The default is the minimum size.
MAX *n*	specifies the maximum size to which the virtual disk can grow in megabytes. The value of *n* in this case may be 1, 10, 20, or 32. The default value is 32. Note that a 32 Mbyte hard disk created under Version 2 of the MS–DOS operating system is not compatible with a 32 Mbyte disk created under MS–DOS Version 3.0.

Remarks

The following message is displayed while NDU is creating a new virtual hard disk file. It is an informational message only. The number of blocks displayed on the screen changes as the file is being created.

```
CREATE DISK Writing Block n
```

Following a CREATE request, NDU displays status information about the hard disk. The information is followed by the NDU> prompt.

As you write files to a virtual disk, free space in the current file is used first. The file is then extended as necessary until the maximum size is reached. Deleting files from the virtual disk will NOT reduce the size of the virtual disk file. To reduce the size of a virtual disk file from which MS–DOS files have been deleted, you must create a new virtual disk and copy all files from the old disk to the new disk.

Example

```
NDU>CREATE DRIVE G NODE DUBLIN NDISK ACCTDEPT.TXT USER JONES(RET)
```

This command creates a new remote file named ACCTDEPT.TXT to be treated as a virtual hard disk, drive G. Future access to this hard disk may be restricted to USER JONES, depending on the specified access control information.

If DRIVE G already existed and was open, the following error message would be displayed:

```
Drive "G" is already OPEN: CREATE function not done.
```

5.5.3 DELETE

The DELETE command deletes an existing virtual disk file on a remote node.

Format

DELETE NODE *node-name* NDISK *file-name* [USER *access-info*]

or

DELETE DRIVE *drive*

where

node-name	identifies and locates the virtual hard disk's resident node.
file-name	identifies the hard disk's file name.
access-info	indicates access control information for a specific remote node. If the access control information is omitted, the access data specified with NCP is used instead. (See the entry under USER in Table 5–1 for a description of access control information.)
drive	is one alphabetic character that specifies the virtual drive to be acted upon.

Remarks

After a DELETE request, NDU displays status information about all the drives. The information is followed by the NDU> prompt.

Example

```
NDU>DELETE NODE MUNICH NDISK TEMPDISK.TXT RET
```

This command deletes a file named TEMPDISK.TXT, at node MUNICH. If a drive was opened with the specified virtual disk, this command closes the drive. The drive is now available for use with another virtual disk.

5.5.4 HELP

The HELP command displays information about NDU commands and entities.

Format

HELP $\left[\begin{matrix}\textit{command-verb}\\ \textit{entity}\end{matrix}\right]$

where

command-verb is the NDU command for which you want information.

entity is the NDU entity for which you want information.

Examples

```
NDU>HELP(RET)
```

The system displays a list of all NDU commands and entities for which help information is available.

```
OPEN   CLOSE   DELETE   CREATE   SHOW      EXIT  HELP
NODE   USER    DRIVE    NDISK    NPRINT    MAX  ALLOCATION    ACCESS
```

```
NDU>HELP CREATE(RET)
```

The system displays help information for the CREATE command.

5.5.5 OPEN

The OPEN command establishes a connection between your local node and an existing virtual hard disk at a remote DECnet node.

Format

OPEN NODE *node-name* NDISK *file-name* [DRIVE *drive* | USER *access-info* | ACCESS { *RW* / *RO* }]

where

node-name identifies and locates the virtual hard disk's resident node.

file-name identifies the file name associated with a drive name.

drive refers to a drive name to be treated as a hard disk. The default is the next free drive.

access-info indicates access control information for a specific remote node. If the access control information is omitted, the access data specified with NCP is used instead. (See the entry under USER in Table 5–1 for a description of access control information.)

RW or *RO* identifies the operating mode for accessing the virtual disk. RO specifies that the disk can be shared, but only for the purpose of reading. RW specifies that the disk can be read from or written to, but it cannot be shared. The default is RW.

Remarks

Following an OPEN request, NDU displays status information about the hard disk followed by the NDU> prompt.

Example

```
NDU>OPEN DRIVE G NODE DUBLIN NDISK ACCTDEPT.TXT(RET)
```

This command opens a connection between the local node and the remote DECnet node DUBLIN. At that node, an existing file named ACCTDEPT.TXT is opened for READ WRITE access. All subsequent operations directed to the hard disk drive G are sent to ACCTDEPT.TXT at node DUBLIN. If the file does not exist, the OPEN operation fails and the following error message is displayed:

```
OPEN function failed: the remote server could not find that file.
```

If the file was already open, the following error message is displayed:

```
Drive name is already open, cannot OPEN it again.
```

5.5.6 SHOW

The SHOW command causes NDU to report on the status of all virtual disk(s) and printer.

Format

SHOW STATUS

Remarks

You can display system status information through the use of DOS commands such as DIR and CHKDSK.

Examples

```
NDU>SHOW STATUS(RET)
```

Sample output from this command appears as follows:

```
Disk     Status/     Access     Node/      Socket     Reads     Writes
Drive    Ndisk                  User

E        CLOSED
F        OPEN        RW         DUBLIN     6          241       68
         ND01.NVD               JONES/PASSWORD
G        CLOSED
H        CLOSED

Printer     Status/      Node    Socket/     Writes
Device      Nprint               User

NPRN        CLOSED
```

This display includes the following information:

- The name of the disk drive.
- The drive's status – OPEN or CLOSED.
- The file name.
- The ACCESS mode – read or write.
- The NODE name, the USER name, and any account information.
- The number of the SOCKET you are connected to. (DECnet assigns you to a socket).
- The number of completed READ operations.
- The number of completed WRITE operations.
- The name of the printer.
- The printer's status – OPEN or CLOSED.

The USER and account information are displayed if you have specified them in the CREATE or the OPEN command. If you do not specify this information and it is obtained from the NCP database, it is not displayed.

5.6 Using the NDU to Control Remote Printers

The NDU utility allows you to direct text to a remote node to be queued later for printing. You enter the NDU utility by using one of the following methods:

- Method A: You enter a command string that includes NDU followed by the name of the specific NDU command. Press the RET key.

  E>NDU command(RET)

  After each NDU command is executed, the MS–DOS prompt is returned.

  E>

  This method can also be used for entering commands in batch mode.

- Method B: You supply the name of the utility, NDU, and press the RET key. You cannot use this method for submitting NDU commands in batch mode.

  E>NDU(RET)

  NDU then responds with its own prompt.

  NDU>

  Enter the NDU command after the NDU prompt, and press the RET key. Continue this procedure until you have entered all NDU commands.

  NDU>command(RET)
  NDU>command(RET)
  NDU>command(RET)

  To exit from NDU, enter the EXIT command and press the RET key. The system returns with the MS–DOS prompt.

  You can also use CTRL/Z RET to exit from NDU. While holding down the CTRL key, press the letter Z key. Then press the RET key.

  NDU>EXIT(RET)
  E>

NOTE

When you run NDU in batch mode, an ERRORLEVEL of 0 is returned for success. A 1 is returned for failure.

5.7 Typing NDU Commands

Figure 5–2 shows an example of the parts of an NDU command:

NDU>OPEN PRINTER NODE LONDON NPRINT file-name(RET)

verb entities value

Figure 5–2: An NDU Command

NDU commands act on entities and their values. When using an NDU command, you must provide the command verb, the entity, and the entity value.

5.7.1 Command Verbs

You can abbreviate command verbs to a character string that is unique to that verb. For example, you can type **CRE** for CREATE. However, for clarity and consistency, all examples in this chapter show the full command format.

5.7.2 Entities

Table 5–3 lists the entities that NDU commands act upon when controlling remote printers. When you specify more than one entity in a command line, use either a space or a tab to separate them.

Table 5–3: NDU Command Entities for Remote Printers

Entity	Function
NODE *node-name*	identifies and locates the printer's resident node. The node name can be from 1 to 6 alphanumeric characters in length and must contain at least one alphabetic character.
NPRINT *file-name*	specifies the name of the file which will contain the text, sent to the virtual printer, before it is queued for printing on the remote node. The file name can consist of 1 to 126 alphabetic characters. It may also include device and directory fields, but the name must be in the proper format for the remote node's system type.
USER *access-info*	specifies access control information for a specific remote node. This is the identification information required to access the print file on the remote node. Each item entered must be between 1 and 39 alphanumeric characters. The access control string has the following format: USER *user-name* [*/password*[*/account*]] where *user-name* is the name of the user on the specified remote node. If USER is not specified, default access control information set by NCP will be used. *password* is the password you need to access files or programs on the specified node. *account* is a 1 to 39 alphanumeric character string that consists of account information. Note that account information cannot be supplied unless you supply a specific password as well.

5.8 Using the HELP Command

If you need assistance in selecting NDU commands, use the HELP command. To view the list of HELP topics, type HELP and press the RET key. For example:

```
NDU>HELP
```

To obtain information about a specific subject, type the HELP command and a topic. Then press the RET key. For example:

```
NDU>HELP OPEN
```

5.9 NDU Command Summary for Remote Printers

NDU provides you with commands that allow you to direct text to a remote node to be queued later for printing. Table 5–4 lists each NDU command and its function. The remainder of the chapter discusses each command in alphabetical order.

Table 5–4: NDU Commands for Remote Printers

Command	Function
CLOSE	Requests that NDU end an existing connection to a remote print file and queues it for printing and deletion.
CREATE and OPEN	Instructs NDU to create a remote printer file and make it ready for use.
HELP	Displays a message explaining the use of the NDU command and the meaning of the command parameters.
SHOW	Displays the activity and current state of the virtual disk(s) and printer.

5.9.1 CLOSE

The CLOSE command ends the connection between the device NPRN and the file on the remote node. The file is then queued to be printed on the remote printer. After the file is printed, it is deleted.

Format

CLOSE PRINTER

Example

```
NDU>CLOSE PRINTER(RET)
```

This command ends the connection between the device NPRN and the printer on the remote node. Any subsequent operations to the printer device result in the following error message:

```
Not ready error writing device NPRN
Abort, Retry or Ignore?
```

5.9.2 CREATE and OPEN

The CREATE and OPEN commands function in the same way.

The CREATE and OPEN commands create a new virtual printer file and make it ready for use. The printer file is created and then linked to the device NPRN.

Format

CREATE PRINTER NODE *node-name* [NPRINT *file-name*] [USER *access-info*]

OPEN PRINTER NODE *node-name* [NPRINT *file-name*] [USER *access-info*]

where

node-name	identifies and locates the printer's resident node.
file-name	specifies the name of the remote printer file. If you omit *file-name*, a name for the remote printer file is created.
access-info	specifies access control information for the remote node. If the access control information is omitted, the access data specified with NCP is used instead. (See the entry under USER in Table 5–3 for a description of access control information.)

Remarks

Following a CREATE or OPEN request, NDU displays status information about the printer queue followed by the NDU> prompt.

Examples

```
NDU>CREATE PRINTER NODE DUBLIN NPRINT DEMO.TXT USER JONES(RET)
```

This command creates a file named DEMO.TXT to be printed at the remote DECnet node DUBLIN.

```
NDU>OPEN PRINTER NODE PARIS NPRINT LETTER.TXT USER SAM(RET)
```

This command opens a file named LETTER.TXT to be printed at the remote DECnet node PARIS.

5.9.3 HELP

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

Format

HELP $\left[\begin{matrix}\textit{command-verb} \\ \textit{entity}\end{matrix}\right]$

where

command-verb is an NDU command for which you want help information.

entity is an NDU entity for which you want help information.

Examples

```
NDU>HELP(RET)
```

The system displays a list of all NDU commands and entities for which help information is available:

```
OPEN    CLOSE    DELETE    CREATE    SHOW      EXIT  HELP
NODE    USER     DRIVE     NDISK     NPRINT    ALLOCATION    ACCESS
```

```
NDU>HELP CREATE(RET)
```

The system displays help information for the CREATE command.

5.9.4 SHOW

The SHOW command causes NDU to report on the status of all virtual disk(s) and printer.

Format

SHOW STATUS

Examples

```
NDU>SHOW STATUS(RET)
```

Sample output from this command appears as follows:

```
Disk      Status/     Access      Node/      Socket      Reads      Writes
Drive     Ndisk                   User

H         CLOSED
I         CLOSED
J         CLOSED
K         CLOSED

Printer     Status/      Node    Socket/      Writes
Device      Nprint               User

NPRN        OPEN                 1             0
            155551.TXT

NDU>
```

This display includes the following information:

- The name of the disk drive.
- The drive's status – OPEN or CLOSED.
- The file name.
- The ACCESS mode – read or write.
- The NODE name, USER name and account information string (no information displayed here).
- The number of the SOCKET you are connected to. (DOS assigns you to a socket).
- The number of completed READ operations.
- The number of completed WRITE operations.
- The name of the printer.
- The printer's status – OPEN or CLOSED.

The USER and account information are displayed if you have specified them in the OPEN or the CREATE command. If you do not specify this information and it is obtained from the NCP database, it is not displayed.

6
Using the DECnet–DOS Mail Utility

The DECnet–DOS Mail utility lets you send messages and text files to other nodes in the network. This chapter describes the Mail utility and explains how to use it.

6.1 Running the Mail Utility

The first time you start the Mail utility, you are prompted for information. The mail program uses this information each time you run the utility. The information is stored in the initialization file (INIT file) called MAIL.DAT. (This file is contained in the DECnet directory.) If the file does not exist when you start Mail, the program asks you a series of questions and creates the INIT file by storing your answers in MAIL.DAT. You can edit the information in MAIL.DAT at any time using a text editor that is installed on your system.

The initialization file can contain some or all of these mail parameters:

- REPLY__ADDRESS = *nodename*::*username*
- DEFAULT__NODE = *nodename*
- CARBON__COPY = *nodename*::*username*
- EDITOR = *pathname*
- PERSONAL__MESSAGE = a line of text

The following sections describe these parameters in more detail.

6.1.1 REPLY__ADDRESS

A DECnet–DOS node cannot receive mail from other nodes. In order for users to respond to mail that is sent from your pc node, they must have a way to return or send messages to you. The DECnet–DOS Mail utility provides a way for other users to send mail to you on a different node by using the REPLY__ADDRESS parameter.

You must define the REPLY__ADDRESS parameter when you first run the Mail utility. If not, other users will not know where to send their replies. The address consists of a node name and a username. The node name can be any node that you want to use for receiving mail. The user name should be a name that identifies you as a valid user on the specified node.

When other users receive mail from you, the REPLY__ADDRESS appears as part of the SUBJ: line in your message. The node name and user name are included in the ''Reply to'' string. For example:

```
From:   MSDOS::TOOK::BROOKS
To:     BERGIL::PORTER
SUBJ:   GROUP MEETING "Reply to TOOK::BROOKS"
```

In this example, the REPLY__ADDRESS is TOOK::BROOKS. The same address appears in the FROM: field. (MSDOS is the name of your local node.) The return address in the FROM: field may vary depending on whether or not your message had to be routed through another node. However, the ''Reply to'' field does not change. For this reason, the person responding to your message should always use the information in the ''Reply to'' field.

If you do not define a REPLY__ADDRESS, the FROM: line will reflect only the default name DECNETDOS. For example:

```
From:   MSDOS::DECNETDOS
To:     BERGIL::PORTER
SUBJ:   GROUP MEETING
```

As indicated in this example, the user Porter will not be able to answer your mail message because the necessary information has been omitted. Be sure to include both a node name and your user name when you define the REPLY__ADDRESS.

6.1.2 DEFAULT__NODE

If you often send mail to the same node, you can shorten the address for that node by defining it as the DEFAULT__NODE. Once you have defined the DEFAULT__NODE in the INIT file, you can send mail to any user on that node simply by specifying their user name. For example, you can define the DEFAULT__NODE as SPAGS by entering it in the INIT file as follows:

```
DEFAULT__NODE = SPAGS
```

Whenever you send mail from your pc node to a user on SPAGS, just include the user's name in the TO: field. For example:

```
MAIL>SEND(RET)
To:      SAM,JOHN,DAVID
Subj:    Lab work
```

In this example, the mail sender program first searches the INIT file for the name of your DEFAULT__NODE. When it finds the name SPAGS, it automatically sends your message to the users SAM, JOHN, and DAVID on node SPAGS.

You can also use the DEFAULT__NODE for routing mail messages. If you want to send mail to a node that is not defined in your NCP database, the mail program will use the DEFAULT__NODE to route your message to the desired node. For example:

```
MAIL>SEND(RET)
To:      BEORN::ANDERSON(RET)
Subj:    Weekly report
```

In this example, the node BEORN is a node that is not defined in the NCP database. In order to send your message to this address, the mail sender program requests that the DEFAULT__NODE (SPAGS) forward the message to the user ANDERSON on node BEORN.

If you do not define a DEFAULT__NODE, the mail sender program cannot route your mail messages. In addition, you will not be able to use the shortened form for mail addresses. You will have to include both the node name and the user name for any messages that you send.

6.1.3 CARBON__COPY

CARBON__COPY is the address to which a copy of your mail message is sent. Each time you send a mail message, a copy of that message will be sent automatically to the CARBON__COPY address. This address requires both a node name and a username. For example, you can define the CARBON__COPY address as DELON::BRIAN by entering it in the INIT file as follows:

```
CARBON__COPY = DELON::BRIAN
```

Now, whenever you send mail to another user, a copy of that mail will automatically be sent to the user BRIAN on node DELON. This node does not appear anywhere in your message. For example:

```
MAIL>SEND(RET)
To:      OLORIN::PETER
Subj:    Corrected problems
```

In this example, your message is sent to the user PETER on node OLORIN as well as to the user BRIAN on node DELON.

6.1.4 EDITOR

EDITOR specifies the complete path name for the editor that the mail program will use when you want to edit mail messages. You must include the /EDIT switch with a mail command if you want to edit a message or a text file before you send it.

6.1.5 PERSONAL__MESSAGE

The PERSONAL__MESSAGE is a message that other users see whenever you send them mail. The message can contain your name and phone number, or it can be any other descriptive text you wish to create. For example, you can create a message containing your name and phone extension by entering it in the INIT file as follows:

PERSONAL__MESSAGE = ''Fred at ext 7227''

When you send a mail message, your personal message appears as part of the From: line. For example:

```
From:   MSDOS::TOOK::BROOKS "Fred at ext 7227"
To:     BERGIL::PORTER
Subj:   Group Meeting "Reply to TOOK::BROOKS"
```

6.2 Starting the Mail Utility

To start the Mail utility, enter a mail command. You can enter commands using one of two methods:

1. Enter a command directly from the command line. This method allows you to include one or more of the following qualifiers with your command:
   - The subject of your message (or a description of a file if you are sending one)
   - The name of a file (if you are sending one)
   - The address of the person who will receive your message or file

   OR

2. Enter a command after the Mail prompt (MAIL>). This method allows you to include a subject, a file name, and an address, or to be prompted for this information. In addition, you have the option of editing the mail message or file before you send it.

6.2.1 Entering Mail Commands from the Command Line

When you enter mail commands from the command line, you can include the name of a file that you want to send as well as the address of the person who will receive the file. For example:

```
MAIL ACCTREPORT.TXT BARKER::JONES(RET)
```

This example sends the file, ACCTREPORT.TXT, to the user JONES on the node BARKER.

You can also include a subject for the file you want to send. For example:

```
MAIL/SUBJECT="Here is the accounting report that you requested" LF
ACCTREPORT.TXT BARKER::JONESRET
```

If you do not include an address for sending a message, the program requests the information by issuing the TO: prompt. For example:

```
MAIL/SUBJECT="Here is your accounting report" ACCTREPORT.TXTRET
TO:
```

If you do not enter a subject, a file name, or an address, the Mail program responds with its prompt, MAIL>. For example:

```
MAILRET
MAIL>
```

6.2.2 Entering Mail Commands from the Mail Prompt

If the Mail prompt (MAIL>) is already on the screen in front of you, you can enter mail commands. You can type either MAIL or SEND to send a mail message. The program prompts you for an address and then a subject. For example:

```
MAIL>MAILRET
TO:     JIMINY::BOOKERRET
SUBJ:   ProceduresRET
```

You can also type MAIL or SEND followed by the name of a file that you want to send. The program prompts you for an address and a subject. For example:

```
MAIL>SEND RECEIPTS.DATRET
TO:     CURTIN::JONESRET
SUBJ:   You need these figures for your reportRET
```

6.3 Using Distribution Lists

Distribution lists transmit mail messages to a previously defined group of people. This saves having to re-type a list of names each time you send mail to the people on the list.

If you frequently send mail to the same people, you can place their names in a distribution list file. When you then specify the list file as the destination for your mail message, the mail sender forwards the message to all of the names and addresses in the file. You can use this file as an addition to the names you have already indicated in the **To:** field.

The distribution list file contains each user's mailing address. The address consists of the user's node name and user name. The node name and the user name are always separated by a double colon (::), and each address is on a separate line in the file.

The following is a sample distribution list:

```
BERGIL::DEERE
SMAAG::JOHNS
ANTICS::M_OLDFIELD
ABRUPT::VOLKER
NEBULA::G_HART
```

To create a distribution list file, use an editor that is installed in your system. (Refer to the appropriate user's guide for the type of editor you are using.) Enter the node names and user names of the people to whom you will be sending mail. Be sure to separate the node name and the user name with two colons (::), and enter each address on a separate line.

If you do not want mail to go to some of the people in your distribution list, you can edit the list to temporarily eliminate their names. To do this, simply place an exclamation mark (!) in front of their name. The next time you specify the distribution list, the mail program will skip over their names and will not send your message to those users. Later, if you want to change the list to include the people you previously excluded, simply edit the file and remove the exclamation marks.

The following is a sample distribution list file with two names temporarily omitted from the list.

```
!BERGIL::DEERE
SMAAG::JOHNS
ANTICS::M_OLDFIELD
!ABRUPT::VOLKER
NEBULA::G_HART
```

In this sample, the users DEERE and VOLKER on nodes BERGIL and ABRUPT will not receive your mail message.

To send mail using a distribution list file, enter the at sign (@) and the name of your distribution list file in the "To:" field.

For example:

```
MAIL>SEND(RET)
TO:      FRANCE::MORGAN,WINTER::G_SMITH,@DEPART.LIS(RET)
SUBJ:    Previous Month's Earnings
```

In this example, the name of the distribution list file is DEPART.LIS. While you can use several names and addresses in the **To:** field, you can only include one distribution list file for each message you send. DECnet–DOS does not support multiple or nested distribution lists.

When using distribution list files, make sure the particular file you want to access is in the current directory. If it is not, then you must also indicate the name of the directory which contains the distribution list file you want to use.

6.4 Using Qualifiers

You can use the qualifiers /EDIT and /SUBJECT with the MAIL and SEND commands. These qualifiers can be used alone or combined together. /EDIT lets you edit the mail message before you send it. This qualifier invokes the editor you have on your system. /SUBJECT lets you include the subject of your message at the same time you enter the mail command, which bypasses the need for the SUBJECT: prompt. The following examples illustrate different possibilities for using /EDIT and /SUBJECT.

1.

```
MAIL>MAIL/EDIT(RET)
TO:     RENEWS::TAILOR(RET)
SUBJ    New information for you(RET)
```

After you press (RET) (once you have completed the subject line), the program invokes the editor and clears the screen for you to enter the text of your message.

2.

```
MAIL>SEND/SUBJECT="New information for you"(RET)
TO:     RENEWS::TAILOR(RET)
```

After you press (RET) (once you have completed the address line), you can enter the text of your message on the line immediately following the address.

3.

```
MAIL>SEND/EDIT/SUBJECT="New information for you"(RET)
TO:     RENEWS::TAILOR(RET)
```

After you press (RET) (once you have completed the address line), the program invokes the editor and clears the screen for you to enter the text of your message.

6.5 Getting Help for Mail

Help is available for the Mail utility by typing HELP at the mail prompt. For example:

```
MAIL>HELP
```

You can get help on the following topics:

EXIT	MAIL Examples
INIT FILE	SEND
MAIL	SEND Examples

Information is also available for the following mail qualifiers:

/EDIT
/SUBJECT

To see HELP text for a specific topic, type HELP followed by the topic. For example:

```
MAIL>HELP INIT FILE
```

This command displays information about the initialization file.

6.6 Mail Command Summary

The Mail utility provides the following commands for sending mail messages:

EXIT
HELP
MAIL
SEND

The following sections describe each of the mail commands in alphabetical order.

6.6.1 EXIT

The EXIT command allows you to leave the Mail utility.

Format

EXIT

Examples

In the following examples, the mail program exits you from the mail utility and returns you to the prompt or drive you were using previously.

```
MAIL>EXIT(RET)
E:\
```

or

```
MAIL>(CTRL/Z)(RET)
E:\
```

6.6.2 HELP

HELP provides information about the Mail utility. HELP topics and qualifiers include:

/EDIT	MAIL Examples
INIT FILE	SEND
EXIT	SEND Examples
MAIL	/SUBJECT

Format

HELP [*topic*]

Examples

To view the HELP file, type HELP at the mail prompt. For example:

```
MAIL>HELP
```

To view information for a specific topic, type HELP followed by the topic. For example:

```
MAIL>HELP SEND
```

This command displays information about how to send mail.

6.6.3 MAIL and SEND

Both the MAIL command and the SEND command allow you to send mail messages and text files to other users in the network. These commands may be used interchangeably.

Format

MAIL[*/switch*] [*filename*]

where

/switch is a valid MAIL (or SEND) switch. The valid switches are:

/EDIT
/SUBJECT

filename is the name of a text file to be sent as a mail message.

Remarks

To use the SEND or the MAIL command, type either command at the mail prompt. The Mail utility will first prompt you (with the To: prompt) for the name of the user (or users) who will receive your message. You can include one or several usernames, as well as the name of a distribution list file. The following example requests a mail message to be sent to the user MORGAN on node FRANCE, the user G__SMITH on node WINTER, and the addresses in the distribution list file DEPART.LIS.

Examples

```
MAIL>SEND(RET)
TO:     FRANCE::MORGAN,WINTER::G_SMITH,@DEPART.LIS(RET)
SUBJ:
```

The utility prompts you for the subject of your message (with the SUBJ: prompt).

You can avoid the SUBJ: prompt by specifying the /SUBJECT qualifier when you first enter the command. For example:

```
MAIL>SEND/SUBJECT:"Previous Month's Earnings"(RET)
TO:     FRANCE::MORGAN,WINTER::G_SMITH,@DEPART.LIS(RET)
```

The /SUBJECT qualifier can be followed by either a colon (:) or by an equal sign (=). The text of the subject must be enclosed in quotation marks (“ ”).

You can include a file specification with SEND or MAIL. This allows you to send a copy of a file to users on other nodes. For example:

```
MAIL>MAIL/SUBJECT="Monthly Report" MARCH.DAT(RET)
TO:     GUEST::GAMBOL(RET)
```

This command sends the file MARCH.DAT to the user GAMBOL on node GUEST. The subject of the file is “Monthly Report”.

If you want to edit your message or file before you send it, use the /EDIT qualifier. /EDIT calls in the editor you specified when you first set up your Mail options. When you include this qualifier, the Mail utility clears the screen for you to enter the text of your message. If you are editing a file before sending it, the text of the file appears on the screen. When you have completed your edits, exit from the editor. The mail message (or file) is then sent to all of the users you specified in the TO: section.

7
Testing the Network

This chapter describes the Network Test Utility (NTU). It describes the NTU commands that you can use to diagnose problems you may have connecting your personal computer to the network. NTU allows you to display information about your node and how it is interacting with the network. However, you cannot set or display information about other nodes in the network.

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

This chapter is divided into five sections:

- **Overview of the Network Test Utility** (see Section 7.1) – describes NTU and lists the functions you can perform with it.
- **Running the Network Test Utility** (see Section 7.2) – explains how to start NTU, and how to exit from it. This section also describes the sequence of operations performed by the utility as it tests the network components.
- **Getting Help for the Network Test Utility** (see Section 7.3) – describes the HELP commands and features that are available with NTU.
- **Using the Network Test Utility Commands** (see Section 7.4) – explains how to use the NTU commands and outlines the proper format to use for each command. This section also lists the NTU commands.
- **The Network Test Utility Commands** (see Section 7.5) – describes each of the NTU commands in detail.

7.1 Overview of the Network Test Utility

NTU is similar to the Network Control Program (NCP) because it provides a way for you to interact with the network. While you can use NCP to set network parameters and display information about your network, NTU performs loopback tests and displays information.

You can use NTU to pinpoint problems connecting your personal computer to the network. Since problems can occur in any of the network components, NTU commands let you perform various tests which check 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 of your components appear to be working properly, you should contact your network manager or the manager of the network you are trying to reach. Have the manager perform further tests to determine the network problem.)

NTU lets you use loop messages to test the following parts of your network:

- The local node
- A turnaround connector on the line or controller
- A local or a remote modem (for DDCMP use only)
- An adjacent node at the circuit level (for DDCMP use only)
- A remote node at the circuit level (for Ethernet only)
- A remote node at the application level

These messages are sent through the communications controller. DECnet–DOS V1.1 supports either Ethernet connections (using a vendor-supplied Ethernet controller) or asynchronous network connections (using the asynchronous serial port, which is included with the pc). DECnet–Rainbow V1.1 supports an asynchronous DDCMP network connection only.

Some of the NTU 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 as you test it. Loop test messages are sent to the connector and returned to the local node. If the messages do not return, or if they do not match the original messages that were sent, the problem is likely to be with the component you are currently testing.

The following figures show how you can progressively check the operation of your local node, your transceiver cable, your local modem, and your remote modem. These examples apply to DDCMP configurations only.

Figure 7–1 illustrates the use of the loopback connector on your local node. It is attached to the system unit at the controller port. With the connector in this position, you can check that your system is working properly (see Section 7.5.11).

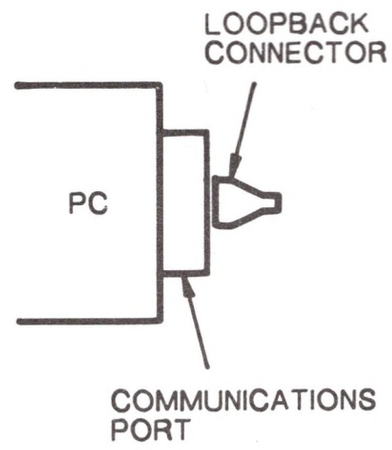


Figure 7–1: LOOP CIRCUIT Test (to the Loopback Plug on the Controller)

Figure 7–2 illustrates the use of the loopback connector on the cable that is attached to your system unit. With the connector in this position, you can check that the cable connecting your system to the network is operating properly (see Section 7.5.11).

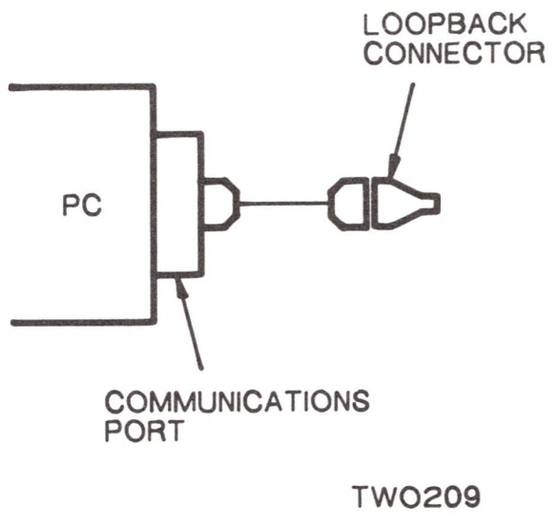


Figure 7–2: LOOP CIRCUIT Test (to the Loopback Plug on the Cable)

Figure 7–3 illustrates the local modem loopback test (see Section 7.5.11). Using this test, you can check that your network is operating properly up to the modem.

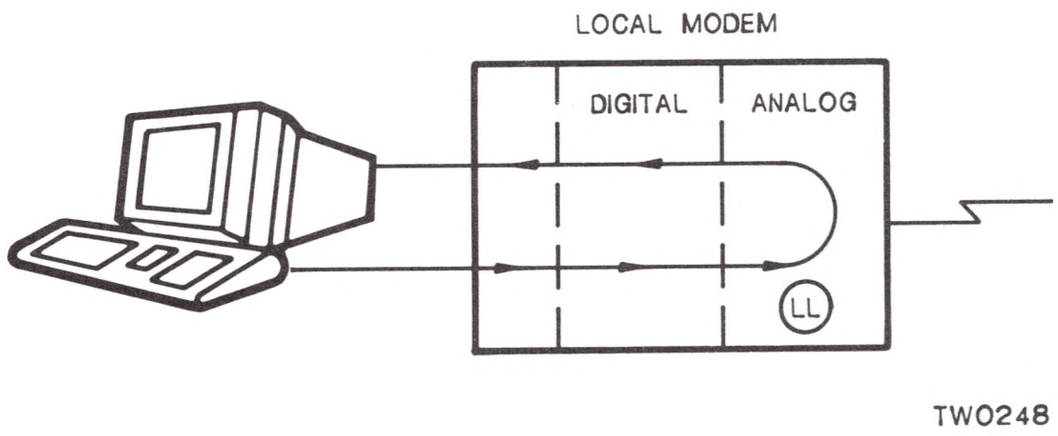


Figure 7–3: LOOP CIRCUIT Test (to the Local Modem)

Figure 7–4 illustrates the remote modem loopback test (see Section 7.5.11). Using this test, you can check that your network is working properly up to the remote modem.

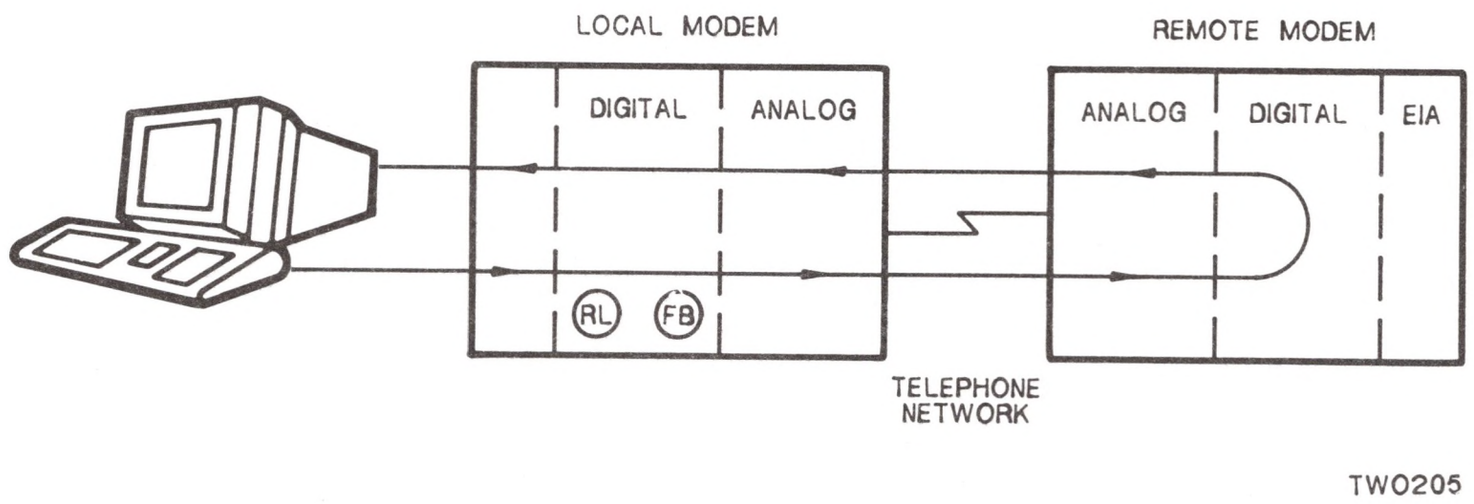


Figure 7–4: LOOP CIRCUIT Test (to the Remote Modem)

When you run the loopback tests, network counters record events that occur while the test is running. The counters record the error and traffic information that occurs 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 ZERO CIRCUIT, ZERO LINE, and ZERO EXECUTOR. When the test is run, the errors and traffic that are recorded reflect the most recent network events. To display the counters after the test has run, use one of the following commands:

- SHOW CIRCUIT COUNTERS
- SHOW LINE COUNTERS
- SHOW EXECUTOR COUNTERS

The loop tests are performed 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 retransmit it back to your node

NTU performs loop tests and analyzes their results. To perform these functions, NTU provides the following options:

- **CLEAR** resets all of the default settings that you specified with SET (see Sections 7.5.1 through 7.5.8).
- **EXIT** leaves NTU and returns you to the previous prompt (see Section 7.5.9).
- **HELP** provides a list of topics to assist you in using the NTU commands (see Sections 7.3 and 7.5.10).

- **LOOP** tests check the operation of your local node, the connection to the remote node, and the communication hardware that connects them. This hardware includes local and remote modems (see Sections 7.5.11 through 7.5.14).
- **MIRROR** is the network application that is used when a loopback test is performed from a remote node to the local node. The mirror allows test messages from a remote node to be echoed back to that same remote node (see Section 7.5.15).
- **MONITOR LOGGING** continuously displays event logging information on the terminal screen (see Section 7.5.16).
- **READ LOGGING** displays the contents of the event logging buffer (see Section 7.5.17).
- **SET** lets you specify the default settings for the LOOP commands (see Sections 7.5.18 through 7.5.27).
- **SHOW** displays statistics about node, line, and circuit characteristics. SHOW also displays counter statistics (see Sections 7.5.28 through 7.5.37).
- **ZERO** resets each of the counters to zero (see Sections 7.5.38 through 7.5.40).

NOTE

Some of the NTU commands are identical in format and function to the NCP commands of the same name described in Chapter 2. These commands are:

EXIT	SHOW EXECUTOR
MONITOR LOGGING	SHOW KNOWN LINKS
READ LOGGING	SHOW KNOWN NODES
SHOW ACTIVE LINKS	SHOW LINE
SHOW ACTIVE NODES	SHOW NODE
SHOW ADJACENT NODE	ZERO CIRCUIT
SHOW CIRCUIT	ZERO EXECUTOR
SHOW DEFAULTS	ZERO LINE

Refer to the command descriptions in Chapter 2 for complete details about the NCP commands.

7.2 Running the Network Test Utility

Before you run any of the NTU tests, NTU performs a sequence of operations to determine if there are any problems with your network components. If there are no problems, NTU displays a series of messages. It then displays the NTU prompt (NTU>). For example:

```
                    Network Test Utility (NTU) Version 1.1.0
NTU>
```

You can then select one of the test options that are available to you (see Section 7.5).

If a problem **prevents** NTU from running, the utility displays an error message describing the problem and the steps you should take to correct it. NTU then exits and returns to the previous command level. As an example, NTU will not function properly if you have not followed the installation steps correctly.

7.2.1 Starting the NTU

To invoke NTU, you can type **NTU** at the current drive prompt and then press RET. For example:

```
E>NTU RET
```

If there are no problems, the system responds with the NTU startup messages, followed by the NTU prompt. You can then enter NTU commands.

You can also use any of the following methods to start NTU:

1. Call up an indirect command file, which contains the NTU commands. For example:

   ```
   E>NTU @COMMANDS.BAT RET
   ```

   However, when NTU reaches the end of this command file, it exits. The system then returns to the previous prompt or command level.

2. Enter individual NTU commands from the command line. For example:

   ```
   E>NTU LOOP NODE REMOTE RET
   ```

   This command sends a loop message to the node named REMOTE.

3. Build a batch file and use that file to execute NTU commands. For example, you can create the file LOOP.BAT containing these commands:

   LOOP.BAT:

   ```
   NTU ZERO CIRCUIT COUNTER
   NTU ZERO LINE COUNTER
   NTU ZERO EXECUTOR COUNTER
   NTU LOOP NODE FORE
   NTU SHOW EXECUTOR COUNTER
   NTU SHOW CIRCUIT COUNTER
   NTU SHOW LINE COUNTER
   ```

   To invoke this batch file, enter the name of the file at the system prompt. For example:

   ```
   E>LOOP RET
   ```

The commands in LOOP.BAT let you zero the counters, run a loop test to the node FORE, and show the resulting counters (which reflect what happened during the loop test). You can redirect the output from the commands in LOOP.BAT to another file. For example:

```
E>NTU <LOOP.BAT >LOOP.LOG(RET)
```

or

```
E>NTU @LOOP.BAT >LOOP.LOG
```

Either of these commands redirects the output from the LOOP.BAT commands into the file LOOP.LOG.

7.2.2 Leaving the NTU

To exit from NTU, type the **EXIT** command at the NTU prompt and then press (RET). For example:

```
NTU>EXIT(RET)
```

You can also use (CTRL/Z)(RET) to exit from NTU. Either method causes you to leave NTU, and return to the prompt or command level you were using previously. NTU also sets the line state ON when it exits.

7.3 Getting HELP for NTU

The HELP command provides assistance for each of the NTU commands. To display HELP, type the command at the NTU prompt. For example:

```
NTU>HELP(RET)
```

The system displays a list of the NTU commands. To display HELP for an individual command, enter HELP and a command or a command plus an entity that applies to that command. For example:

```
NTU>HELP SET(RET)
```

NOTE

Before you can display on–line help text, the file NTUHELP.BIN must be installed in the same directory as the DECnet database (DECPARM.DAT).

7.4 Using the NTU Commands

The NTU commands are made up of several parts. When you enter a command, be sure to separate each part with a space. The parts of NTU commands include:

- The **command verb** (the command itself)
- The **entity** (a noun that defines which network component will be affected by the command)

- The **parameter** (a definition of how the command will affect the network component)
- The **qualifier** (a further definition of how the command will affect the network component)

You do not have to type the entire keyword when you enter it. The system recognizes the first three letters of each keyword. For example, to show the default settings, you enter:

```
NTU>SHO DEF(RET)
```

This helps save time if you want to enter a long command string. However, for clarity and consistency, all examples in this chapter will show the full command format.

The parameters used by the SHOW commands are defined in Table 7–1.

Table 7–1: NTU SHOW Command Parameters

Parameter	Function
CHARACTERISTICS	displays parameters that are set for the specified entity.
COUNTERS	displays error and traffic information for the specified entity. A **counter** displays statistics about the flow of traffic over the network. The counters also record error conditions and accumulate their totals for you. **Traffic** refers to the messages that travel over the network.
STATUS	displays whether the specified entity is available (ON or OFF) for communication over the network.
SUMMARY	displays only the most useful information for the specified entity. SUMMARY is the default display type.

Table 7–2 lists variables that are commonly used in NTU commands. They are described here to avoid needless repetition in the command descriptions on the following pages.

Table 7–2: Common NTU Command Variables

Qualifier	Function
account	specifies a string of 1 to 39 alphanumeric characters that you may need to access the loopback mirror on a remote node. (This field is not used by most systems, including DECnet–DOS and DECnet–Rainbow.)
circuit-id	specifies either ASYNC–1 or ETHER–1 as the circuit to use for testing network communications.
file-id	specifies the name of the output file which stores information resulting from the execution of an NTU command.
n	a decimal number.
node-id	specifies the name or the address of a node. A node name is a 1 to 6 alphanumeric character 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 to uniquely identify the node in the network.
password	specifies a unique character string (1 to 39 alphanumeric characters) that you must use to access the loopback mirror on the specified remote node.
user-id	specifies the name of the user (1 to 39 alphanumeric characters) on the specified remote node.

Most systems do not require a user-id, password, or an account for accessing the remote loopback mirror. The only time you may need to define these parameters is if you receive an error message when trying to connect to the mirror. An example of an error message would be: ''Connection Refused''. If the remote test fails and you see this type of message, you should then define the necessary information using the SET command. If the tests are successful without including any access information, there is no need for you to define it.

When you enter an NTU command, you must provide the command verb, the entity, and one or more of the parameters or qualifiers which apply to that command. Be sure to separate each of the elements that follow the command with a space. For example:

```
NTU>SHOW NODE FROG STATUS(RET)
```

This command string requests that NTU display the status of the node FROG.

7.5 NTU Command Summary

The NTU commands are presented in this section in alphabetical order. Table 7–3 summarizes the commands and their functions.

Table 7–3: NTU Command Summary

Command	Function
CLEAR ACCESS	Removes the userid, password, and the account information you previously specified with the SET or LOOP command.
CLEAR ACCOUNT	Removes accounting information you previously specified with the SET or LOOP command.
CLEAR COUNT	Resets the number of times to send the test message, which you previously changed with the SET or LOOP command. The default setting is 1.
CLEAR DEFAULTS	Restores the initial settings for each of the loopback commands, which you previously changed with the SET or LOOP command. The USER, PASSWORD, and ACCOUNT strings are reset to null (nothing specified). The value for COUNT is reset to 1. The value for LENGTH is reset to 40. The setting for WITH is reset to MIXED.
CLEAR LENGTH	Resets the length of the test message you previously changed with the SET or LOOP command. The default length is 40 bytes.
CLEAR PASSWORD	Removes the password you previously specified with the SET or LOOP command. The string is reset to null (no password specified).
CLEAR USER	Removes the userid you previously specified with the SET or LOOP command. The string is reset to null (no userid specified).
CLEAR WITH	Resets the binary format indicated for the test message to the default format, which you previously changed with the SET or LOOP command. The default format is MIXED (ones and zeroes).
EXIT	Exits you from the NTU utility.
HELP	Displays additional information about NTU commands. HELP also displays diagrams for some of the loop tests.
LOOP CIRCUIT	For asynchronous DDCMP configurations: performs a loop test on the DECnet circuit out to the connector (which can be attached to the controller, the cable, the local or remote modem, or an adjacent node). For Ethernet configurations: performs an Ethernet loopback test on the local Ethernet.

(continued on next page)

Table 7–3 (cont.): NTU Command Summary

Command	Function
LOOP EXECUTOR	Performs a loop test in your local node.
LOOP LINE CONTROLLER	Performs a loop test to the Ethernet controller.
LOOP NODE *nodeid*	Performs a loop test to a remote node, using a node name or address that you specify.
MIRROR	A network application that echoes LOOP NODE messages received from a remote node back to the remote node.
MONITOR LOGGING	Displays event logging information on your terminal screen.
READ LOGGING	Displays the contents of the event logging buffer on the screen in front of you.
SET ACCOUNT *account*	Specifies the account information to be used for accessing the mirror on a remote node.
SET CIRCUIT	Sets specific parameters for the circuit.
SET COUNT *n*	Specifies the number of times to send a test message. The number can be from 1 to 65,535.
SET ECHO	Displays the batch file as it is executed.
SET LENGTH *n*	Specifies the length of the loopback message to be sent, where *n* is the number of bytes. The value of *n* must be no greater than 1024.
SET LINE	Sets the line state to ON or OFF.
SET LINK	Detaches the sockets and breaks the logical links.
SET PASSWORD *password*	Specifies the password to be used for accessing the mirror on a remote node.
SET USER *userid*	Specifies the user information necessary for accessing a loopback mirror at the remote node.
SET WITH	Specifies the binary format to use for the loopback message you want to send. The message can consist of ONES (111111), ZEROES (000000), or a combination of the two (011001). The default format is MIXED (011001).

(continued on next page)

Table 7–3 (cont.): NTU Command Summary

Command	Function
SHOW ACTIVE LINKS	Displays information for all logical links from the executor node to other nodes.
SHOW ACTIVE NODES	Displays information about all the nodes with currently active links to the local node.
SHOW ADJACENT NODE	Displays information about the adjacent node (only if you are using asynchronous DDCMP communication).
SHOW CIRCUIT	Displays information about the DECnet circuit.
SHOW DEFAULTS	Displays the current default settings that you specified with the SET command. If you specified USER, PASSWORD, and ACCOUNT parameters with the SET command, these parameters are also displayed. If you are recording test results, this command displays the name of the file being used to store the results.
SHOW EXECUTOR	Displays executor (local node) information.
SHOW KNOWN LINKS	Displays information for all logical links known to the local node.
SHOW KNOWN NODES	Displays a list of all the nodes known to the local node. You can set these nodes by using NCP.
SHOW LINE	Displays information about the DECnet line.
SHOW NODE *nodeid*	Displays information about a specific node.
ZERO CIRCUIT	Resets the circuit counters to zero.
ZERO EXECUTOR	Resets the executor (local node) counters to zero.
ZERO LINE	Resets the line counters to zero.

7.5.1 CLEAR ACCESS

The CLEAR ACCESS command removes the user-id, password, and accounting information that was stored by a previous SET or LOOP command.

Format

CLEAR ACCESS

Example

```
NTU>CLEAR ACCESS(RET)
```

7.5.2 CLEAR ACCOUNT

The CLEAR ACCOUNT command clears the accounting information that was stored by a previous SET or LOOP command.

Format

CLEAR ACCOUNT

Example

```
NTU>CLEAR  ACCOUNT RET
```

7.5.3 CLEAR COUNT

The CLEAR COUNT command resets the number that indicates how many times you want to repeat a specific test to its default value of 1.

Format

CLEAR COUNT

Example

```
NTU>CLEAR  COUNT(RET)
```

7.5.4 CLEAR DEFAULTS

The CLEAR DEFAULTS command resets all the loopback parameters to their original values. The original values are: null for *user-id*, *password*, and *account*; 1 for COUNT, 40 for LENGTH, and MIXED for WITH.

Format

CLEAR DEFAULTS

Example

```
NTU>CLEAR DEFAULTS(RET)
```

7.5.5 CLEAR LENGTH

The CLEAR LENGTH command resets the number that indicates the message length to its default value of 40.

Format

CLEAR LENGTH

Example

```
NTU>CLEAR  LENGTH(RET)
```

7.5.6 CLEAR PASSWORD

The CLEAR PASSWORD command removes the password that was stored by a previous SET or LOOP command.

Format

CLEAR PASSWORD

Example

```
NTU>CLEAR  PASSWORD(RET)
```

7.5.7 CLEAR USER

The CLEAR USER command removes the user-id that you initially stored with the SET or LOOP command.

Format

CLEAR USER

Example

NTU>CLEAR USER(RET)

7.5.8 CLEAR WITH

The CLEAR WITH command resets the binary format indicated for the test message to its default value of MIXED.

Format

CLEAR WITH

Example

The following example resets the default format to be MIXED.

```
NTU>CLEAR WITH(RET)
```

7.5.9 EXIT

The EXIT command lets you leave NTU. Whenever you issue this command, NTU attempts to restore the DECnet line to the **ON** state.

Format

EXIT

Examples

NTU>EXIT(RET)

You can also use (CTRL/Z) to exit from NTU.

NTU>(CTRL/Z) (RET)

7.5.10 HELP

The HELP command provides information about the NTU commands. Entering HELP alone results in a display of the available commands. You can further qualify the information you seek by specifying HELP followed by a command name (and optionally an entity).

Format

HELP [*command_verb* [*entity*]]

where

command verb — specifies the NTU command for which you would like help information.

entity — specifies that you would like information about the specified command as it relates to this component.

Example

```
NTU>HELP SET COUNT RET
```

This example displays help text for the SET COUNT command.

7.5.11 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 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, 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 run 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.

NTU performs each test *n* times, as long as the test is successful. (The default value for *n* is 1. You can change this value using the SET COUNT command.) When the test is complete, NTU displays a success message. If the test fails, NTU displays an error message.

Format (for DDCMP Configurations)

LOOP CIRCUIT [COUNT *n*] [WITH ONES/ZEROES/MIXED] [LENGTH *n*]

where

COUNT n	is the number of times to repeat the loop test. The default is 1.
MIXED/ONES/ZEROES	indicates that the message type should be mixed ones and zeroes, all ones, or all zeroes. The default is MIXED.
LENGTH n	is the length of the loop message to be sent. The maximum value you can use is 512. The default is 40.

Format (for Ethernet Configurations)

LOOP CIRCUIT [COUNT *n*] [WITH ONES/ZEROES/MIXED] [LENGTH *n*]

or

LOOP CIRCUIT NODE *node-id*
[COUNT *n*] [WITH ONES/ZEROES/MIXED] [LENGTH *n*]

or

LOOP CIRCUIT PHYSICAL–ADDRESS *hh–hh–hh–hh–hh–hh*
[COUNT *n*] [WITH ONES/ZEROES/MIXED] [LENGTH *n*]

where

COUNT n	is the number of times to repeat the loop test. The default is 1.
MIXED/ONES/ZEROES	indicates that the message type should be mixed ones and zeroes, all ones, or all zeroes. The default is MIXED.
LENGTH n	is 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*	is the name of the designated node to which you want to send loop messages. You can use either the node's name or its address. The node must already be defined by the NCP command SET NODE.
PHYSICAL-ADDRESS *hh–hh–hh–hh–hh–hh*	is a hexadecimal number representing the actual physical address of the node you are trying to contact. The range for this number is 0 – F.

Remarks

When you issue the LOOP CIRCUIT command, NTU determines whether you are using DDCMP or Ethernet on your node. The prompts that NTU returns for this command will vary depending upon your node's configuration.

When you run the loop circuit test for DDCMP, NTU prompts you with information needed to run the test. You may 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.

When you run the loop circuit test for Ethernet, you do not need to perform any special operations. If you issue the command without specifying a responding node, a loop request is sent out to the multicast loopback assistance address. This means that any node on the Ethernet may reply to the loop request. Once a node responds to this request, NTU displays information about the loop message that was sent and the number of the responding node.

Examples

The following example illustrates a DDCMP loop circuit test.

```
NTU>LOOP CIRCUIT(RET)

LOOP CIRCUIT (DDCMP) - Do one of the following, then press any key to
begin test. Check the DECnet-DOS User's 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.
```

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

```
NTU>LOOP CIRCUIT(RET)
LOOP CIRCUIT test started at 1-Feb-1986 14:22:34

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 1-Feb-1986 14:22:35

NTU>
```

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

```
NTU>LOOP CIRCUIT NODE 55.101 COUNT 3(RET)
LOOP CIRCUIT test started at 4-Feb-1986 15:23:57

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 4-Feb-1986 15:23:58

NTU>
```

7.5.12 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, NTU performs the test *n* times, as long as the test is successful. (The default value for *n* is 1. You can change this value using the SET COUNT command.)

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

Format

LOOP EXECUTOR [COUNT *n*] [WITH MIXED/ONES/ZEROES] [LENGTH *n*]

where

COUNT n	is the number of times to repeat the loop test. The default is 1.
MIXED/ONES/ZEROES	indicates that the message type should be mixed ones and zeroes, all ones, or all zeroes. The default is MIXED.
LENGTH n	is the length of the loop message to be sent. The maximum value you can use is 512. The default is 40.

Example

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

```
NTU>LOOP EXECUTOR COUNT 10 WITH MIXED LENGTH 50(RET)
```

7.5.13 LOOP LINE CONTROLLER

The LOOP LINE CONTROLLER command is used in Ethernet configurations only. It performs a loopback test within the local node at the device (or controller) level.

The loop line controller test requires the use of the DECNA loopback connector (part number 12–22196–01). You must attach this connector to the Ethernet controller in order to perform this test.

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

Format

LOOP LINE CONTROLLER

Example

The following example illustrates a sample loop line controller test.

```
NTU>LOOP LINE CONTROLLER(RET)

LOOP LINE CONTROLLER (ETHERNET)
1. Place the loopback plug on the controller.
2. Press any key to begin test.
3. At the end of the test, reconnect the controller to the network.

LOOP LINE CONTROLLER test started at 5-Feb-1986 11:08:47

LOOP LINE CONTROLLER test finished successfully at 5-Feb-1986 11:08:49

NTU>
```

7.5.14 LOOP NODE

The LOOP NODE command invokes the loop test for 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 that are necessary for accessing the loopback mirror on the remote node. These parameters may include the node-id, the user-id, the password, and the account. You can enter these parameters before you run the test by using SET commands (see Chapter 2).

When you issue this command, NTU performs the test *n* times, as long as the test is successful. (The default value for *n* is 1. You can change this value using the SET COUNT command.)

Before running the test, LOOP NODE sets the line state to ON. When the test is complete, NTU displays a success message. If the test is unsuccessful, NTU displays an error message.

Format

LOOP NODE *node–id* [USER *user-id* | PASSWORD *password* | ACCOUNT *account* | COUNT *n* | WITH ONES/ZEROES/MIXED | LENGTH *n*]

where

node-id specifies the name or the address of a remote node. A node name is a 1 to 6 alphanumeric character 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 to uniquely identify the node in the network.

user-id is the user information you may use for accessing the loopback mirror on the remote node.

password is a unique character string that you may 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 only 8 characters).

account is alphanumeric information you may need to access the remote node's loopback mirror.

COUNT n is the number of times to repeat the loop test.

ONES/ZEROES/MIXED indicates that the message type should be all ones, all zeroes, or mixed ones and zeroes.

LENGTH n is the length of the loop message to be sent.

Remarks

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

Example

The following example specifies the node GRAHAM as the remote node to be used in the remote node test. It specifies that the loop message should be repeated 10 times, with a length of 40. This example also illustrates a possible system response to the command.

```
NTU>LOOP NODE GRAHAM COUNT 10 LENGTH 40(RET)
Connect complete to node GRAHAM

Remote node maximum buffer size for loopback:  4096

LOOP NODE test started at 4-FEB-1986  14:32:14

Sending loop message 1, 40 bytes.
Receiving loop message 1, 40 bytes.
Successful send and receive, message 1.
Sending loop message 2, 40 bytes.
Receiving loop message 2, 40 bytes.
Successful send and receive, message 2.
Sending loop message 3, 40 bytes.
Receiving loop message 3, 40 bytes.
Successful send and receive, message 3.
Sending loop message 4, 40 bytes.
Receiving loop message 4, 40 bytes.
Successful send and receive, message 4.
Sending loop message 5, 40 bytes.
Receiving loop message 5, 40 bytes.
Successful send and receive, message 5.
Sending loop message 6, 40 bytes.
Receiving loop message 6, 40 bytes.
Successful send and receive, message 6.
Sending loop message 7, 40 bytes.
Receiving loop message 7, 40 bytes.
Successful send and receive, message 7.
Sending loop message 8, 40 bytes.
Receiving loop message 8, 40 bytes.
Successful send and receive, message 8.
Sending loop message 9, 40 bytes.
Receiving loop message 9, 40 bytes.
Successful send and receive, message 9.
Sending loop message 10, 40 bytes.
Receiving loop message 10, 40 bytes.
Successful send and receive, message 10.

LOOP NODE test finished at 4-FEB-1986 14:32:44

NTU>EXIT(RET)
```

7.5.15 MIRROR

The MIRROR command is used for the LOOP NODE test. MIRROR allows the LOOP NODE test to be run from a remote system which supports the NCP loop node test to your local node. The maximum size of a message that NCP will loop back is 512 bytes.

Not all loop tests are available on all remote nodes. The tests you can use depend on which tests are available at the remote node you want to use. 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), which echoes test messages that it receives from a remote node back to that same remote node.

Format

MIRROR

Remarks

When the test run from the remote node is complete, you can press any key to return to the NTU command level.

Example

```
NTU>MIRROR(RET)
```

7.5.16 MONITOR LOGGING

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 a network support person if network problems occur. Refer to READ LOGGING for more information on event recording.

You can stop the display of information logging by pressing any key on the keyboard. (The logging will also stop when there are no more events to be displayed.) If you want to stop the display and also exit from NTU, type CTRL/C. The system requests verification with the prompt, "Terminating process?". If you answer YES, NTU exits and returns you to the prompt or system you were using previously. If you answer NO, monitoring continues.

For DECnet–DOS systems, event logging is always turned on, and you cannot select only certain events to be displayed. After the events are displayed or redirected, they are then discarded.

If an event is lost, NTU displays an error message before displaying the rest of the logged events:

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

Each event can display the following information:

```
Event type nn.nn Name of event type
Occurred dd-mmm-yyyy hh:mm:ss
Other event-dependent text
.
.
.
```

Format

MONITOR LOGGING [TO *file-id*]

where

TO *file-id*	see definition in Table 7–2.

Example

```
NTU>MONITOR LOGGING(RET)
```

The system responds with:

```
Events logged as of 17-Jan-1986 11:17:41
Event type 4.10 Circuit up
Occurred 17-Jan-1986 11:18:00
Adjacent Node: 55.2 (LOWEND)

Event type 4.7 Circuit down - circuit fault
Occurred 17-Jan-1986 11:18:09
Adjacent Node: 55.2 (LOWEND)
Reason: Line synchronization lost

Event type 4.10 Circuit up
Occurred 17-Jan-1986 11:18:15
Adjacent Node: 55.2 (LOWEND)
.
.
.
```

7.5.17 READ LOGGING

The READ LOGGING command causes the system to display information about network events that is contained in the event logging buffer. These events are logged continuously. READ LOGGING lets you read the most current network information. The information appears on the screen in front of you.

After events are displayed, they are discarded unless you use the TO file switch to store network information in a file.

If an event is lost, NTU displays the following error message:

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

Event messages have the following format:

```
Event type nn.nn event-description
Occurred dd-mmm-yyyy hh:mm:ss
event-dependent text
```

Format

READ LOGGING [TO *file-id*]

where

TO *file-id* see definition in Table 7–2.

Example

The following example specifies NETWORK.LOG as the file name to use for storing current information about network activities.

```
NTU>READ LOGGING TO NETWORK.LOG(RET)
```

7.5.18 SET ACCOUNT

The SET ACCOUNT command lets you specify the account to use for accessing the mirror on a remote node. You can use up to 39 characters to define the account.

If you want the characters representing the account to be stored in uppercase letters, simply type the characters. If you do not want the characters to be stored as uppercase, place the string in quotation marks. For example:

abc will be stored in the database as **ABC**

"Aab" will be stored in the database as **Aab**

Format

SET ACCOUNT *account*

where

account	is alphanumeric information you may need to access the adjacent node's loopback mirror.

Example

The following example specifies SAM as the account to use for accessing the remote node's mirror.

```
NTU>SET ACCOUNT SAM(RET)
```

7.5.19 SET CIRCUIT

The SET CIRCUIT command defines or modifies circuit information. Most parameters are set by your network coordinator. Therefore, you must obtain this information from that person.

Format

SET CIRCUIT [*circuit-id*] { HELLO TIMER *seconds* | OWNER {DECnet / MOP} | SERVICE {DISABLED / ENABLED} | STATE {ON / OFF} }

where

CIRCUIT [*circuit-id*] — specifies the circuit (either ASYNC–1 or ETHER–1) for which parameters are to be defined or modified.

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.

OWNER — specifies the circuit owner.

- DECnet — is the default setting.
- MOP — is required for running loopback tests.

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.

- 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

```
NTU>SET CIRCUIT ASYNC-1 OWNER DECnet RET
```

This command identifies the owner of circuit ASYNC–1 as DECnet.

7.5.20 SET COUNT

The SET COUNT command lets you specify the number of times to repeat any of the NTU tests. The default value is 1, but you can change this at any time using SET COUNT. You can assign any value that is in the range of 1 to 32,767.

The value you set remains in effect for each test until you reset it using this command, a loop command, or a CLEAR command, or until you exit from NTU.

Format

SET COUNT *n*

where

n is the numerical value you assign for the number of times to repeat each test.

Example

The following example specifies that the NTU tests be repeated 8 times.

```
NTU>SET COUNT 8(RET)
```

7.5.21 SET ECHO

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

Format

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

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.

Example

```
NTU>SET ECHO ON(RET)
```

This command causes the redirected input from a file to be displayed on the screen.

7.5.22 SET LENGTH

The SET LENGTH command lets you specify the length of the test messages to be sent over the network. The default value is 40, but you can change this at any time using SET LENGTH. You can assign any value from 1 up to 512.

The value you set remains in effect for each test until you reset it using this command, a loop command, or a CLEAR command.

Format

SET LENGTH *n*

where

n is the numerical value you assign for the length of each message.

Example

The following example specifies the length of each test message to be 100 bytes.

```
NTU>SET LENGTH 100(RET)
```

7.5.23 SET LINE STATE

This command allows you to turn the state of the line to **OFF** or **ON**. The line is the communication path that connects your node to the adjacent node. When the line state is OFF, no communication is possible.

Setting the line state to OFF allows you to disconnect cables and attach the loopback connector (in order to run the loop tests) without disrupting the performance of other devices in your network.

NOTE

If you are performing a loop test to a remote node, do NOT set the line state to OFF.

When you have completed testing, you can reset the line state to ON. You can then continue to use other NTU commands.

Format

SET LINE [*line-id*] STATE $\begin{Bmatrix} \text{OFF} \\ \text{ON} \end{Bmatrix}$

Remarks

The SET LINE STATE command does not change any of the line parameters that are currently in effect. To change line parameters, you must exit from NTU. You can then use the SET LINE command while you are in NCP. See Chapter 2 for information about setting and changing line parameters.

Example

The following example sets the line state to OFF.

```
NTU>SET LINE STATE OFF(RET)
```

7.5.24 SET LINK

The SET LINK command terminates logical link connections that are associated with a specific socket. The socket is also detached.

Format

SET LINK *socket* STATE OFF

where

LINK *socket* specifies the socket to be detached.

STATE OFF turns the logical link off.

Remarks

Use this command only if you are familiar with the socket level programming interface. If the socket option SO__KEEPALIVE has been set, the socket cannot be terminated. Refer to the *DECnet–DOS Programmer's Reference Manual* for additional information.

Example

```
NTU>SET LINK 3 STATE OFF(RET)
```

This command detaches socket 3 and breaks its logical links. To determine the socket number, use the SHOW KNOWN LINKS command.

7.5.25 SET PASSWORD

The SET PASSWORD command lets you specify the password to use for accessing a mirror on the remote node. You can use up to 39 characters to define the password.

If you want the characters representing the password to be stored in uppercase letters, simply type the characters. If you do not want the characters to be stored as uppercase, place the string in quotation marks.

For example:

Money will be stored in the database as **MONEY**

"Book" will be stored in the database as **Book**

Format

SET PASSWORD *password*

where

password	is the unique string of characters you need to access the remote node's mirror. The password can consist of 1 to 39 alphanumeric characters.

Remarks

Some systems restrict the length of passwords to a maximum of 8 characters.

Example

The following example specifies EARNEST as the password to use for accessing the mirror on the remote node.

```
NTU>SET PASSWORD EARNEST(RET)
```

7.5.26 SET USER

The SET USER command lets you specify the access control information necessary for accessing the mirror on the remote node. If you decide to provide the user's name, you must also provide the appropriate password for that name.

If you want the characters representing the user's name or address to be stored in uppercase letters, simply type the characters. If you do not want the characters to be stored as uppercase, place the string in quotation marks.

For example:

Worthy will be stored in the database as **WORTHY**

"WinTer" will be stored in the database as **WinTer**

Format

SET USER [*user-id*]

where

user-id is the user information required to access the mirror on the remote node. This information can consist of 1 to 39 alphabetic characters.

Example

The following example specifies WARDEN as the user who receives the test message at the remote node.

```
NTU>SET USER WARDEN(RET)
```

7.5.27 SET WITH

The SET WITH command lets you specify the type of format to use for sending loop-back messages. The format is always binary (ONES and ZEROES), but you have a choice as to how the binary numbers are arranged. There are three ways to arrange the binary format:

- All ONES (11111111)
- All ZEROES (00000000)
- Mixed ONES and ZEROES (01010101)

The default format is MIXED.

Format

SET WITH $\left\{\begin{matrix}\text{ONES}\\\text{ZEROES}\\\text{MIXED}\end{matrix}\right\}$

where

ONES	indicates you want to send only ONES (11111111) in the test message.
ZEROES	indicates you want to send only ZEROES (00000000) in the test message.
MIXED	indicates you want to send both ONES and ZEROES (01010101) in the test message. This is the default.

Examples

The following example specifies a mixture of ONES and ZEROES as the format to use in the test message.

```
NTU>SET WITH MIXED(RET)
```

The following example specifies that only ONES be used as the format for the test message.

```
NTU>SET WITH ONES(RET)
```

7.5.28 SHOW ACTIVE LINKS

The SHOW ACTIVE LINKS command displays summary type information for all active links between the executor node and the adjacent node (or other remote nodes with active logical links to your node).

Format

SHOW ACTIVE LINKS [TO *file-id*]

Example

```
NTU>SHOW ACTIVE LINKS(RET)
```

7.5.29 SHOW ACTIVE NODES

The SHOW ACTIVE NODES command displays the executor node, the adjacent node, and a list of all the nodes that currently have active links to your local node.

Format

SHOW ACTIVE NODES [TO *file-id*]

Example

The following example displays a list of all the active nodes that currently have logical links to your node.

```
NTU>SHOW ACTIVE NODES(RET)
```

7.5.30 SHOW ADJACENT NODE

The SHOW ADJACENT NODE command displays information for the adjacent node. This command only applies to asynchronous DDCMP communication.

Format

SHOW ADJACENT NODE [TO *file-id*]

Example

```
NTU>SHOW ADJACENT NODE(RET)
```

7.5.31 SHOW CIRCUIT

The SHOW CIRCUIT command displays information about the specified circuit.

Format

SHOW CIRCUIT [*circuit-id*] [CHARACTERISTICS / COUNTERS / STATUS / SUMMARY] [TO *file-id*]

where

circuit-id	identifies the circuit (either ASYNC–1 or ETHER–1) for which you want information displayed.
CHARACTERISTICS COUNTERS STATUS SUMMARY	see definitions in Table 7–1.

Remarks

Circuit counters monitor traffic and errors on individual circuits. You can periodically check the counters for the circuit to assess its performance and to determine if there are any problems.

Example

The following example displays the counters for your circuit.

```
NTU>SHOW CIRCUIT ASYNC-1 COUNTERS(RET)
```

7.5.32 SHOW DEFAULTS

The SHOW DEFAULTS command displays all of the current default values set either by the system or with the SET or LOOP command. SHOW DEFAULTS displays the values for the following:

- Loop Count
- Data Length
- Data Type
- User-id (if applicable)
- Password (if applicable)
- Account (if applicable)

Format

SHOW DEFAULTS

Example

The following example displays all the defaults that are currently set for use with the NTU commands.

```
NTU>SHOW DEFAULTS(RET)
NTU Defaults

Loop Count     =  1       User-id    =  ""
Data Length    = 40       Password   =  ""
Data Type      = MIXED    Account    =  ""

NTU>
```

7.5.33 SHOW EXECUTOR

The SHOW EXECUTOR command displays information about the local node.

Format

SHOW EXECUTOR [CHARACTERISTICS | COUNTERS | STATUS | SUMMARY] [TO *file-id*]

where

CHARACTERISTICS
COUNTERS
STATUS
SUMMARY
: see definitions in Table 7–1.

Example

The following example displays the characteristics of your executor node.

```
NTU>SHOW EXECUTOR CHARACTERISTICS(RET)
```

7.5.34 SHOW KNOWN LINKS

The SHOW KNOWN LINKS command displays summary type information for all logical links known to the local node.

Format

SHOW KNOWN LINKS [TO *file-id*]

Example

```
NTU>SHOW KNOWN LINKS(RET)
```

7.5.35 SHOW KNOWN NODES

The SHOW KNOWN NODES command displays a list of all nodes that are known to DECnet on your local node. You can set these nodes by using NCP. You can perform NTU tests (those that require the use of node names) only to nodes that are known by your system.

Format

SHOW KNOWN NODES [TO *file-id*]

Example

The following example displays the nodes that have been defined in your network database.

```
NTU>SHOW KNOWN NODES(RET)

Node Volatile Summary as of 1-FEB-1986 2:31:18

Node              Node      Active   LAT      Account
Address           Name      Links              Information

4.10              SMAUG     0        L
4.19              ELROND    0
4.30              OLORIN    0        M
4.209             BAGELS    0
7.110             KL1026    0
7.116             KL2116    0        LM
7.120             KL2102    0
55.2              LOWEND    0        /TESTER/password/TESTACCT
55.165            GODIVA    0
```

7.5.36 SHOW LINE

The SHOW LINE command displays information about the DECnet line (ASYNC–1 or ETHER–1) that you are using to communicate over the network.

Format

SHOW LINE [*line-id*] [CHARACTERISTICS | COUNTERS | STATUS | SUMMARY] [TO *file-id*]

where

line-id identifies the line (ASYNC–1 or ETHER–1) for which you want information displayed. Since there is only one line to use for each configuration in both DECnet–DOS V1.0 and DECnet–DOS V1.1, the *line-id* is optional.

CHARACTERISTICS
COUNTERS
STATUS
SUMMARY — see definitions in Table 7–1.

Remarks

Line counters monitor traffic and errors on individual lines. You can periodically check the counters for the line to assess its performance and to determine if there are any problems.

Example

The following example displays status information about your line.

```
NTU>SHOW LINE STATUS(RET)
```

7.5.37 SHOW NODE

The SHOW NODE command displays information about a remote node, which you specify by its name or address. The node information appears on the screen in front of you.

Format

SHOW NODE *node-id* [STATUS / SUMMARY] [TO *file-id*]

where

node-id — specifies the name or the address of the node for which you want to display information. A node name is a 1 to 7 alphanumeric character 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 to uniquely identify the node in the network.

STATUS
SUMMARY — see definitions in Table 7–1.

Remarks

If you are checking the status of a particular node, that node's name and address must be in the node database. You can use NCP to establish a node database.

Example

The following example displays the characteristics for the node GRAHAM.

```
NTU>SHOW NODE GRAHAM CHARACTERISTICS(RET)
```

7.5.38 ZERO CIRCUIT

The ZERO CIRCUIT command resets all the counters on the circuit to zero. The counters display statistics about the flow of network traffic and messages as they travel in and out of your node. The counters also record error conditions and accumulate their totals for you.

Format

ZERO CIRCUIT [*circuit-id*] [COUNTERS]

where

circuit-id specifies the circuit (either ASYNC–1 or ETHER–1) for which counters are to be zeroed. Since there is only one circuit to use for each configuration in both DECnet–DOS V1.0 and DECnet–DOS V1.1, the *circuit-id* is optional.

Remarks

When you zero the counters, any information about error conditions and their totals is erased.

Example

The following example resets the counters for your circuit to zero.

```
NTU>ZERO CIRCUIT ETHER-1(RET)
```

7.5.39 ZERO EXECUTOR

The ZERO EXECUTOR command resets all the counters for your local node to zero. The counters display statistics about the flow of network traffic and messages as they travel in and out of your node. The counters also record error conditions and accumulate their totals for you.

Format

ZERO EXECUTOR [COUNTERS]

Remarks

When you zero the executor counters, any information about error conditions and their totals is erased.

Example

The following example resets the counters for your local node to zero.

```
NTU>ZERO EXECUTOR(RET)
```

7.5.40 ZERO LINE

The ZERO LINE command resets all the counters for your line to zero. The counters display statistics about the flow of network traffic and messages as they travel in and out of your node. The counters also record error conditions and accumulate their totals for you.

Format

ZERO LINE [*line-id*] [COUNTERS]

where

line-id specifies the line (either ASYNC–1 or ETHER–1) for which counters are to be zeroed. Since there is only one line to use for each configuration in both DECnet–DOS V1.0 and DECnet–DOS V1.1, the *line-id* is optional.

Remarks

When you zero the counters, any information about error conditions and their totals is erased.

Example

The following example resets all the counters for your line to zero.

```
NTU>ZERO LINE ETHER-1(RET)
```

A
NCP Error Messages

A.1 Introduction

Some common NCP error messages are listed on the following pages. An explanation is also included with each error message.

A.2 Types of NCP Error Messages

NCP error messages fall into three categories:

- **Command line errors.** These messages are displayed when you use the incorrect syntax for a command line.
- **File Input/Output (I/O) errors.** These messages are displayed when you try to access MS–DOS files.
- **Network errors**. These messages are displayed when you try to run NCP on a system where network support, or some part of it, is missing or not working properly.

You may also see some NTU error messages while running NCP. The NTU messages are also included in this appendix.

A.3 Command Line Errors

The messages listed in this section are all related to accessing MS–DOS files.

Cannot set DDCMP parameter in Ethernet configurations.

You are running DECnet–DOS over an Ethernet connection, and you tried to set a parameter used only by DDCMP asynchronous lines.

Action to be taken: Use a different parameter, or reconfigure your connection for Ethernet communication.

Cannot set Ethernet parameter in DDCMP configurations.

You are running DECnet–DOS on an asynchronous line, and you tried to set a parameter used by Ethernet.

Action to be taken: Use a different parameter, or reconfigure your connection for DDCMP communications.

Could not find node address with node name = *node name.*

The node address and node name pair did not match.

Action to be taken: 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.*

NCP was unable to find the node name defined by the specific node address.

Action to be taken: Check your list of node names and node addresses and try to execute the NCP command again.

Executor is in wrong state.

The executor is not in the proper state for changing the parameter you specified. (For example: you cannot set the EXEC NAME or EXEC ADDR when the state is ON.)

Action to be taken: Use the DEFINE EXECUTOR command rather than SET EXECUTOR.

Line is in wrong state.

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.)

Action to be taken: Change the line state first, then change the parameter.

Node address must be in the range 1 to 1023.

The node address has exceeded the valid range.

Action to be taken: Enter a node address that is in the range of 1 to 1023.

Node area must be in the range 1 to 63.

The area number has exceeded the valid range.

Action to be taken: Enter a node area that is in the range of 1 to 63.

Node name already defined for another node number.

You tried to define a node name that has already been matched to a different node number.

Action to be taken: Select another name to apply to this node number.

Node name must contain an alphabetic character.

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.

Action to be taken: Re-define the node name using at least one letter.

Node number already defined for another node name.

You tried to define a node number that has already been matched to a different node name.

Action to be taken: Select another number to apply to this node name.

Value out of range; check the documentation or type HELP.

The value you selected for this parameter is not within the valid range.

Action to be taken: The values for NCP and NTU parameters are defined in Chapters 2 and 7, respectively. 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.

Maximum characters in an executor ID is 32.

The identification string for the executor node was invalid.

Action to be taken: Enter an identification string that is in the range of 1 to 32 alphanumeric characters.

The maximum characters in an account is 39.

The number of characters in the account string exceeded the valid range.

Action to be taken: Enter an account number that is in the range of 1 to 39 alphanumeric characters.

The maximum characters in a node name is 6.

The number of characters in the node name was invalid.

Action to be taken: Enter a node name with a maximum of 6 alphanumeric characters. At least one of the characters must be alphabetic.

The maximum characters in a password is 39.

The number of characters in the password was invalid.

Action to be taken: Enter a password with a maximum of 39 alphanumeric characters.

The maximum characters in a user ID is 39.

The number of characters in the user identification string was invalid.

Action to be taken: Enter an identification string with a maximum of 39 alphanumeric characters.

The maximum characters in a receive password is 32.

The number of characters in the receive password was invalid.

Action to be taken: Enter a receive password with a maximum of 32 alphanumeric characters.

The maximum characters in a transmit password is 32.

The number of characters in the transmit password was invalid.

Action to be taken: Enter a transmit password with a maximum of 32 alphanumeric characters.

^User command error

The command line contains an invalid or missing keyword or parameter. The incorrect command line will be displayed with this error message on the next line. The ^ (circumflex) will point to the portion of the command line that was incorrect.

Action to be taken: Request on-line help for the specific command by using the NCP command HELP.

A.4 File Input/Output (I/O) Errors

The messages listed in this section are all related to accessing MS–DOS files.

Could not open file: {*file name*}

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.

Action to be taken: Refer to the *DECnet–DOS Installation Guide* or the *DECnet–Rainbow Installation Guide* for details on the \DECNET directory.

Could not read from file: {*file name*}

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.

Action to be taken: Contact the person responsible for your network for assistance.

Could not seek in file: {*file name*}

An error occurred while you attempted to modify the current seek position in the file.

Action to be taken: Contact the person responsible for your network for assistance.

Could not write to file: {*file name*}

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.

Action to be taken: Contact the person responsible for your network for assistance.

A.5 Network Error Messages

The messages listed in this section are displayed 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

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.

Action to be taken: You should specify a different node name and node address. Retry the operation.

Network Error: Connection timed out

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.

Action to be taken: Set the executor's timeout parameters to longer periods of time. Try the command again.

Network Error: Executor is in wrong state

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.

Action to be taken: Delete the DECPARM.DAT file in your \DECNET directory. Run the SETUP program and @NCP.TXT file as detailed in the *DECnet–DOS Installation Guide* or the *DECnet–Rainbow Installation Guide*.

Network Error: Host is unreachable

A request to set up a node name and node address from a remote node has failed. The remote node cannot be reached.

Action to be taken: Try a different remote node.

Network Error: Line is in wrong state

An attempt has been made to set certain line parameters when the line state is already ON.

Action to be taken: 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

An attempt was made to access the network with the executor state OFF.

Action to be taken: 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

The DECnet driver was not properly installed.

Action to be taken: See the *DECnet–DOS Installation Guide* or the *DECnet–Rainbow Installation Guide* for installation instructions. Repeat the installation procedure.

Network Error: No response from object

A request to set up a node name and node address from a remote node has failed. The logical link connection is broken, or the remote network management listener has failed.

Action to be taken: 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

A request to set up a node name and node address from a remote node has failed. The remote node has no network management listener object.

Action to be taken: Try another remote node.

Network Error: Permission denied

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.

Action to be taken: Use the SET EXECUTOR command to set the executor's node address, then turn the executor state ON with the SET EXECUTOR command.

To 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

The specified socket number was invalid.

Action to be taken: Specify a valid socket number using the SET LINK command. Refer to the *DECnet–DOS Programmer's Reference Manual* for more details.

A.6 NTU Errors

When you test the network using NTU commands, you may see the following error messages in addition to the NCP error messages.

Please run NCP to SET NODE *n* NAME.

There is no node name defined for performing the NTU loop tests.

Action to be taken: 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–DOS network driver is installed and that NCP runs to:

SET EXEC ADDR *node address* NAME *node name*
SET EXEC STATE ON

You have not defined the executor node name and address.

Action to be taken: Make sure DECnet–DOS 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
NFT Error Messages

When an NFT error occurs, you should receive one or more of the error messages listed in this appendix.

If you are accessing a remote system and the remote node reports an error for which no text has been defined, the following message is displayed:

DAP error reported by remote node: *error code/error code*

The error message consists of a pair of DAP error codes in octal. The first error code indicates a specific error type. The second error code identifies the specific reason for that error.

To determine the meanings of the displayed error codes, refer to the Appendix section of the *DECnet–DOS Programmer's Reference Manual*. If you need further assistance, contact your network coordinator.

If a local error occurs for which there is no defined text, the following message is displayed:

Unexpected network error, ERRNO value: *n*

where *n* is in decimal.

If you receive either of these messages, see your network coordinator.

B.1 Types of NFT Error Messages

NFT error messages fall into three general categories:

- **Command line error messages**

  These messages are displayed when you mistype part of a command line (such as the verb or a switch). For example:

  ```
  NFT>COPI  PAGE1.TXT  LONDON::PAGE1.TXT;1
  ```

  Error: Unrecognized command: COPI

  Where the COPY command is mistyped.

- **File Input/Output (I/O) errors**

  These messages are displayed when you are accessing a local or remote file or submitting a remote file. Remote file I/O error messages are displayed on two lines:

  - The first line states the problem.
  - The second line explains why the problem occurred and the line begins with the word ''Because''. These messages are referred to as secondary messages, and are listed at the end of this appendix.

    For example:

    Error Opening File: *node-name filename.typ*
    Because cannot assign address – possibly node name is undefined.

    This message indicates that you typed a remote node name that was not previously defined.

- **Network errors**

  These messages are displayed when you try to run NFT on a system where network support, or some part of it, is missing or not working properly.

Within each category, the messages are listed in this appendix alphabetically.

When a command line specifies multiple operations and an error is encountered during one operation, NFT tries to execute subsequent operations, even after the error has been detected.

B.2 Command Line Error Messages

The error messages listed in this section are all related to the syntax of a command line.

Error: Cannot APPEND to list of files.

You tried to append one or more files to the end of more than one file. You can have only one output file.

Error: Cannot APPEND to wildcard file name.

You included a wildcard in the output file name when you tried to append an input file to an output file.

Error: Cannot COPY from wildcard specification to explicit files.

You tried to copy multiple files using a wildcard while you wanted the output files to have specific names or file types.

Error: Cannot SUBMIT list of files.

You tried to submit more than one remote command file (in a list) to be run on the remote node.

Error: Command string too long.

The command line you typed was longer than 256 characters.

Error: File lists not supported for DELETE command.

You tried to delete a list of files. You can only delete multiple files using wildcards.

Error: File lists not supported for DIRECTORY command.

You tried to display a directory of a list of files. You can only display multiple file names using wildcards.

Error: Illegal ALLOCATION quantity: *text*

You used the /ALLOCATION switch with the COPY command and specified a non-numeric value.

Error: Illegal Maximum Record Size: *value*

You used the /MRS switch with the COPY command and specified either a non-numeric value or a value outside of the range of 0 to 1024.

Error: Illegal VFC amount: *value*

You used the /VFC switch and specified either a non-numeric value or a value that is less than 0 or greater than 255.

Error in allocation amount

You used the /ALLOCATION switch without a value. Please supply one.

Error in allocation quantity

You used the /ALLOCATION switch with the COPY command and the remote node did not accept this value.

Error in maximum record size

You used the /MRS switch with the COPY command and the remote node did not accept the value you specified.

Error in VFC size.

You used the /VFC switch with the COPY command and specified a fixed length header which was not accepted by the remote node.

Error: Quoted local files are illegal.

You enclosed a local file specification in quotation marks (‘‘). Quotation marks are used only for remote file specifications.

Error: Too many file names in list. Limit is 10.

You included more than ten file names in a list (for example, with the APPEND command).

Error: Two paths in a row.

You specified more than one path name without specifying the corresponding file names in between.

Error: Unrecognized command: *command*

You mistyped an NFT command or tried to use a command that NFT could not recognize.

Error: Unrecognized /CC option: *option*

You used the /CC switch and specified an incorrect record attribute. The attributes must be one of the following:

None – No attributes
FTN – FORTRAN carriage control
CR – Implied carriage return/line feed
PRN – Fixed header with carriage control

Error: Unrecognized switch: *switch*

You mistyped the name of a switch or used a switch that NFT could not recognize.

Error: Wildcards are not supported in file name lists.

You tried to list files using wildcards (for example, with the APPEND command).

Warnings

The following messages are warnings. When they are displayed, the current operations will continue.

Print switch not supported with this command.

You used the /PRINT switch with a command that does not support the switch (for example, DELETE).

Warning: Cannot DELETE both remote and local files.
xxx file name(s) will be ignored.

You specified a local and a remote file to be deleted. NFT deletes the file you listed first. Therefore, if you listed the remote file name first, *xxx* is replaced by the word Local. If you listed the local file name first, *xxx* is replaced by the word Remote.

Warning: Cannot perform DIRECTORY for both local and remote files.
xxx file name(s) will be ignored.

You requested a directory listing of both remote and local file names. NFT displays the directory you listed first. Therefore, if you listed the remote directory first, *xxx* is replaced by the word Local. If you listed the local directory first, *xxx* is replaced by the word Remote.

Warning: Cannot print file(s) on local printer.

You included the /PRINT switch in a COPY operation from a remote node to the local node. You cannot print remote files on the local printer. (However, you can print local files on a remote printer.)

Warning: Cannot TYPE from one file to another.

You included two file specifications with the TYPE command.

Warning: Renaming files with wildcard specifications not supported.

You tried to rename a file with the COPY command and used a wildcard in both the input file name and the output file name.

Warning: File(s) will NOT be deleted after copy.

The /DELETE switch was used as an invalid switch with the COPY command. /DELETE is only valid with /PRINT.

Name is too long for local file. Limit is 12 characters.
Enter local file name:

You entered a file name that is too long for DOS to accept. The local file name must be 12 characters or less.

Unable to make local file name from remote name: *node-name::filename.typ*
Enter local file name:

You omitted the local file name with the COPY command and NFT was not able to create a local file name from the specified remote file name.

B.3 File I/O Error Messages

The messages listed in this section are all related to accessing local and remote files or submitting a remote command file. If you receive a remote node error message, refer to the documentation for that operating system or see your network coordinator for instructions on how to correct the problem.

Examples of file I/O errors are improper file specifications or a file read error. Remember that remote file I/O messages are displayed on two lines. The second line is a secondary message that describes why the error occurred. Many of the secondary messages can be displayed with more than one file I/O message.

Cannot delete file: *filename.typ*

NFT cannot delete the local file you specified because it is a directory file or has a Read Only protection.

Error Closing File: *node-name::filename.typ*

NFT cannot close the remote file you specified. The secondary message explains the specific reason.

Error Deleting File: *node-name::filename.typ*

NFT cannot delete the remote file you specified. The secondary message explains the specific reason.

Error In Directory of: *node-name::*

There is a problem with the remote directory you specified. The secondary message explains the specific reason.

Error Opening File: *node-name::filename.typ*

NFT cannot open the remote file you specified. The secondary message explains the specific reason.

Error Reading from: *node-name::filename.typ*

NFT cannot copy from the remote file you specified. The secondary message explains the specific reason.

Error Submitting File: *node-name::filename.typ*

NFT has a problem running the remote command file you specified. The secondary message explains the specific reason.

Error Printing File: *node-name::filename.typ*

NFT has a problem queuing the remote file to a printer. The secondary message explains the specific reason.

Error Writing to: *node-name::filename.typ*

NFT cannot copy to the remote file you specified. The secondary message explains the specific reason.

Error: Cannot SUBMIT local file.

You tried to run a local command file.

Error: Unable to type non-ASCII file: *filename.typ*

You tried to type a file that has a non-ASCII format. NFT displays the name of the file you tried to type.

File not found: *filename.typ*

You tried to access a file that NFT could not find. The file may not exist, or you may have mistyped the file name.

Unable to open file: *filename.typ*

NFT cannot open the local file you specified.

B.4 Network Error Messages

The error messages listed in this section are all related to network software including temporary files created by NFT.

Network errors pertain to network operation failures or rejections. For example, connection requests can be rejected by the network due to insufficient network resources or to an invalid node name.

Allocation quantity too large.

The disk on the remote system is too full to process your request.

Host is down.

You are trying to access a remote host and it is not in operation.

Network is down.

You are trying to perform network activity and the network is not currently available.

No route to host.

This message is an indication of one of the following conditions:

- The network driver is not loaded or you are not using the correct version.
- The line state is OFF.
- The remote host is currently unreachable.

Problem with temporary file for remote file names.

You are copying a file from a remote node, and you included a wildcard in the command. For example, COPY NODE::*.*. NFT first requests a directory list of the remote files. NFT then stores these file names in a temporary local file on the default disk. If a problem occurs with the temporary file due to a lack of disk space, for example, this message is displayed.

Remote system DAP buffer size < 256.

Two programs on different systems have such different buffer sizes that they cannot communicate with one another.

B.5 Secondary Error Messages

Secondary error messages are displayed with remote file I/O messages. Many of the secondary messages can occur with more than one file I/O message. Also, a file I/O message can be displayed with one of several secondary messages depending on the cause of the problem.

Because bad record size

The specified record size is either invalid or illegal for the specified operation.

Because cannot assign address – possibly node name is undefined

The network rejected an attempted connection because the remote node name did not correspond to any node name defined at the local node.

Because cannot open file

An error occurred on a file open operation.

Because cannot position to end of file

NFT cannot append the specified input file to the end of the specified output file.

Because device is write locked

The specified output file cannot be created because the output device is write locked.

Because directory full

You tried to copy a file to a remote node and the remote directory is already full.

Because directory not found

The specified directory does not exist on the node/device that was specified or defaulted in the file specification.

Because disk quota exceeded

You try to perform a remote file operation and there is no room on the remote disk.

Because error in directory name

The specified directory does not conform to the syntax of the target system.

Because error in file name

The specified file name does not conform to the syntax of the target system.

Because error in record attribute

You specified remote file record attributes (such as FORTRAN, implied line feed/carriage return, embedded or VMS print file) that are not valid.

Because error in record format

You specified a remote file record format (such as fixed, variable, VFC, or stream) that is not valid.

Because file locked by other user

Another user currently has the specified file locked or open for writing. You can also receive this message in a full directory listing.

Because file not found

The specified file or files do not exist.

Because file read error

An irrecoverable error has occurred while reading on the device.

Because file write error

An irrecoverable error has occurred while writing on the device.

Because illegal record attributes

The file's record attributes (that is, FORTRAN, implied CR/LF, embedded, or VMS print file) are invalid or unsupported by NFT.

Because incorrect user access information

The network rejected an attempted connection because the specified access control information (userid, password, and account) does not match a valid account on the remote node.

Because invalid record format

The file's record format (that is, fixed, variable, VFC, or stream) is invalid or unsupported by NFT or the remote FAL.

Because invalid wildcard operation

The remote system rejects the specified wildcard as inappropriate for the specified operation.

Because privilege violation

An operation was specified for which you do not have privileges.

C
SETHOST Error Messages

This appendix lists the error messages you may encounter when using the SETHOST utility. The SETHOST error messages fall into two categories. They are listed alphabetically.

Network error messages.

These messages are displayed when you try to run SETHOST on a system where network support, or some part of it, is missing or not working properly.

Execution error messages.

You may encounter these messages when you attempt to perform a SETHOST function, such as redirecting input.

C.1 Network Errors

A network error consists of a pair of messages that are separated by a colon. The message format is:

SETHOST specific message. : DECnet specific message.

The first message can be one of the following SETHOST messages:

SETHOST: Could not send BIND request accept message to host:

SETHOST: Did not receive BIND request message from host:

SETHOST ERROR checking for message from host:

SETHOST ERROR checking for message from host: connection broken:

SETHOST ERROR connecting to host:

SETHOST ERROR receiving message from host:

SETHOST ERROR sending message to host:

The second half of the message string can be one of the following DECnet messages:

Connect failed, Access control rejected.

The remote host rejected the access control information supplied at log-in.

Action to be taken: Check to see if you have a valid user name and password. Try again to log-in to the remote system.

Connect failed, Connection rejected by object.

The remote host rejected your attempt to connect to the system.

Action to be taken: If you need assistance, contact the person responsible for your network.

Connect failed, Insufficient network resources.

There is a problem with the network. You will usually see this error when the maximum number of links has been exceeded.

Action to be taken: Increase the number of maximum links for DECnet–DOS by using the NCP DEFINE EXECUTOR command (refer to Chapter 2 for more information on NCP commands). Reboot the system.

As an alternative, you can reduce the number of links you are currently using and then try to reconnect.

Connect failed, Invalid node name format.

You supplied an invalid node name.

Action to be taken: A valid node name consists of 1 to 6 alphanumeric characters. At least one character must be alphabetic. Enter a valid node name.

Connect failed, Invalid object name format.

An invalid format was used for the object name.

Action to be taken: If you need assistance, contact the person responsible for your network.

Connect failed, Local node shutting down.

Your local system is terminating network activity.

Action to be taken: Try to log-in to the system once it resumes network activity.

Connect failed, No response from object.

There is a problem with the network.

Action to be taken: If you need assistance, contact the person responsible for your network.

Connect failed, Node unreachable.

The remote node is not available for network activity. It is also possible that DECnet is not installed.

Action to be taken: Make sure that DECnet is installed, and that the DECnet circuit is running. (You can check the circuit by using the NCP SHOW CIRCUIT STATUS command.) Try to log-in to the system once it becomes operational again.

Connect failed, Object too busy.

There is a problem with the network.

Action to be taken: If you need assistance, contact the person responsible for your network.

Connect failed, Remote node shutting down.

The remote node is terminating network activity.

Action to be taken: Try to log-in to the system once it resumes network activity.

Connect failed, Unrecognized node name.

The system did not accept the node name that was supplied, or the node name was not defined in your DECnet NCP database.

Action to be taken: A node name consists of 1 to 6 alphanumeric characters. At least one character must be alphabetic. Enter a valid node name. If the name is not defined in the database, use the NCP SET command to define the node you want to use.

Connect failed, Unrecognized object.

There is a problem with the network. The system did not accept your connection request.

Action to be taken: If you need assistance, contact the person responsible for your network.

Unrecognized error code (*error code*).

The error code was not valid.

You should **not** see this message. If this message does appear, contact the person responsible for your network.

There are also some network error messages which contain only one line of text. These messages include:

SETHOST: Remote system does not support correct protocol version.

SETHOST: Remote system does not support SETHOST protocol.

SETHOST ERROR: Could not process the initiate message.

SETHOST ERROR: Protocol violation by host.

SETHOST ERROR: Unknown protocol message from host.

C.2 Execution Error Messages

The following error messages may occur when you attempt to use the SETHOST utility.

Could not restore Session *n* from file SETHOST.*n* in the DECnet directory.

SETHOST could not access the DECnet directory to restore your session.

Action to be taken: Make sure the DECnet directory has been installed correctly. Then try again to restore the session.

Could not save Session *n* to file SETHOST.*n* in the DECnet directory.

SETHOST could not access the DECnet directory to save your session, or the disk may be full.

Action to be taken: Make sure the DECnet directory has been installed correctly. If the disk is full, delete unnecessary files to create more disk space. Then try again to save the session.

Could not write index file SETHOST.*n*.

SETHOST could not access the DECnet directory to create an index file, or the disk may be full.

Action to be taken: Make sure the DECnet directory has been installed correctly. If the disk is full, delete unnecessary files to create more disk space. Then try again to create an index file.

Error writing session log file to disk.

The disk may be full, or SETHOST could not access the DECnet directory to create a log file of your session.

Action to be taken: If the disk is full, delete unnecessary files to create more disk space. Make sure the DECnet directory has been installed correctly. Then try again to create a log file.

Session *n* deleted, connection has broken.

DECnet has disconnected the network link.

Action to be taken: None. SETHOST has updated the index file and deleted the session file that was saved. This is just an information message.

Session *n* does not exist.

There is no saved session with this number.

Action to be taken: Try a different session number, or use the /SHOW switch to see which sessions do exist.

SETHOST could not open input file.

Your attempt in redirecting input was unsuccessful.

Action to be taken: Make sure the input file exists before attempting to redirect input from that file.

SETHOST could not open log file.

Your attempt in redirecting output to a specified file was unsuccessful. The disk may be full.

Action to be taken: Make sure the disk is not full. Delete any unnecessary files and try again.

Unrecognized Command.

The command is not a valid SETHOST command.

Action to be taken: Type SETHOST/HELP to view the valid SETHOST commands.

Unrecognized nodespec.

The nodespec is not valid.

Action to be taken: Type SETHOST/HELP = NODESPEC to view proper nodespec formats.

Unrecognized Parameter.

The parameter you specified is not a SETHOST parameter.

Action to be taken: Type SETHOST/HELP to view the valid SETHOST parameters.

Unrecognized Help option; try typing just /HELP.

The option you entered is not a SETHOST help option.

Action to be taken: Type SETHOST/HELP to view the valid SETHOST help options.

D
NDU Error Messages

This appendix lists the error messages you may encounter when using the NDU utility. The first section deals with Virtual Disk error messages, the second section lists Virtual Printer error messages. The NDU error messages fall into three categories. They are listed alphabetically.

Command line error messages.

These messages are displayed when you mistype part of a command line. The error message includes a circumflex (^) which appears directly under the error, followed by explanatory text. For example:

```
CLOSE DRIVE one
            ^ Disk DRIVE must be a single character.
```

Network error messages.

These messages are displayed when you try to run NDU on a system where network support, or some part of it, is missing or not working properly.

Execution error messages.

You may encounter these messages when you attempt to perform an NDU function, such as CREATE or OPEN.

D.1 Virtual Disk Error Messages

The following error messages may occur when you perform operations on virtual disks.

D.1.1 Network Errors

A network error consists of a pair of messages that are separated by a colon. The message format is:

NDU specific message. : DAP specific message.

The first message can be one of the following NDU messages:

CLOSE during DELETE function failed.

CLOSE function failed.

CREATE function failed.

DELETE function failed.

OPEN function failed.

The second half of the message string can be one of the following DAP messages:

A received DAP message was poorly formed.

During the dialogue with the File Access Server on the remote node (which was necessary to execute your request), a message was received of the proper type but it did not contain the proper contents.

Action to be taken: For an explanation of the DAP message, refer to the Appendix section of the *DECnet–DOS Programmer's Reference Manual*. If you need additional assistance, contact the person responsible for your network.

An unexpected DAP message was received.

During the dialogue with the File Access Server on the remote node (which was necessary to execute your request), a message was received of the wrong type.

Action to be taken: For an explanation of the DAP message, refer to the Appendix section of the *DECnet–DOS Programmer's Reference Manual*. If you need further assistance, contact the person responsible for your network.

Internal error.

An internal error message was detected.

Action to be taken: Contact the person responsible for your network for assistance.

The remote server's buffer is too small.

During the dialogue with the File Access Server on the remote node, which was necessary to execute your request, a message was received which included the maximum possible buffer size for the remaining messages. This size is too small to support the communication.

Action to be taken: Contact the person responsible for your network for assistance.

The remote server could not find that file.

The NDISK file that you specified was not found at that remote node.

Action to be taken: You must specify a file which exists.

The remote server reports DAP error *x:x*.

The remote File Access Listener is relaying a DAP error message to you. The error message consists of a pair of DAP error codes. The first error code indicates a specific error type. The second error code identifies the specific reason for that error.

Action to be taken: To determine the meanings of the displayed error codes, refer to the Appendix section of the *DECnet–DOS Programmer's Reference Manual*. If you need further assistance, contact the person responsible for your network.

The specified file is not a virtual disk file.

The NDISK file that you specified does not have the proper attributes. This can happen if you make an error in the file name.

Action to be taken: Specify a file which has been created with the CREATE command.

D.1.2 Command Line Error Messages

The error messages listed in this section are all related to the syntax of a command line.

ACCESS must be either RO or RW.

You can only specify RO or RW as valid ACCESS options.

Disk DRIVE *name* must be a single character.

A disk drive's name cannot exceed one alphabetic character.

Initial ALLOCATION must be a decimal number.

You can only specify a decimal value greater than 50 for initial disk allocation size.

Unexpected syntax error.

You incorrectly typed a command.

Unrecognized command.

You mistyped an NDU command or tried to use a command that NDU could not recognize.

Unrecognized HELP option; try typing just HELP.

You typed an invalid HELP option. Just type HELP for assistance.

Unrecognized NDISK file name.

A valid file name consists of 1 to 126 alphanumeric characters. The file name may include device and directory fields, but it must be in the proper format for the remote node's system type.

Unrecognized NODE name.

A valid node name consists of 1 to 6 alphanumeric characters. It must contain at least one alphabetic character. The node name can also be a unique numeric address in the format of *area.number*.

Unrecognized NPRINT file name.

A valid file name consists of 1 to 126 alphanumeric characters. The file name may include device and directory fields, but it must be in the proper format for the remote node's system type.

Unrecognized Parameter.

You supplied an invalid parameter with an NDU command. Type HELP for assistance.

Unrecognized SHOW option.

NDU could not recognize the option. The only valid option for the SHOW command is STATUS.

Unrecognized USER name.

NDU could not recognize the USER name. A valid USER's name consists of access control information: *user-id*, *password* and *account*. Each field can include 1 to 39 alphanumeric characters. Enter a valid user name.

D.1.3 Execution Error Messages

The following error messages may occur when you attempt to execute an NDU command or fail to install the virtual disk driver beforehand.

ALLOCATION used for CREATE only, value IGNORED.

The ALLOCATION parameter was not accepted with the command you specified.

Action to be taken: The ALLOCATION parameter can only be supplied with the CREATE command. You cannot use it with other NDU commands.

All the drives are in use, cannot OPEN another drive.

All four disk drives are in use.

Action to be taken: Close one of the open drives before opening another.

All the drives are in use, CREATE function not done.

You cannot create a new virtual disk because all drives are in use.

Action to be taken: Close one of the drives.

Cannot CLOSE a drive, the disk driver is not installed.

You cannot close a virtual disk without having the virtual disk driver installed.

Action to be taken: Refer to the *DECnet–DOS Installation Guide* or the *DECnet–Rainbow Installation Guide* for instructions on installing the Virtual Disk Driver. Install the virtual disk driver and run the NDU utility to close the drive.

Cannot CREATE a drive, the disk driver is not installed.

You cannot create a virtual disk without having the virtual disk driver installed.

Action to be taken: Refer to the *DECnet–DOS Installation Guide* or the *DECnet–Rainbow Installation Guide* for instructions on installing the Virtual Disk Driver. Install the virtual disk driver and run the NDU utility to create the disk drive.

Cannot CREATE a disk file without including NDISK parameter.

You cannot issue a CREATE command without the NDISK parameter.

Action to be taken: Reissue the CREATE command, and specify the name of the file that will contain the virtual disk on the remote node.

Cannot CREATE a disk file without including NODE parameter.

You cannot issue a CREATE command without the NODE parameter.

Action to be taken: Reissue the CREATE command, and specify the name of the remote node with NODE.

Cannot DELETE a disk file without including NDISK parameter.

You cannot issue a DELETE command without the NDISK parameter.

Action to be taken: Reissue the DELETE command, and specify the name of the file that will contain the virtual disk on the remote node.

Cannot DELETE a disk file without including NODE parameter.

You cannot issue a DELETE command without the NODE parameter.

Action to be taken: Reissue the DELETE command, and specify the name of the remote node with NODE.

Cannot DELETE, the disk driver is not installed.

You cannot delete a virtual disk without having the virtual disk driver installed.

Action to be taken: Refer to the *DECnet–DOS Installation Guide* or the *DECnet–Rainbow Installation Guide* for instructions on installing the Virtual Disk Driver. Install the virtual disk driver and run the NDU utility to delete the disk drive.

Cannot open a drive, the disk driver is not installed.

You have not installed NDU properly.

Action to be taken: Refer to the *DECnet–DOS Installation Guide* or the *DECnet–Rainbow Installation Guide* for instructions on installing the Virtual Disk Driver. Repeat the procedure and run the NDU utility.

Cannot open a drive without including NDISK parameter.

You cannot issue an OPEN command without the NDISK parameter.

Action to be taken: Reissue the OPEN command, and specify the name of the file that will contain the virtual disk on the remote node.

Cannot open a drive without including NODE parameter.

You cannot issue an OPEN command without the NODE parameter.

Action to be taken: Reissue the OPEN command, and specify the name of the remote node with NODE.

Drive *name* is already OPEN; CREATE function not done.

The drive is already open. You cannot format the new data file.

Action to be taken: Use the SHOW STATUS command to list the open drives.

Drive *name* is already OPEN, cannot OPEN it again.

The drive is already open.

Action to be taken: You cannot open a drive that is already open.

Drive *name* is not a network disk.

The drive name that you specified is invalid.

Action to be taken: The drive name consists of one alphabetic character. Use the SHOW STATUS command to list the valid drive names.

Drive *name* is not a network disk, cannot CLOSE it.

You cannot close a drive which is not a virtual disk drive.

Action to be taken: Use the SHOW STATUS command to list the open drives.

Drive *name* is not a network disk, cannot OPEN it.

You can only use the drive names listed by the SHOW STATUS command.

Action to be taken: Use a proper drive name.

Drive *name* is not a network disk, CREATE function not done.

You can only use the drive names listed by the SHOW STATUS command.

Action to be taken: Use a proper drive name.

Drive *name* is not OPEN, cannot CLOSE it.

You cannot close an unopened drive.

Action to be taken: Use the SHOW STATUS command to list the status of the drives.

Drive *name* is not OPEN, cannot DELETE its disk.

You cannot delete a disk without having the drive opened.

Action to be taken: Use the SHOW STATUS command to list the status of the drives.

Drive *name* is too long. It can only be a single character.

You supplied an invalid name for the disk drive.

Action to be taken: A valid drive name consists of one alphabetic character. Use the SHOW STATUS command to list the valid drive names.

Either DRIVE or NDISK must be specified, the CLOSE function not done.

You cannot issue the CLOSE function without specifying DRIVE or NDISK as a parameter.

Action to be taken: To perform the CLOSE function, you must specify either the name of the virtual disk drive or the file that will be the virtual disk on the remote node.

Length of NDISK parameter is too long.
File name must be less than 127 characters.

An invalid length was supplied for the file name.

Action to be taken: The NDISK parameter specifies the name of the file which will be the virtual disk on the remote node. The valid range for the file name is 1 to 126 alphanumeric characters. Enter a valid file name.

Length of NODE parameter is too long.
Node name must be less than 6 characters.

An invalid node name was supplied.

Action to be taken: A valid node name consists of 1 to 6 alphanumeric characters. It must contain at least 1 alphabetic character. Enter a valid node name.

Length of USER parameter is too long.
The string must be less than 120 characters.

You supplied an invalid user name.

Action to be taken: NDU could not recognize the USER name. A valid user's name consists of access control information: *user-id*, *password* and *account*. Each field can include 1 to 39 alphanumeric characters. Enter a valid user name.

NDISK not found, the CLOSE function was not done.

Either DRIVE or NDISK must be specified, the CLOSE function was not done.

Action to be taken: To determine which drive is to be closed, you must specify a drive name, an NDISK file specification, or both.

The Disk Driver is not installed.

You cannot create or use a virtual disk without first installing the Virtual Disk Driver.

Action to be taken: Refer to the *DECnet–DOS Installation Guide* or the *DECnet–Rainbow Installation Guide* for instructions on installing the Virtual Disk Driver. Repeat the installation procedure and then try to run NDU.

D.2 Virtual Printer Error Messages

The following error messages may occur when you use the virtual printer feature.

D.2.1 Network Errors

A network error consists of a pair of messages that are separated by a colon. The message format is:

NDU specific message. : DAP specific message.

The first message can only be:

OPEN/CREATE function failed.

The second half of the message string can be one of the following DAP messages:

A received DAP message was poorly formed.

During the dialogue with the File Access Server on the remote node (which was necessary to execute your request), a message was received of the proper type but it did not contain the proper contents.

Action to be taken: For an explanation of the DAP message, refer to the Appendix section of the *DECnet–DOS Programmer's Reference Manual*. If you need additional assistance, contact the person responsible for your network.

An unexpected DAP message was received.

During the dialogue with the File Access Server on the remote node (which was necessary to execute your request), a message was received of the wrong type.

Action to be taken: For an explanation of the DAP message, refer to the Appendix section of the *DECnet–DOS Programmer's Reference Manual*. If you need further assistance, contact the person responsible for your network.

Internal error.

An internal error message was detected.

Action to be taken: Contact the person responsible for your network for assistance.

The remote server's buffer is too small.

During the dialogue with the File Access Server on the remote node (which was necessary to execute your request), a message was received which included the maximum possible buffer size for the remaining messages. This size is too small to support the communication.

Action to be taken: Contact the person responsible for your network for assistance.

The remote server could not find that file.

The NDISK file that you specified was not found at that remote node.

Action to be taken: You must specify a file which exists.

The remote server reports DAP error *x:x*.

The remote File Access Listener is relaying a DAP error message to you. The error message consists of a pair of DAP error codes. The first error code indicates a specific error type. The second error code identifies the specific reason for that error.

Action to be taken: To determine the meanings of the displayed error codes, refer to the Appendix section of the *DECnet–DOS Programmer's Reference Manual*. If you need further assistance, contact the person responsible for your network.

D.2.2 Command Line Error Messages

The error messages listed in this section are all related to the syntax of a command line.

Unexpected syntax error.

You incorrectly typed a command.

Unrecognized command.

You mistyped an NDU command or tried to use a command that NDU could not recognize.

Unrecognized HELP option; try typing just HELP.

You typed an invalid HELP option. Just type HELP for assistance.

Unrecognized NODE name.

A valid node name consists of 1 to 6 alphanumeric characters. It must contain at least one alphabetic character. The node name can also be a unique numeric address in the format of *area.number*.

Unrecognized NPRINT file name.

A valid file name consists of 1 to 126 alphanumeric characters. The file name may include device and directory fields, but it must be in the proper format for the remote node's system type.

Unrecognized Parameter.

You supplied an invalid parameter with an NDU command. Type HELP for assistance.

Unrecognized SHOW option.

NDU could not recognize the option. The only valid option for the SHOW command is STATUS.

Unrecognized USER name.

NDU could not recognize the USER name. A valid USER's name consists of access control information: *user-id*, *password* and *account*. Each field can include 1 to 39 alphanumeric characters. Enter a valid user name.

D.2.3 Execution Error Messages

The following error messages may occur when you attempt to execute an NDU command or fail to install the virtual printer driver beforehand.

Cannot CLOSE the printer, the Printer Driver is not installed.

You cannot close the printer without having the virtual printer driver installed.

Action to be taken: Refer to the *DECnet–DOS Installation Guide* or the *DECnet–Rainbow Installation Guide* for instructions on installing the Virtual Printer Driver. Install the virtual printer driver and run the NDU utility to close the printer.

Cannot OPEN/CREATE the printer, the Printer Driver is not installed.

You cannot open/create the virtual printer without having the virtual printer driver installed.

Action to be taken: Refer to the *DECnet–DOS Installation Guide* or the *DECnet–Rainbow Installation Guide* for instructions on installing the Virtual Printer Driver. Install the virtual printer driver and run the NDU utility to open/create the printer.

Cannot OPEN/CREATE the printer without including NODE parameter.

You cannot issue the OPEN or CREATE command without the NODE parameter.

Action to be taken: Reissue the OPEN or CREATE command, and specify the name of the remote node with NODE.

Length of NODE parameter is too long.
Node name must be less than 6 characters.

You supplied an invalid node name.

Action to be taken: A valid node name consists of 1 to 6 alphanumeric characters. It must contain at least 1 alphabetic character. Enter a valid node name.

Length of NPRINT parameter is too long.
File name must be less than 127 characters.

You supplied an invalid file name for the NPRINT parameter.

Action to be taken: A valid file name consists of 1 to 126 alphanumeric characters. The file name may include device and directory fields, but it must be in the proper format for the remote node's system type. Enter a valid file name.

Length of USER parameter is too long.
The string must be less than 120 characters.

You supplied an invalid user name.

Action to be taken: NDU could not recognize the USER name. A valid user's name consists of access control information: *user-id*, *password* and *account*. Each field can include 1 to 39 alphanumeric characters. Enter a valid user name.

Printer is not OPEN, cannot CLOSE it.

You cannot close an unopened printer.

Action to be taken: Use the SHOW STATUS command to list the status of the printer.

The Printer Driver is not installed.

You cannot create or use a virtual printer without first installing the Virtual Printer Driver.

Action to be taken: Refer to the *DECnet–DOS Installation Guide* or the *DECnet–Rainbow Installation Guide* for instructions on installing the Virtual Printer Driver. Repeat the installation procedure and then try to run NDU.

Glossary

access control information

Optional security information that you may need to access a remote node.

account

A character string consisting of 1 to 39 alphanumeric characters.

adjacent node

A node that you are physically connected to by a single line.

alphanumeric

A string of characters that contains letters (alphabetic characters), numbers (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 in order to meet user requirements. Applications available with DECnet–DOS include the Network File Transfer utility, the Network Device Utility, the Network Test Utility, and the SETHOST utility with network virtual terminal services.

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 can be stored.

circuit

The communications path that operates over a physical line connecting two nodes.

command switch

A word or character string that modifies the way in which a command operates. Typing the command **DIRECTORY LONDON::** causes the system to display a full directory of all files located on node LONDON. When you modify the command using the **/BRIEF** switch, (as in **DIRECTORY/BRIEF LONDON::**), the system only displays the file names.

communication path

The route through hardware components and a cable that a message takes when sent from one node to another.

counter

A counter displays statistics about the flow of network messages for your node. The counters also record error conditions and accumulate their totals for you.

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

A set of software and hardware components designed for use with the IBM PC, the IBM PC/XT and the IBM Personal Computer AT. DECnet–DOS allows you to combine the capabilities of the IBM personal computers with DECnet applications to access the network as an end node.

DECnet–Rainbow

A set of software and hardware components designed for use with the Rainbow personal computers. DECnet–Rainbow allows you to combine the capabilities of the Rainbow personal computers with DECnet applications to create end-node only networks.

device

A specific name for a disk or diskette that is currently storing data files.

directory

A specific name assigned to a collection of files stored on a disk or a diskette. A directory can also provide a list of the file names that it contains.

emulate

To imitate the performance of a terminal on another system. When your personal computer emulates a terminal on a host system, it performs many of the same functions as the host terminal.

end node

A node that can only send and receive information for its own use. An end node cannot receive and then forward information intended for another node. Your personal computer is an end node.

executor node

A node that runs NCP commands. For a personal computer, this is the local node.

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 that has a format other than a format recognizable to DECnet–DOS. A foreign file specification must be 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.

line

The physical line connecting the local node with the adjacent node.

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.

loopback connector

A hardware device that temporarily connects the output portion of a circuit back to the input portion of a circuit for the purpose of testing communication lines. The loopback connector verifies the operation of devices on the same communications link.

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 **loopback connector**).

modem

A hardware device capable of changing communications signals from digital to analog and back to digital. The term *modem* is a contraction of the words *mod*ulation and *dem*odulation.

modulation

The process of varying an original computer communication (digital) signal into a telephone communication (analog) signal.

network

A group of computers or systems that are connected together and can communicate with each other to share resources and information.

network coordinator

A person who assigns and updates node names and node addresses. The network coordinator also provides administrative assistance to network users.

network link

A temporary connection that establishes communication between programs running on different nodes. The SETHOST utility with network virtual terminal services is a program that establishes a network link with another node, enabling you to log into a remote host system.

node

A computer with the necessary hardware and software to participate in a network with other computers.

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 alphanumeric character string (containing at least one letter) that identifies a node to other users in the network.

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.

protocols

Rules or formats that operating systems must follow to conduct effective communications with other computers in a network.

queue

A waiting line for completing a service, such as transferring files.

reachable node

A node to which the local node has a usable communications path.

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 available for communication on the same network.

router

A DECnet node that can receive messages and information from one node and forward them to another node.

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.

segment

A set of data that can be stored anywhere in memory, and can be retrieved at any time.

TO *file-id*

A qualifier that specifies the name of the output file which stores information resulting from the execution of an NCP command.

topology

The physical arrangement of nodes and connecting hardware that makes up the network.

user name

A character string consisting of 1 to 39 alphabetic characters identifying a user at a remote node.

Index

A

Abbreviating NCP commands, 2–5
Access control information, 2–8 to 2–11, 2–21, 2–42, 3–2
  changing and overriding, 2–9
  changing defaults, 3–4
  defining with NFT, 3–3
  displaying, 2–10, 3–5
  displaying access defaults, 3–34
  setting defaults, 3–33
  specifying, 2–8
  using defaults, 3–4
Accessing a remote printer, 5–2
Accessing remote files, 3–1
Account information, 2–8, 3–3
Active nodes, 2–12
Adjacent node, 1–3
Appending files, 3–17, 3–22
ASCII file, 3–8, 3–10

B

Batch files
  used with NCP, 2–2
  used with NDU, 5–3, 5–14
  used with NFT, 3–2, 3–20
  used with NTU, 7–6

C

Circuit
  defining parameters, 2–33, 7–35
  displaying information, 2–44
  zeroing counters, 2–51
Circuit state, 2–12
Circuit substates, 2–13
Command files, 3–20
  submitting, 3–35
Copying a local file to a remote node, 3–18
Copying a remote file to a local node, 3–18
Copying files, 3–6, 3–9, 3–10, 3–18, 3–24
Counters, 2–51, 2–52, 2–54

D

Data
  ASCII, 3–8, 3–10
  Image, 3–8, 3–9
DECnet
  terms and concepts, 1–1
DECnet–DOS
  capabilities, 1–3
  definitions
    see Glossary
  utilities
    FAL, 3–37
    Job Spawner, 3–42

DECnet–DOS (Cont.)
utilities (Cont.)
MAIL, 6–1
NCP, 2–1
NDU, 5–1
NFT, 3–1
NTU, 7–1
SETHOST, 4–1
summary of, 1–4
Deleting files, 3–18, 3–27
Directories
displaying files with wildcards, 3–19
displaying information, 3–19, 3–28
Disconnecting from a remote node, 4–29

E

Editing text during a remote session, 4–28
End node, 1–3
Event log
continuous, 2–28, 7–31
reading contents of, 2–31, 7–33
sample, 2–29, 2–31, 7–32
Executor node, 1–3
clearing parameters, 2–20
defining parameters, 2–24, 2–35
displaying characteristics, 2–45
displaying information, 2–11
zeroing counters, 2–52

F

FAL, 3–37
File Access Listener
see FAL
File operations
logging, 3–20
File specifications, 3–6
for naming files, 3–6
foreign, 3–7
local, 3–6
remote, 3–6
using wildcards, 3–7, 3–8
Files
appending, 3–17, 3–22
copying, 3–18, 3–24
Files (Cont.)
deleting, 3–18, 3–27
displaying contents of, 3–19, 3–36
listing, 3–17
printing, 3–19, 3–32
types of, 3–8

G

Getting help for
MAIL, 6–7, 6–8
NCP, 2–5, 2–27
NDU, 5–6, 5–11, 5–16, 5–19
NFT, 3–16, 3–31
NTU, 7–7, 7–22
SETHOST, 4–7, 4–15

H

Host node, 4–1
disconnecting from, 4–29

I

Image file, 3–8, 3–9
Installing virtual device programs, 5–2

J

Job Spawner, 3–42
creating a DECSPAWN.DAT file, 3–42
starting, 3–42

K

Keys
Line feed, 3–11

L

LAT
displaying information, 2–47
zeroing counters, 2–53
Line
defining parameters, 2–39
displaying information, 2–12, 2–13, 2–48
setting line speed, 2–39
zeroing counters, 2–54
Line states, 2–12
Line substates, 2–12

Listing files, 3–17
Local file
  copying to a remote node, 3–6, 3–10, 3–18
Local files
  naming, 3–6
  providing access, 3–1, 3–37
Local node, 1–3
Logging file operations, 3–20
Logical links
  and sockets, 2–38, 2–41, 7–40
  displaying information, 2–11, 2–49
  setting maximum number, 2–24

M

MAIL, 6–1
  getting help for, 6–7, 6–8
  initialization file, 6–1
  initialization file parameters, 6–1
    CARBON__COPY, 6–3
    DEFAULT__NODE, 6–2
    EDITOR, 6–4
    PERSONAL__MESSAGE, 6–4
    REPLY__ADDRESS, 6–2
  qualifiers
    using, 6–7
    /EDIT, 6–7
    /SUBJECT, 6–7
  starting, 6–4
  using distribution lists, 6–5
MAIL command summary, 6–8
MAIL commands
  entering (from the command line), 6–4
  entering (from the mail prompt), 6–5
  EXIT, 6–8
  HELP, 6–8
  MAIL, 6–9
  SEND, 6–9

N

NCP, 2–1
  command format, 2–4
  exiting from, 2–3, 2–26
  getting help for, 2–5, 2–27
NCP (Cont.)
  starting, 2–2
  typing commands, 2–5
NCP command parameters, 2–17
NCP command summary, 2–15
NCP commands
  CLEAR ACCESS, 2–18
  CLEAR EXECUTOR, 2–20
  CLEAR KNOWN ACCESS, 2–19
  CLEAR NODE, 2–21
  COPY KNOWN NODES, 2–22
  DEFINE CIRCUIT, 2–23
  DEFINE EXECUTOR, 2–7, 2–24
  DEFINE LINE, 2–25
  EXIT, 2–3, 2–26
  HELP, 2–5, 2–27
  MONITOR LOGGING, 2–28
  QUIT, 2–3
  READ LOGGING, 2–31
  SET ACCESS, 2–32
  SET CIRCUIT, 2–33
  SET ECHO, 2–34
  SET EXECUTOR, 2–35
  SET LINE, 2–39
  SET LINK, 2–41
  SET LINKS, 2–38
  SET NODE, 2–9, 2–42
  SHOW ACTIVE NODES, 2–12
  SHOW CIRCUIT, 2–44
  SHOW EXECUTOR, 2–11, 2–45
  SHOW KNOWN ACCESS, 2–46
  SHOW KNOWN NODES, 2–10
  SHOW LAT, 2–47
  SHOW LINE, 2–12, 2–13, 2–48
  SHOW LINK, 2–49
  SHOW NODE, 2–11, 2–50
  ZERO CIRCUIT, 2–51
  ZERO EXECUTOR, 2–52
  ZERO LAT, 2–53
  ZERO LINE, 2–54
NCP error messages
  see Appendix A
NDU, 5–1
  command format, 5–4, 5–14
  ending connection with remote disk, 5–7
  exiting from, 5–3, 5–14
  getting help for, 5–6, 5–11, 5–16, 5–19

NDU (Cont.)
prompts, 5–3, 5–14
sample disk status display, 5–13, 5–20
sample printer status display, 5–13, 5–20
starting (remote printers), 5–14
starting (virtual disks), 5–3
NDU command summary
remote printers, 5–16
virtual disks, 5–6
NDU commands
CLOSE, 5–7, 5–17
CREATE, 5–8, 5–18
DELETE, 5–10
EXIT, 5–3, 5–14
HELP, 5–6, 5–11, 5–16, 5–19
OPEN, 5–12, 5–18
SHOW, 5–13, 5–20
NDU error messages
see Appendix D
Network
changing information, 2–13
changing parameters, 2–13
defined, 1–1
displaying information, 2–11
Network Control Program
see NCP
Network Device Utility
see NDU
Network File Transfer utility
see NFT
Network Mail Utility
see MAIL
Network Test Utility
see NTU
Network Virtual Terminal Utility
see SETHOST
NFT, 3–1
abbreviating commands, 3–10
abbreviating switches, 3–10
access control information, 3–3
command format, 3–10
exiting from, 3–2, 3–30
file operations, 3–16
getting help for, 3–16, 3–31
starting, 3–2
using long commands, 3–11
using switches, 3–11
NFT command summary, 3–21
NFT commands
APPEND, 3–17, 3–22
COPY, 3–6, 3–18, 3–24
DELETE, 3–18, 3–27
DIRECTORY, 3–19, 3–28
EXIT, 3–30
HELP, 3–11, 3–16, 3–31
PRINT, 3–32
SET, 3–4, 3–5, 3–33
SHOW, 3–5, 3–34
SUBMIT, 3–35
TYPE, 3–19, 3–36
typing, 3–10
NFT error messages
see Appendix B
NFT switches
for APPEND, 3–22
for COPY, 3–24
for DELETE, 3–27
for DIRECTORY, 3–28
for HELP, 3–31
for PRINT, 3–32
for SUBMIT, 3–35
summary of, 3–12 to 3–15
/BRIEF, 3–19
/IMAGE, 3–10
/NOCONVERT, 3–9
/PRINT, 3–19
Node
clearing access control information, 2–21
clearing node name, 2–21
clearing parameters, 2–21
defining access control information, 2–42
defining node name, 2–42
Node address, 2–7, 2–8
Node name, 2–6, 2–7, 2–8
Nodes, 1–1
adjacent, 1–3
defining address, 2–7
defining name, 2–7
displaying information, 2–50
end, 1–3
executor, 1–3
host, 4–1
local, 1–3
remote, 1–3, 2–8

Nodes (Cont.)
routing, 1–3
setting address, 2–6
setting name, 2–6
Nonstream file system, 3–9
NTU, 7–1
exiting from, 7–7
getting help for, 7–7, 7–22
overview, 7–2
prestartup, 7–5
starting, 7–6
using commands, 7–7
NTU command summary, 7–9 to 7–10
NTU commands
CLEAR ACCESS, 7–13
CLEAR ACCOUNT, 7–14
CLEAR COUNT, 7–15
CLEAR DEFAULTS, 7–16
CLEAR LENGTH, 7–17
CLEAR PASSWORD, 7–18
CLEAR USER, 7–19
CLEAR WITH, 7–20
EXIT, 7–21
HELP, 7–22
LOOP CIRCUIT, 7–23
LOOP EXECUTOR, 7–26
LOOP LINE CONTROLLER, 7–27
LOOP NODE, 7–28
MIRROR, 7–30
MONITOR LOGGING, 7–31
READ LOGGING, 7–33
SET ACCOUNT, 7–34
SET CIRCUIT, 7–35
SET COUNT, 7–36
SET ECHO, 7–37
SET LENGTH, 7–38
SET LINE STATE, 7–39
SET LINK, 7–40
SET PASSWORD, 7–41
SET USER, 7–42
SET WITH, 7–43
SHOW ACTIVE LINKS, 7–44
SHOW ACTIVE NODES, 7–45
SHOW ADJACENT NODE, 7–46
SHOW CIRCUIT, 7–47
SHOW DEFAULTS, 7–48
NTU commands (Cont.)
SHOW EXECUTOR, 7–49
SHOW KNOWN LINKS, 7–50
SHOW KNOWN NODES, 7–51
SHOW LINE, 7–52
SHOW NODE, 7–53
ZERO CIRCUIT, 7–54
ZERO EXECUTOR, 7–55
ZERO LINE, 7–56
NTU error messages
see Appendix A

P

Password, 2–8, 3–2
Printing a file, 3–19
Printing files, 3–32
Prompts
NCP, 2–2
NDU, 5–3
NFT, 3–2, 3–11
NTU, 7–5

R

Reading input from a file, 2–34, 7–37
Redirected input files
used with NCP, 2–2, 2–34
used with NTU, 7–6, 7–37
Remote file
copying to a local node, 3–9, 3–18
printing, 3–19
Remote files
accessing, 3–1
Remote node, 1–3, 2–8
displaying information, 2–11
displaying names, 2–10
naming, 2–8
Remote printers, 5–2
closing down connection, 5–17
creating a print queue, 5–18
opening a print queue, 5–18
Routing node, 1–3

S

Server functions
using the job spawner
with DTR, 3–42, 3–43
with FAL, 3–42

SET-Up
exiting from, 4–28
recalling parameters, 4–28
saving parameters, 4–28
selecting parameter menu, 4–27
setting VT102 class terminal features, 4–26
SETHOST, 4–1
command parameters, 4–2
latservicename, 4–8
nodespec, 4–8
command switches, 4–2
/ABORT, 4–3
/CTERM, 4–4
/EDIT, 4–5
/EXIT, 4–6
/HELP, 4–7
/LOG, 4–9
/SAVEDEFAULT, 4–10
/SESSION, 4–11
/SHOW, 4–12
/TAKE, 4–13
exiting from, 4–29
getting help for, 4–7, 4–15
interrupting a session, 4–14
menu, 4–14
<A>BORT SESSION, 4–16
<E>DIT, 4–15
<H>ELP, 4–15
<L>OG, 4–15
<N>EW SESSION, 4–16
<R>ECONNECT, 4–15
<RETURN>, 4–15
<S>ESSION, 4–16
SETHOST error messages
see Appendix C
Stream file system, 3–9
Switches
NFT, 3–11
SETHOST, 4–2

U

Username, 2–8, 3–2

V

Virtual disk
deleting space, 5–10
Virtual disk volumes, 5–2
Virtual disks
creating, 5–8
establishing connection with virtual disk, 5–12

W

Wildcards
combining, 3–8
in file specifications, 3–7
question marks, percent signs, and asterisks, 3–8

DECnet–DOS
User's Guide
AA–EB45B–TV

READER'S COMMENTS

What do you think of this manual? Your comments and suggestions will help us to improve the quality and usefulness of our publications.

Please rate this manual:

	Poor				Excellent
Accuracy	1	2	3	4	5
Readability	1	2	3	4	5
Examples	1	2	3	4	5
Organization	1	2	3	4	5
Completeness	1	2	3	4	5

Did you find errors in this manual? If so, please specify the error(s) and page number(s).

_____

_____

_____

_____

General comments:

_____

_____

_____

_____

_____

Suggestions for improvement:

_____

_____

_____

_____

Name _____ Date _____

Title _____ Department _____

Company _____ Street _____

City _____ State/Country _____ Zip Code _____

DO NOT CUT FOLD HERE AND TAPE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY LABEL

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

digital

SOFTWARE DOCUMENTATION
550 KING STREET
LITTLETON, MA 01460–1289

DO NOT CUT FOLD HERE

CUT ALONG DOTTED LINE