

DECnet/E Utilities Guide

Order No. AA-H504C-TC

January 1987

The DECnet/E utilities allow a terminal user to: talk to a user locally or at another node, log in at and use another DECnet/E node, manipulate files between nodes, copy the contents of storage devices between DECnet/E nodes, and shut down the network.

OPERATING SYSTEM AND VERSION:	RS/TS/E	V9.3
SOFTWARE VERSION:	DECnet/E	V4.0

digital equipment corporation, maynard, massachusetts

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

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

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

Copyright © 1982, 1987 by Digital Equipment Corporation.
All rights reserved.

The postage-paid READER'S COMMENTS form on the last page of this document requests your critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

digital ™	DECUS	Rainbow
DATATRIEVE	DECwriter	ReGIS
DEC	DIBOL	RSTS
DECmail	FMS-11	RSX
DECmate	LA	RT
DECnet	MASSBUS	UNIBUS
DECsystem-10	PDP	VAX
DECSYSTEM-20	P/OS	VMS
DECtape	Professional	VT
	Q-Bus	Work Processor

CONTENTS

PREFACE

SUMMARY OF TECHNICAL CHANGES

CHAPTER 1 Introduction

CHAPTER 2 TLK: Terminal Communication Utility

One-Line Mode	2-2
Initiator's Input	2-2
Receiver's Output	2-4
Dialog Mode	2-5
Initiator's Input	2-5
Receiver's Interaction	2-7
TLK Error Messages	2-10

CHAPTER 3 NET: Network Command Terminal Utility

Running NET	3-2
NET Commands	3-4
CONTINUE Command	3-6
CTRL/P Command	3-7
DETACH Command	3-8
EXIT Command	3-9
HELP Command	3-10
NORMAL Command	3-11
ODT Command	3-12
NET Error Messages	3-13

CHAPTER 4 NFT: Network File Transfer Utility

The Relationships Between NFT, FAL, and RMS	4-2
FAL Access to ISAM Files	4-4
Running NFT	4-5
General Format of NFT Commands	4-6
Node Names	4-7
File Specifications	4-9
Wildcards	4-11
Quoted Strings	4-12
NFT Commands	4-13
AP[PEND] Command	4-14
CO[PY] Command	4-16
DE[LETE] Command	4-19
DI[RECTORY] Command	4-21
EXI[T] Command	4-23
RE[LP] Command	4-24

NOD[ESPECIFICATION] Command	4-25
PR[INT] Command	4-27
SU[BMIT] Command	4-29
TY[PE] Command	4-31
VE[RSION] Command	4-33
NFT Switches	4-34
/AP[PEND] Switch	4-36
/ASCII Switch	4-36
/BL[OCK] Switch	4-36
/BR[IEF] Switch	4-37
/CO[NTIGUOUS] or /CTG Switch	4-38
/DE[LETE] Switch	4-38
/FU[LL] Switch	4-38
/ID[ENTIFY] Switch	4-38
/IN[AGE] Switch	4-39
/IN[QUIRY] Switch	4-39
/LI[ST] Switch	4-40
/LO[G] Switch	4-40
/MO[RE] or - Switch	4-40
/NA[TIVE] Switch	4-41
/NOA[TTIBUTES] Switch	4-42
/NOCO[NTIGUOUS] Switch	4-42
/NODE[LETE] Switch	4-42
/NOHE[ADING] Switch	4-42
/NOSU[PERSEDE] Switch	4-43
/POS Switch	4-43
/RWC Switch	4-43
/RWC Switch	4-43
/SU[PERSEDE] Switch	4-44
/TO[TAL] Switch	4-44
/VA[RIBLE] Switch	4-44
NFT Error Messages	4-44

CHAPTER 5

NETCPY: Network Copy Between Devices

Running NETCPY	5-2
General Format of NETCPY Command	5-2
NETCPY Switches	5-4
/BL[OCK]:n Switch	5-5
/DE[NSITY]:n Switch	5-5
/PC Switch	5-6
/HE[LP] Switch	5-6
/NC Switch	5-7
/PA[RITY]:n Switch	5-7
/VE[RIFY] Switch	5-7
NETCPY Error Messages	5-9

CHAPTER 6

NETOFF: Network Shutdown Utility

APPENDIX A

NSP Reason Codes for Connect Reject and Link Abort

APPENDIX B

Error Messages Generated by RMS and DAP

NFT Error Messages from RMS-11	B-1
NFT Error Messages from DAP	B-11
Macro (aa) Field Values	B-11
Micro (nnnn) Field Values	B-13

INDEX

FIGURES

1-1	DECnet/E Features for the Terminal User	1-2
1-2	Sample Network Composed of Seven Nodes.	1-4
2-1	The TLK Utility: TLK and LSN Provide Communication Between Terminals	2-2
3-1	NET Provides Access to Another DECnet/E Node	3-2
4-1	NFT Transfers Files to, from, and between Remote Nodes	4-2
4-2	At DECnet/E Nodes, NFT, PAL, and RMS Work Together to Allow File Operations	4-4
4-3	Sample Network Showing Transfers Between Nodes with DECnet/E NFT	4-8
4-4	Example of the NFT APPEND Command	4-15
4-5	Example of the NFT COPY Command	4-18
4-6	Example of the NFT DELETE Command	4-20
4-7	Example of the NFT DIRECTORY Command	4-22
4-8	Example of the NFT PRINT Command	4-28
4-9	Example of the NFT SUBMIT Command	4-30
4-10	Example of the NFT TYPE Command	4-32
5-1	NETCPY Copies an Entire Device from One DECnet/E Node to Another DECnet/E Node	5-1

TABLES

2-1	TLK Error Messages	2-10
3-1	NET Commands	3-5
3-2	NET Error Messages	3-13
4-1	NFT Commands	4-13
4-2	NFT Switches	4-34
4-3	NFT Error Messages	4-45
5-1	NETCPY Switches	5-4
5-2	NETCPY Error Messages	5-9
A-1	NSP Reason Codes	A-1
B-1	RMS Error Codes	B-2
B-2	Macro (aa) Field Values	B-12
B-3	Micro (nnnn) Field Values	B-13
B-4	Micro (nnnn) Format	B-20
B-5	Micro (nnnn) Values for Macro (aa) Value Octal 12	B-28

Preface

Objectives

This guide describes the DECnet/E utilities that allow you to:

- o Talk to a user locally or at another node
- o Log in at and use another DECnet/E node
- o Manipulate files between nodes
- o Copy the contents of entire disks and tapes between devices at different DECnet/E nodes
- o Shut down your network in an orderly fashion

Audience

This guide is for any RSTS/E terminal user who wants to use the DECnet/E utilities to perform any of the DECnet/E functions previously listed. You should understand the operation of the local RSTS/E system and the remote system you want to access.

Document Structure

This guide has six chapters and two appendixes:

- | | |
|------------|---|
| Chapter 1 | Introduces the DECnet/E utilities and summarizes their use. |
| Chapter 2 | Describes the TLK Terminal Communication utility. |
| Chapter 3 | Describes the NET Network Command Terminal utility. |
| Chapter 4 | Describes the NET Network File Transfer utility. |
| Chapter 5 | Describes the NETCPY Network Copy Device utility. |
| Chapter 6 | Describes the NETOFF Network Shutdown utility. |
| Appendix A | Lists the NSP Reason Codes. |
| Appendix B | Lists the error messages that RMS and DAP generate. |

Related Documents

This guide references the following documents:

- o The DECnet/E Network Installation and Update Guide and the DECnet/E System Manager's Guide tell how to generate and start a DECnet/E system, control and monitor a running DECnet/E system, and give other information useful to a system manager.
- o The RSTS/E Quick Reference Guide and the RSTS/E Guide to Writing Command Procedures tell how to use DCL commands.
- o The RSTS/E System User's Guide describes the functions and use of RSTS/E for the general user.
- o The RSTS/E Utilities Reference Manual tells how to use RSTS/E utilities.
- o The Introduction to DECnet Phase IV gives an overview of the concepts and capabilities of DECnet networks implemented by Phase IV.
- o The DECnet DIGITAL Network Architecture (Phase IV) General Description describes the design of DIGITAL Network Architecture (DNA) that serves as a model for Phase IV DECnet implementations.

Conventions

This guide uses the following conventions:

<RET> Represents the RETURN key on your terminal. You use the RETURN key to complete lines and commands.

<LF> Represents the LINE FEED key on your terminal.

CTRL/x Indicates a control key combination, such as CTRL/Z or CTRL/P. To enter a control key combination, hold the CTRL key down while you press the character key.

The circumflex preceding an uppercase character indicates a control character. For example, ^Z indicates the combination of the CTRL key and the Z character and can also be expressed as CTRL/Z. All control key combinations echo on your terminal screen as ^x.

() Square brackets in a command format mean the bracketed parts are optional. For example:

EX[IT]

The characters IT in the EXIT command are optional.

() Braces around two or more items in a command format mean that you must select one of the bracketed items. For example:

dstdevice:=srcdevice:{/FC}
{/NC}

You must include either /FC or /NC in the command.

UPPERCASE Uppercase letters indicate keywords you must enter as shown.

Unless shown otherwise, spaces or tabs must separate items in a command string. The system treats a tab or more than one space between items as a single space.

Most examples in this guide do not show the RETURN key symbol, <RET>. If you try examples at your terminal, always press the RETURN key at the end of a command line unless the example indicates otherwise.

Summary of Technical Changes

The following changes apply to the DECnet/E V4.0 utilities:

- o You cannot use alias node names.
- o You can use long passwords of up to 14 alphanumeric characters.
- o The account from which NPT and FAL run must have total job and detached job quotas of one or more.
- o DECnet/E no longer supports the NETACT remote database.
- o NPT and FAL support the Print/Batch Services (PBS) package for remote PRINT and SUBMIT commands:
 - The NPT PRINT command sends a print job to the NET\$PRINT queue if PBS is running.
 - The NPT SUBMIT command submits a file having a .COM type to the PBS batch server and the NET\$BATCH queue if PBS is running.
 - FAL can perform record-level access to ISAM (RMS-indexed) files.
- o The NETCOPY /DENSITY switch accepts densities of 800, 1600, 6250, 6667, and 8000 bpi.
- o You must have the SHUTDOWN privilege to use the NETOFF utility.

Chapter 1

Introduction

This chapter introduces some basic network concepts.

DECnet is the collective name for the software and hardware products that allow various DIGITAL operating systems to participate in a network. DECnet/E is the implementation of DECnet that allows a RSTS/E operating system to function as a network node.

A DECnet/E node can communicate with other DECnet/E nodes in the network or with any other DIGITAL operating system that supports DECnet.

Using DECnet/E utilities, you can:

- o Communicate with someone at a terminal on another DECnet/E system. Each person can type messages that display at the other's terminal (see Chapter 2).
- o Do work on another DECnet/E system as if your terminal were connected directly to the other system (see Chapter 3).
- o Copy files to or from other DECnet/E systems, submit command files to other DECnet/E systems, and delete files at other DECnet/E systems (see Chapter 4).
- o Copy the contents of an entire storage device to or from another DECnet/E system (see Chapter 5).
- o Shut down your DECnet/E software in an orderly fashion (see Chapter 6).

Figure 1-1 shows the DECnet/E utility functions.

Introduction

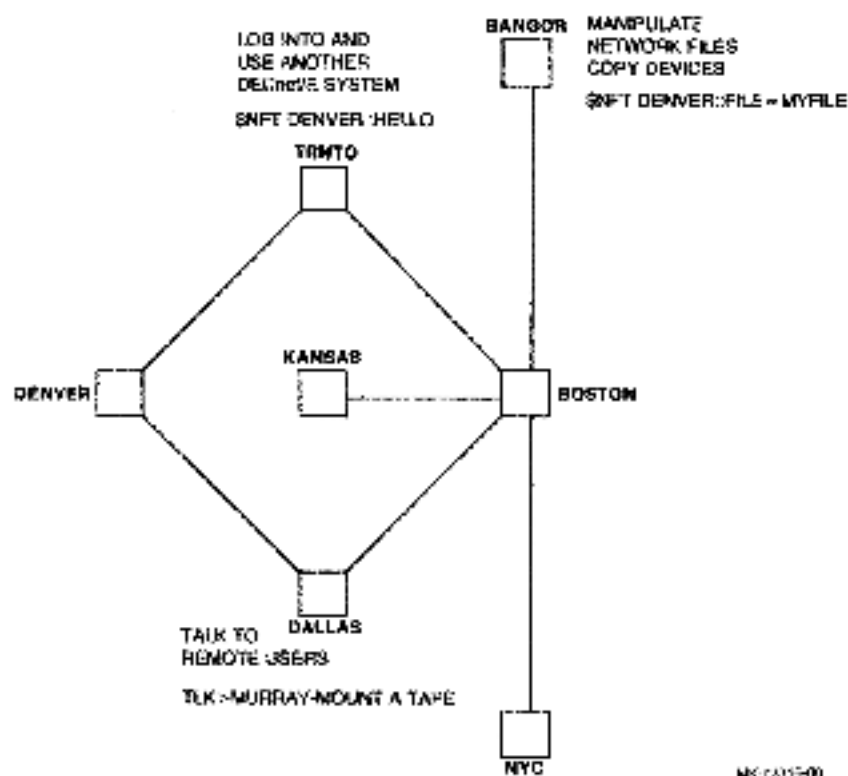


Figure 1-1: DECnet/E Features for the Terminal User

The term DECnet network refers to two or more computer systems running some version of DECnet software, connected by communications links, and sharing system resources. The computer systems, called nodes, need not all be running DECnet/E software. Some other DECnet systems that DECnet/E nodes can communicate with are:

- c DECnet-VAX
- c DECnet-RSX
- c DECnet-Rt
- c DECnet-IAS
- c DECnet-10
- c DECnet-20
- c PRO/DECnet

Although all DECnet systems use the same network concept defined by the DIGITAL Network Architecture (DNA), all do not support the same set of network functions (see the DECnet DIGITAL Network Architecture (Phase IV) General Description). You can use the DECnet/E utilities described in this guide with other DECnet systems as long as those systems support compatible functions. The communications links, called physical links, can be:

- o Cable
- o Microwave
- o Satellite

DIGITAL has developed DECnet products over several years and has introduced new network features periodically through the process of phases. DECnet/E V4.0 is a Phase IV DECnet product, which (with some restrictions) can communicate with earlier DECnet/E Phase III products (see the DECnet/E System Manager's Guide). Note that DECnet/E V4.0 cannot communicate with DECnet/E Phase II products.

In the sample network shown in Figure 1-2, a user at node BOSTON can refer to BOSTON as the local node and to nodes DENVER, DALLAS, NYC, BANGOR, LONDON, and KANSAS as remote nodes. At node BOSTON, the nodes DENVER, DALLAS, and LONDON are physically adjacent nodes. That is, there is a direct physical link to these nodes with no intervening nodes. To a user at node KANSAS, BOSTON is the only adjacent node.

Introduction

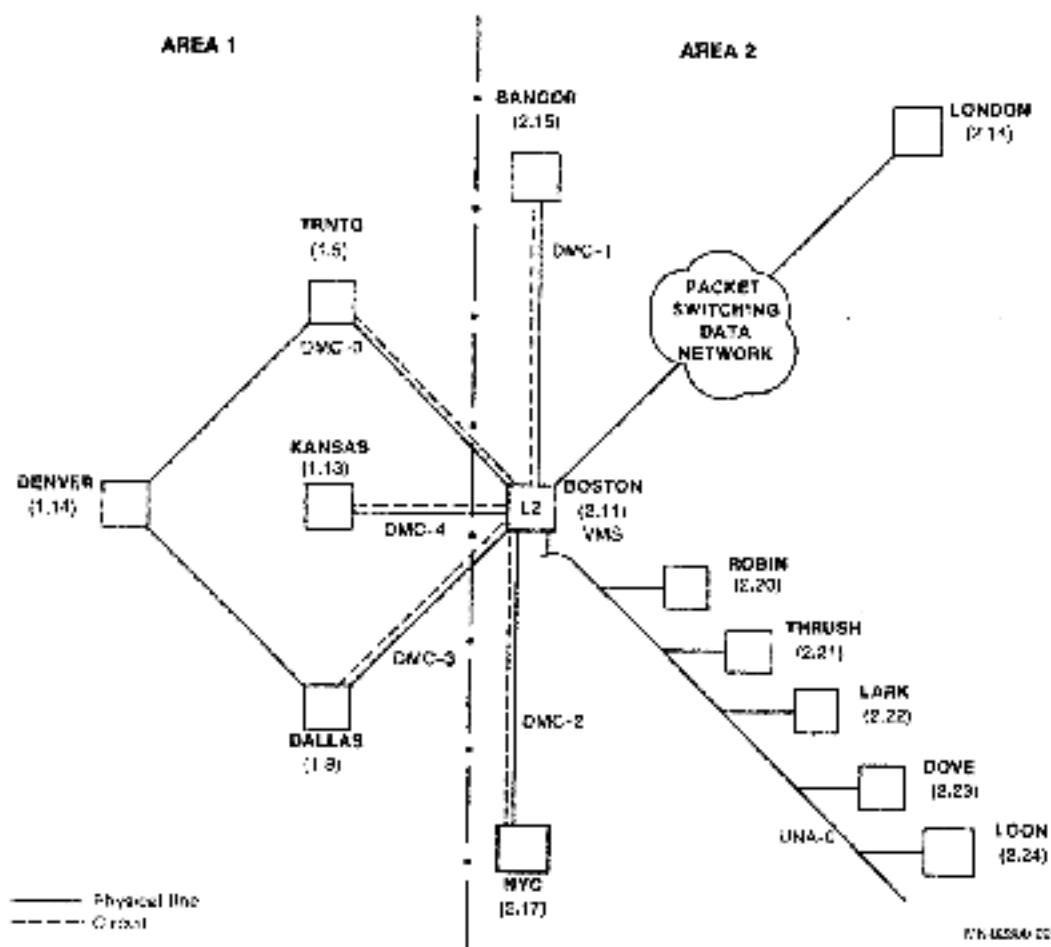


Figure 1-2: Sample Network Composed of Seven Nodes.

Chapter 2

TLK: Terminal Communication Utility

The TLK (Talk) utility lets you send messages to another person at a terminal, including the Operator Services Console (OSC), on any DECnet/E node in the network. Two communication modes are available:

- o One-line mode--Allows transmission of a one-line message to another terminal.
- o Dialog mode--Allows two-way communication between two terminals. The person at either terminal can stop the dialog at any time.

In the RSTS/E environment, the talk utility actually consists of two programs: TLK (talk) and LSN (listen). The sample conversation in dialog mode in Figure 2-1 shows the general relationship of TLK and LSN. As the initiator, you run TLK at your terminal. At the receiver terminal, LSN completes the communications circuit.

TLK: Terminal Communication Utility

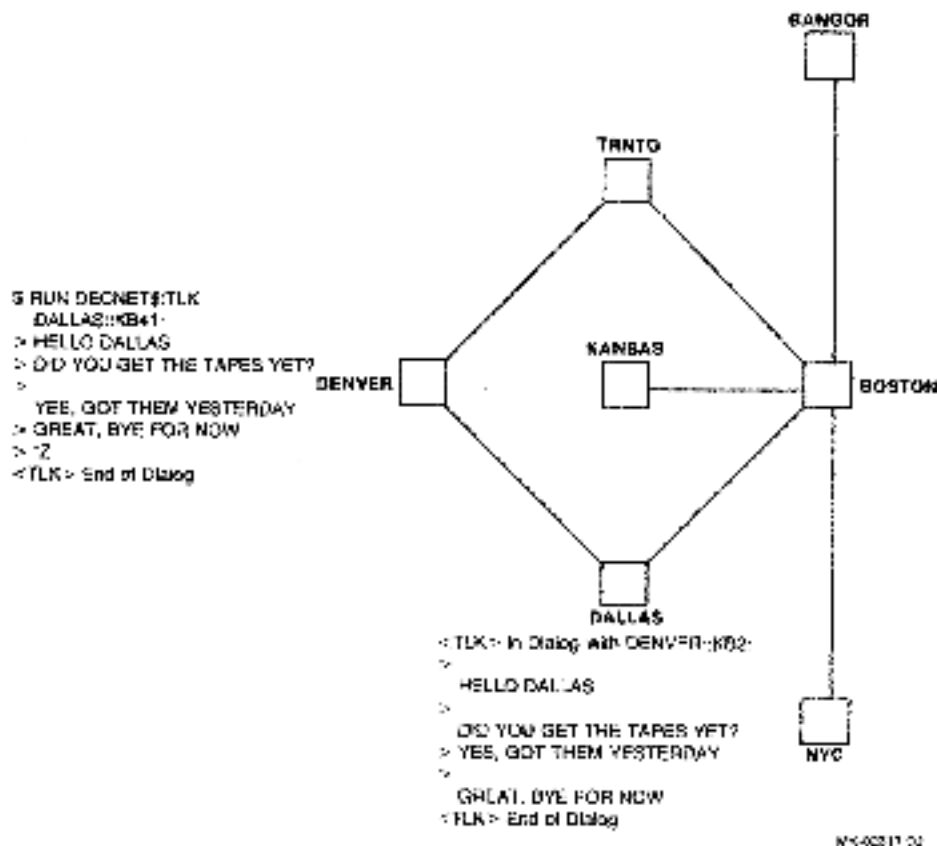


Figure 2-1: The TLK Utility: TLK and LSN Provide Communication Between Terminals

One-Line Mode

Using the TLK utility in one-line mode, you can transmit one line of information to another terminal. The following sections describe the user input required to initiate a message and the output displayed at the receiver's terminal.

Initiator's Input

To send a one-line message to another terminal, type:

```
$RUN DECNET$:TLK
TLK V4.0 RSTS V9.3 Timesharing
*|target-node::||target-terminal:] 'message
```

where:

target-node:: is the one- to six-character alphanumeric node name to which the message is to be sent. If the target node is your local node, you can omit the node name. If you specify a node name, you must include the two colons (::). You can determine the names of nodes currently active on the network by using the NCP command, SHOW ACTIVE NODES. See the DECnet/E System Manager's Guide for a discussion of the SHOW commands.

target-terminal: is the device designation of the terminal to which the message is to be sent (for example, TT5: or KB41:). If you omit target-terminal, TLK sends the message by default to the currently assigned OSC at the specified node or to your local node. Under the RSTS/E system, the system manager can assign the OSC to different physical devices at different times:

If the node you specify is your local node, TLK checks to make sure the target terminal is a terminal known to the system. The usual RSTS/E designation for terminals is Kbn: or Ttn:, where n is the physical terminal number. However, you can also use logical target terminal names for your local node, after assigning them using the ASSIGN command. See the RSTS/E System User's Guide for a discussion of logical device names.

If the node you specify is a remote node, TLK uses only the decimal terminal number from the target terminal specification and forwards that number to the TLK (or LSN) program at the remote node. The remote node's TLK (or LSN) program returns an error message if the device number is invalid for that node.

' message is any ASCII string that is the message you want to send. You must begin the message with a single quotation mark ('). The entire line, including the target specifications and the message, cannot be longer than 132 characters. TLK deletes any leading and trailing spaces and tabs.

If your system manager has defined a Concise Command Language (CCL) command for TLK (TLK for example), you can type the CCL command, the target specifications, and the message all on one line:

TLK [target-node::][target-terminal:] 'message

TLK: Terminal Communication Utility

The following example shows a one-line message sent from local node MAYNRD:: to terminal KB5: at remote node SANFRN:: using the RUN command:

```
$RUN DECNET$;TLK
TLK V4.0 RSTS V9.3 Timesharing
*SANFRN::KB5: 'Relay my calls to (617) 123-4567.  Sheldon.
```

The following example shows a one-line message sent to the OSC at the local node using the CCL command TLK:

```
$TLK 'MOUNT TAPE MM015 ON MM1:
```

Receiver's Output

The receiver's terminal displays a one-line message in this format:

```
<TLK> source-node:: source-terminal: message
```

where:

source-node::	is the name of the node where the message originated. TLK does not display the node name when the target terminal is on the source node.
source-terminal:	is the device designation of the terminal from which the message originated.
message	is the message sent.

The TLK program at the source node sends the message to the RSTS/E target terminal. If the target terminal is busy, LSN displays the message but does not stop the executing program. The following example shows the displayed message sent by terminal KB4: on node MAYNRD::

```
<TLK> MAYNRD:: KB4: 'Relay my calls to (617)123-4567.  Sheldon.
```

If a message arrives for the local node's OSC while the Operator Services (OPSER) program is running, TLK displays the message in the OPSER format:

```
$MESSAGE 23402 : 24-Sep-86 12:45 PM JOB:13 LSN[1,2]
<TLK> KB4:
'MOUNT TAPE MM015 ON MM1:
```


If CPSEB is not installed on the local node, or if for some reason it is not running, TLK displays the message on the OSC in the TLK format:

```
$<TLK> KB4: 'MOUNT TAPE MM015 ON MN1:
```

Dialog Mode

When you use TLK in the dialog mode, you can exchange messages with a user at another terminal.

Initiator's Input

To initiate a dialog with a person at another RSTS/E terminal, type:

```
SRUN DECNETS:TLK
TLK V4.0 RSTS V9.3 Timesharing
*[target-node:;](target-terminal:;
>message
```

where:

target-node:;	is the one- to six-character alphanumeric node name to which the message is to be sent. If the target node is your local node, you can omit the node name. If you specify a node name, you must include the two colons (:;). You can determine the names of nodes currently active on the network by using the NCP command, SHOW ACTIVE NODES. See the DECnet/E System Manager's Guide for a discussion of the SHOW commands.
target-terminal:;	is the device designation of the terminal to which the message is to be sent (for example, TT5; or KB4;). If you omit target-terminal, the default target is OSC. (TLK automatically sends the message to the OSC at the specified node or to your local node.) Under the RSTS/E system, the system manager can assign the OSC to different physical devices at different times.

If the node you specify is your local node, TLK checks to make sure the target terminal is a terminal known to the system. The usual RSTS/E designation for terminals is KBn; or TTn;, where n is the physical terminal number. However, you can also use logical target terminal names for your local node, after assigning them using the ASSIGN command. See the RSTS/E System User's

TLK: Terminal Communication Utility

Guide for a discussion of logical device names.

If the node you specify is a remote node, TLK uses only the decimal terminal number from the target terminal specification and forwards that number to the TLK (or LSN) program at the remote node. The remote node's TLK (or LSN) program returns an error message if the device number is invalid for that node.

Message is any ASCII string that is the message you want to send. The entire line cannot be longer than 132 characters.

Wait for the right angle bracket (>) prompt before typing the first line of the message. TLK continues prompting for new lines with >. TLK alternates returned messages from the receiver end with the lines you type. Either you (the initiator) or the receiver can terminate the dialog by pressing CTRL/Z at the > prompt.

If your system manager has defined a Concise Command Language (CCL) command for TLK (TLX for example), you can type the CCL command and the target specifications on one line and then type your message at the > prompt:

```
$TLK [target-node::][target-terminal:]  
>message
```

The following example shows how the dialog looks at the initiator's terminal and how the conversation proceeds. The receiver terminates the dialog:

```
$RUN DECNET$TLK  
TLK V4.0 RSTS V9.3 Timesharing  
*MAYNRD::KB<:  
>Sheldon--a call came long-distance; couldn't be transferred.  
>  
Oh. Who was it? Did they leave a number to call back?  
>Yes--Gladys Freen. Call back at  
>(415)987-6543 after 10:00 (1:00 pm your time).  
>  
OK--thanks again. Sheldon.  
<TLK> End of Dialog  
$
```

Receiver's Interaction

If the target terminal is not busy (no one is logged in), the LSN program displays the first line of a dialog-mode message as follows:

```
<TLK> in Dialog with source-node::source-terminal:
>
message-1
>
```

where:

source-node	is the name of the node from which the dialog originated.
source-terminal	is the terminal designator that originated the dialog.
message-1	is the first message line of the dialog.

The receiver then continues the dialog by pressing RETURN at the > prompt. For example:

```
<TLK> in dialog with NOD5::KB12:
```

If the target terminal is busy (someone is logged in at the terminal), the LSN program displays a message in the following format:

```
<TLK> in Dialog with source-node::source-terminal:
Type 'TLK/n source-node:: source-terminal:' to continue dialog
```

where:

source-node	is the name of the node from which the dialog originated.
source-terminal	is the terminal designator that originated the dialog.
n	is the job number of the receiver's LSN program. Include at least one space between the job number and the source node.

The text enclosed in single quotation marks is what the receiver must type to continue the dialog. LSN then waits up to two minutes for the receiver to continue the dialog. The receiver can stop any active

TLK: Terminal Communication Utility

work at the terminal and type the indicated command to continue the dialog or go to another terminal, log in, and type the indicated command to continue the dialog. For example:

```
TLK/6 NOD5::KB12:
```

The receiver can also choose to continue the dialog using the RUN DECNET\$::TLK command and then typing the continuation command at the * (asterisk) prompt:

```
RUN DECNET$::TLK
TLK V4.0 RSTS V9.3 Timesharing
*/6 NOD5::KB12:
```

After the receiver types the continuation command, LSN displays the first message and the dialog continues in the usual manner until either user presses CTRL/Z at the > prompt. LSN then breaks the connection and displays the following message:

```
<TLK> End of Dialog
$
```

If the target terminal is the local OSC and OPSEK is running, the display for a busy terminal is made in the OPSEK format:

```
$MESSAGE nnnnn:date time JOB:nn jobname(ppn)
<TLK> in Dialog with source-node:: source-terminal:
Type 'TLK/n source-node:: source-terminal:' to continue Dialog
```

Operation then proceeds the same as for a busy terminal.

The following example shows how the dialog looks at the receiver's terminal. The user is logged in and is using a text editor:

```
.
.
.
*!And so, after/
<TLK> in Dialog with SANFRN::KB5:
Type 'TLK/13 SANFRN::KB5:' to continue dialog

*exit
.    (user exits
.    from the editor
.    and starts TLK)

$TLK/13 SANFRN::KB5:
```

>
Sheldon--a call came long-distance; couldn't be transferred.
>Oh, who was it, did they leave a number to call back?
>
Yes--Gladya Freen. Call back at
>
(415)987-6543 after 10:00 (1:00 pm your time).
>OK--thanks again. Sheldon.
>CTRL/Z
<TLK> End of Dialog
\$

TLK Error Messages

Table 2-1 lists the TLK error messages.

Table 2-1: TLK Error Messages

Message and Meaning	
COMMAND SYNTAX ERROR-ILLEGAL JOB NUMBER - /N	The job number you typed to continue a dialog is not a valid RSTS/E job number. Check the display that TLK printed and retype the line to continue the dialog.
COMMAND SYNTAX ERROR - ILLEGAL MESSAGE	You tried to type a message on the same line as the COMMAND line for continuing a dialog. Retype the command line. When TLK prints the > prompt, you can then type the message you want sent.
COMMAND SYNTAX ERROR - ILLEGAL NODE NAME - NODENAME	TLK does not recognize the node name you specified in a command line. Retype the line with the correct node name.
COMMAND SYNTAX ERROR - ILLEGAL TERMINAL SPECIFICATION - SPEC	The terminal specification shown (spec) is incorrect. Retype the command line.
COMMAND SYNTAX ERROR - JOB N IS NOT A MESSAGE RECEIVER	You typed a valid job number to continue a dialog but the job is not a message receiver. Check the display that TLK printed and retype the line to continue the dialog.
COMMAND SYNTAX ERROR - JOB N DOES NOT MEET DIALOG REQUIREMENTS	You typed a valid job number to continue a dialog and the job is a message receiver, but the job is not the same one that initiated the dialog. Check the display that TLK printed and retype the line to continue the dialog.
COMMAND SYNTAX ERROR - MESSAGE TOO LONG	The maximum line length that TLK accepts is 132 characters. Retype the message using shorter lines.
CONNECT INITIATE ERROR - TEXT	TLK tried to set up a logical link with the remote TLK or LSN program to exchange messages, but it received an error on the initial connection request. The text is the error text TLK received in response to its attempt to send a Connect Initiate Message. See the DECnet/E Network Programmer's Reference Manual for a further description of the error text.

Table 2-1: TLK Error Messages (Cont.)

Message and Meaning
CONNECTION REJECTED BY NSP - CODE:N TLK tried to set up a logical link with the remote TLK or LSN program to exchange messages but the connection was rejected by the remote NSP. The code n is the reason code that the remote NSP returned. See Appendix A for a list of the NSP reason code meanings.
CONNECTION REJECTED BY REMOTE TLK - CODE:N TLK tried to set up a logical link with the remote TLK or LSN program to exchange messages but the connection was rejected by the remote TLK or LSN program. The remote TLK then returned the code n. Contact a DIGITAL Software Support representative if this error occurs often.
CONNECTION REJECTED BY REMOTE TLK - ILLEGAL VALUE IN TLK MODE BYTE TLK tried to set up a logical link with the remote TLK or LSN program to exchange messages but the connection was rejected by the remote TLK or LSN program. Contact a DIGITAL Software Support representative if this error occurs often.
CONNECTION REJECTED BY REMOTE TLK - TERMINAL IS BUSY You initiated a dialog and the remote terminal is busy, but no one has responded to continue the dialog. The remote user has two minutes to respond and continue the dialog; in this instance, no one responded.
CONNECTION REJECTED BY REMOTE TLK - TERMINAL IS NONEXISTENT The remote terminal you requested does not exist at the remote node. Check for possible error in the command line.
CONNECTION REJECTED BY REMOTE TLK - TEXT The remote TLK or LSN program did not accept the logical link connection that allows the one-line or dialog exchange. The text is the error message that the remote TLK or LSN program received in trying to issue its connect confirmation. Contact a DIGITAL Software Support representative if this error occurs often.
CONNECTION REJECTED BY REMOTE TLK - UNKNOWN FORMAT The remote TLK program did not recognize the parameters passed in the request to set up a logical link that allows the one-line or dialog exchange. This occurs when the remote TLK program is part of an earlier DECnet implementation such as DECnet Phase II. Contact a DIGITAL Software Support representative.

Table 2-1: TLK Error Messages (Cont.)

Message and Meaning
DECLARE RECEIVER ERROR - TEXT The local TLK program could not begin to set up message exchange facilities at the local node. The text is the error message received in its Declare Receiver system function call.
DECNET IS NOT INSTALLED DECnet was not installed at system installation time. You cannot execute any network functions.
DIALOG ALREADY CONTINUED BY JOB N, ACCOUNT [PPN], FROM KBN: You or someone else has already typed the line requesting a dialog continuation. You can use the account number and terminal number to determine who continued the dialog and from which terminal.
DISCONNECTED BY REMOTE TLK - CODE:N The remote TLK program disconnected the logical link that allows the one-line or dialog exchange. The remote TLK program then returned the code n. Contact a DIGITAL Software Support representative if this error occurs often.
DISCONNECTED BY REMOTE TLK - TRANSFER ERROR An error occurred at the remote node in transferring a message to or from the remote terminal. Contact a DIGITAL Software Support representative if this error occurs often.
JOB N IS NOT IN DIALOG WITH SPECIFIED TARGET In continuing a dialog, you typed the wrong terminal number or node name. Check the message that TLK displayed and retype the command line.
LINK ABORTED BY NSP - CODE:N The logical link allowing the one-line or dialog exchange has been aborted. The code n is the reason code returned by NSP in the Link Abort Message terminating the link. See Appendix A for a list of the NSP reason code meanings.
LINK COULD NOT BE ESTABLISHED - TIMEOUT Two minutes elapsed after the remote TLK program printed its message for continuing the dialog and no one responded. The dialog was terminated.

Table 2-1: TLK Error Messages (Cont.)

Message and Meaning
LINK COULD NOT BE ESTABLISHED - TEXT
The local TLK program received an error in the Connect Confirm message from the remote TLK or LSN program. The logical link that allows the one-line or dialog exchange was not set up. The text is the NSP error message that the local TLK program received in response to its Receive system function call. Contact a DIGITAL Software Support representative.
MESSAGE TOO LONG
You typed a line in dialog mode that was longer than 132 characters. Retype the message using shorter lines.
NSP IS NOT ENABLED
NSP, the part of DECnet that allows the logical link connections providing the one-line or dialog exchange, was not enabled at the local node. See your system manager.
RECEIVE ERROR - TEXT
The local TLK or LSN program received an error while processing a message from the remote node. The text is the error message received in response to the Receive system function call. Contact a DIGITAL Software Support representative.
SEND DATA ERROR - TEXT
The local TLK or LSN program received an error while sending a message to the remote node. The text is the error message received in response to the Send Network Data message system function call. See the DECnet/E Programmer's Reference Manual for further details of the error text, and contact a DIGITAL Software Support representative.

Chapter 3

NET: Network Command Terminal Utility

The NET utility lets you access another DECnet/E system in the network and do work there as if your terminal were directly connected to that system.

NET is implemented as two programs: NET and NPKDVR (Network Pseudo Keyboard Driver). Figure 3-1 shows a user running NET at node DENVER. NET requests that a logical link be set up with NPKDVR at the remote node BOSTON. NPKDVR is a job that opens a pseudo keyboard at node BOSTON and accepts the logical link connection request. After NET sets up the logical link, NET relays the information you type at node DENVER over the logical link to NPKDVR, which relays it to the pseudo keyboard. The RSTS/E system at remote node BOSTON treats a pseudo keyboard as a normal keyboard.

When the remote node RSTS/E monitor receives information from the pseudo keyboard (whatever you type at node DENVER), the monitor creates a job and runs LOGIN to process the input. The remote node system directs responses generated by the monitor, LOGIN, and any other programs you run at the remote node to the pseudo keyboard. NPKDVR, in turn, passes responses directed to the pseudo keyboard back over the logical link to the NET program at node DENVER. NET then displays the responses on your terminal.

NET passes your input directly through to NPKDVR at the remote node, except for CTRL/P and its related commands. NET makes no attempt to interpret any of the data it passes through. Thus, if you type a system-level command, for example, the remote system executes the command and the command has no effect on your system.

To reduce overhead, system managers can install more than one permanent copy of NPKDVR at a node. Each permanently installed copy can handle multiple logical links. When all permanently installed copies are busy, an incoming connection request makes the system start a temporary copy. See the discussion of the SET/DEFINE OBJECT command in the DECnet/E System Manager's Guide for more information about installing programs for automatic start-up.

NET: Network Command Terminal Utility

```
$ RUN DECNETS:NET
```

```
Node Name: BOSTON
```

```
Connection Established to RSTS/E node BOSTON
```

```
RSTS V8.0 10-JUN-86 03:47 PM
```

```
System Password:
```

```
User: T00.1
```

```
Password:
```

```
$
```

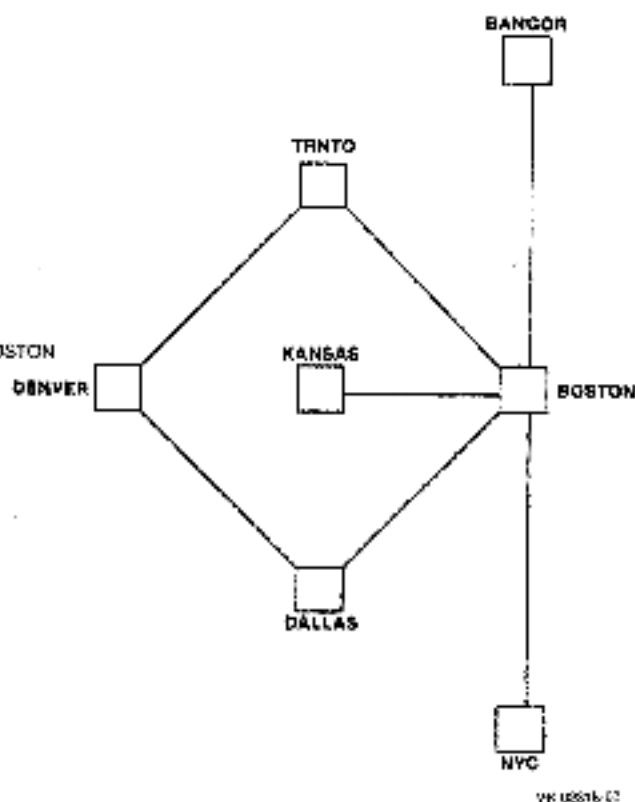


Figure 3-1: NET Provides Access to Another DECnet/E Node

Running NET

You can run NET from any terminal in a DECnet/E system by typing:

```
RUN DECNETS:NET
```

or

```
SET HOST [node-name]
```

or

```
NET [node-name]
```

where [node-name] is the name of the remote node you want to access.

If you do not include the node name on the command line, NET will prompt you for it. NET displays a connection message, the remote node system banner, and prompts for login information. For example:

```
SRUN DECNET$;NET
NET V4.0 RSTS V9.3 DECnet System
Node name? BOSTON
Connection established to RSTS/E Node BOSTON
RSTS V9.3 Timesharing Job 31 KB33 24-Sep-86 8:00AM
User: 2,250
Password:
$
```

If you include the node name on the command line, NET displays its banner, a connection message, the remote system banner, and prompts for the login information. For example:

```
$ SET HOST BOSTON
NET V4.0 RSTS V9.3 DECnet System
Connection established to RSTS/E Node BOSTON
RSTS V9.3 Timesharing Job 31 KB33 24-Sep-86 8:00AM
User: 2,250
Password:
$
```

If the remote node requires a system password, you can optionally include only the node name on the command line. You must include a space or tab between the command and the node name. In this case, NET displays its banner, a connection message, the remote system banner, and prompts for the login information. For example:

```
$ RUN DECNET$;NET BOSTON
NET V4.0 RSTS V9.3 DECnet System
Connection established to RSTS/E Node BOSTON
RSTS V9.3 Timesharing Job 31 KB33 24-Sep-86 8:00AM
System password:
User: 2,250
Password:
$
```

If the remote node requires a system password and you do not include the node name on the command line, NET displays its banner and prompts for the node name. For example:

```
$ RUN DECNET$;NET
NET V4.0 RSTS V9.3 DECnet System
Node Name? BOSTON
```

NET: Network Command Terminal Utility

NET then displays a connection message, the remote system banner, and prompts for the log-in information. For example:

```
Connection established to RSTS/E Node BOSTON
RSTS V9.3 Timesharing Job 31 KB33 24-Sep-86 8:00AM
System password:
User: 2,250
Password:
$
```

If your system manager has defined a CCL command for DECNETS:NET (NET for example), you can run NET by typing:

```
NET (node-name)
```

NET then proceeds as described in the previous paragraphs.

If your node is running the DCL Run-Time System, you can also connect to a remote DECnet/E node by using the DCL command SET HOST (see the *RSTS/E Quick Reference Guide* and the *RSTS/E System User's Guide*).

NET Commands

After running NET, your local terminal is logically connected to the last remote node you specified and sends commands directly to that remote node. Your local terminal displays the system prompt for that remote node and you can type any of the system commands available to you on that node. However, you may want to send commands directly to a remote node between your node and the last one in the chain of remote node connections, or you may want use the single-character transmission ODI mode. You use the NET commands to perform these functions.

Before you can use any of the NET commands, you must first communicate with your local node and indicate that you want to enter the NET command mode. You communicate this to your local node by pressing CTRL/P, then RETURN. NET displays its nodename::NET> prompt and waits for you to type a NET command. For example:

```
$ RUN DECNETS:NET
NET V4.0 RSTS V9.3 DECnet System
Node name? BOSTON
Connection established to RSTS/E Node BOSTON
RSTS V9.3 Timesharing Job 31 KB33 24-Sep-86 8:00AM
System password: (Not echoed)
User: 2,250
Password: (Not echoed)
$ ^P<RET>
DENVER::NET>
```

In this example, you run NET from your local node DENVER and log in to remote node BOSTON. You then enter the NET command mode by pressing CTRL/P, then RETURN. Table 3-1 lists all seven NET commands you can now type.

Table 3-1: NET Commands

Command	Meaning
CO[NTINUE]	Return to terminal mode.
CT[RL/P]	Send a ^P to the next remote node in a chain of remote nodes.
DE[TACH]	Abort link (detaching remote job) and exit.
EX[IT]	Kill remote job, disconnect the link, and exit.
HE[LP] or ?	Display the help text.
NO[RMAL]	Exit ODT mode.
OD[T]	Enter ODT mode.

NET: Network Command Terminal Utility
CONTINUE

CONTINUE Command

Use the CONTINUE command to cause the NET program to stop displaying its nodename::NET> prompt and resume sending the data you type at your local terminal over the connection to the remote node you were last at. After typing CONTINUE, the cursor drops down to the next line. However, the system does not display its prompt. You can type system commands at this point. For example:

```
RUN DECNETS:NET
NET V4.0 RSTS V9.3 DECnet System
Node name? BOSTON
Connection established to RSTS/E Node BOSTON
RSTS V9.3 timesharing Job 31 FB33 24-Sep-86 10:00AM
System password:
User: 3,247
Password:
$ ^?<RET>
DENVER::NET>HELP
```

(Help display for NET commands)

```
DENVER:;NET>CONTINUE
<RET>
$ DIRECTORY
```

(Directory listing for user
at node BOSTON)

```
$ (system prompt at node BOSTON)
```

In this example, you use the `CONTINUE` command to stop the `DENVER::NET>` prompt. You then ask the Boston node for a directory display.

CTRL/P Command

Use the CTRL/P command to pass a CTRL/P character to an intermediate remote node in a chain of remote nodes. This command is useful when you have set up a chain of three or more remote nodes and you want to operate in the NET command mode at one of the intermediate remote nodes.

To use this command you must type the characters CTRL/P, then press RETURN while you are at the nodename::NET> prompt. NET then displays the nodename::NET prompt of the next remote node in the chain.

Note

This command is spelled C T R L / P . To enter this command, you must type the characters "CTRL/P" rather than pressing the CTRL key and then pressing P.

By repeatedly using the CTRL/P command you can logically connect your local terminal to any intermediate remote node in the chain. For example:

```
$ RUN DECNET$::NET
NET V4.0 RSTS V9.3 DECnet System
Node Name? BOSTON
Connection established to RSTS/E Node BOSTON

RSTS V9.3 Timesharing Job 31 KB33 24-Sep-86 8:00AM
System password:
User: 2,250
Password:
$ RUN DECNET$::NET
NET V4.0 RSTS V9.3 DECnet System
Node Name? TULSA
Connection established to RSTS/E Node TULSA

RSTS V9.3 Timesharing Job 31 KB33 24-Sep-86 9:01AM
System password:
User: 3,120
Password:
$ ^P<RET>
DENVER::NET>CTRL/P
BOSTON::NET>
```

In this example, you run NET at your local node DENVER to set up a connection with BOSTON. At node BOSTON, you run NET to set up a connection with TULSA. While logically connected to node TULSA, you press CTRL/P and RETURN to enter the NET command mode at DENVER. You then type out CTRL/P to enter the NET command mode at node BOSTON. Now you can use any of the NET commands at BOSTON.

NET: Network Command Terminal Utility

DETACH

DETACH Command

Use the DETACH command to detach the job at a remote node, disconnect the link between your node and the remote node, and exit the NET program. For example, you can use the DETACH command to run a large command procedure at a remote node and then return to your local node to continue work. Later you can set up the remote connection again and attach to the job that detached when you used the DETACH command. For example:

```
$ RUN DECNETS:NET
NET V4.0 RSTS V9.3 DECnet System
Node Name? BOSTON
Connection established to RSTS/E Node BOSTON

RSTS V9.3 Timesharing Job 31 KB33 24-Sep-86 8:00AM
System password:
User: 2,250
Password:
$ ^P<RET>
$ @BIGJOB.COM
$ ^P<RET>
DENVER::NET>DETACH
Remote job detached...
Control returned to node DENVER
```

(continue work at node DENVER)

```
$ RUN DECNETS:NET
NET V4.0 RSTS V9.3 DECnet System
Node Name? BOSTON
Connection established to RSTS/E Node BOSTON

RSTS V9.3 Timesharing Job 31 KB33 24-Sep-86 8:00AM
System password:
User: 2,250
Password:
Jobs detached under this account:
  Job What Size State Run-time RTS
  18 DCL  4K  HB      3.0  DCL
Job number to attach to? 18
Attaching to job 18
$ (do work at node BOSTON)
```

In this example, you submit BIGJOB to the BOSTON node, then use the DETACH command to disconnect and go back to the DENVER node. Later, you reconnect to BOSTON and attach to BIGJOB's job number again.

EXIT Command

Use the EXIT command to break the connection with the remote node. There are two ways to terminate the NET program:

- o Simply log out at the remote node.
- o Use the EXIT command.

In the second case, NPEDVR at the remote node automatically terminates the job to which you are connected. NET then displays the following message:

Control returned to node nodename

The following example shows how to log out and use the EXIT command to terminate the NET program:

```
$ RUN DECNET$::NET
Node Name? BOSTON
Connection established to RSTS/E Node BOSTON
RSTS V9.3 Timesharing Job 31 KB33 24-Sep-86 8:00AM
System password:
User: 2,613
Password:
$ RUN DECNET$::NET
Node Name? TULSA
Connection established to RSTS/E Node TULSA
RSTS V9.3 Timesharing Job 22 KB17 24-Sep-86 9:01AM
System password:
User: 2,210
Password:
$
```

(do work at node TULSA)

```
$ BYE
Control returned to node BOSTON
$ ^P<RET>
DENVER::NET>EXIT
Control returned to node DENVER
$
```

In this example, you net over from Denver to the Boston node and then to the Tulsa node. Later, you log off the Tulsa node, then use the EXIT command to stop the NET program and leave the Boston node for the Denver node.

NET: Network Command Terminal Utility
HELP

HELP Command

Use the HELP command at the nodename::NET> prompt to get the help display of the NET commands at your terminal. You can also type a question mark (?) to obtain the help display. After NET displays the help information, NET returns to its nodename::NET> prompt. For example:

```
$ RUN DECNET:SNET
NET V4.0 RSTS V9.3 DECnet System
Node name? BOSTON
Connection established to RSTS/E Node BOSTON

RSTS V9.3 Timesharing Job 25 KB27 24-Sep-86 10:23AM
System password:
User: 1,221
Password:
$ ^P<RET>
DENVER::NET>HELP
```

NETWORK commands

```
CO[NTINUE] Return to terminal mode.
DE[ATCH]   Abort link (detaching remote job) and exit.
EX[IT]     Kill remote job, disconnect link, and exit.
HE[LP]     Print help text.
OD[T]      Enter ODT mode.
CT[RL/P]   Send a "^P" to other end.
NO[RMAL]   Exit ODT mode.
```

DENVER::NET>

In this example, you net over from Denver to the Boston node, use CTRL/P to return to the NET program and use the HELP command. NET produces the help display and goes back to the nodename::NET> prompt.

NORMAL Command

Use the NORMAL command immediately after exiting from a program that needs single-character input, such as ODT (Octal Debugging Tool). For example:

```
$ RUN DECNET:$NET
NET V4.0 RSTS V9.3 DECnet System
Node name? BOSTON
Connection established to RSTS/E Node BOSTON

RSTS V9.3 Timesharing Job 25 KB27 24-Sep-86 10:23AM
System password:
User: 1,221
Password:
$ ^P<RET>
DENVER::NET>ODT
.
.
.
(do ODT work at node BOSTON)
.
.
.
$ ^P<RET>
DENVER::NET>NORMAL
^P<RET>
DENVER::NET>EXIT
Control returned to node DENVER
$
```

In this example, you net over from the Denver node to the Boston node and use ODT there. You then use CTRL/P to return to the NET program, then use the NORMAL command to leave the single-character mode, then use another CTRL/P to return to the DENVER::NET> prompt.

NET: Network Command Terminal Utility
ODT

ODT Command

Use the ODT command when you use programs such as RSTS/E ODT and EDT that do not wait for an entire line but process characters as you type them.

Normally, RSTS/E programs receiving input from terminals accept an entire line at a time where a line is a string of characters terminated by a RETURN/LINE FEED combination, an escape character, and so forth. However, a few programs do not wait for an entire line but process characters immediately. When you run such a program at a remote node, you must first issue the ODT command to the NET program at the local node (and any intermediate remote nodes). NET then forwards characters to the remote node as you type them. For example:

```
$ RUN DECNET$:NET
NET V4.0 RSTS V9.3 DECnet System
Node name? BOSTON
Connection established to RSTS/E Node BOSTON
RSTS V9.3 Timesharing Job 25 KB27 24-Sep-86 10:23AM
System password:
User: 1,221
Password:
$ RUN AUXLIB$:ODT
* ^P<RET>
DENVER::NET>ODT
210/ 1077
*
```

In this example, you run NET at the local node (DENVER), requesting a connection to node BOSTON. You then run the RSTS/E utility ODT at node BOSTON and when ODT displays its prompt, you press CTRL/P, then RETURN to return to the NET command level at node DENVER. NET responds with its nodename::NET> prompt and you type ODT. NET processes the command and returns you to remote node BOSTON. You type an ODT command, ODT on node BOSTON receives it, and processing continues as normal.

Note

Using this feature places a heavy load on the network because DECnet/E must send many more messages over the link than it would normally. Use the ODT mode with discretion.

NET Error Messages

Table 3-2 lists the NET error messages.

Table 3-2: NET Error Messages

Message and Meaning
?CONNECT FAILURE - INVALID NODE NAME NODENAME The node name you gave in response to the Node name? prompt is not valid. There is no such node in the network. Run NET again, specifying a valid node name.
?CONNECT FAILURE - NODE SHUTTING DOWN The node you requested is scheduled for shutdown. No new logical links to the node are allowed. Try again later.
?CONNECT FAILURE - TOO MANY LINKS TO NODE NODENAME Network traffic to the node you requested is too heavy at this time to establish another logical link. Try again later.
?CONNECT REJECT - DESTINATION NODE IS UNREACHABLE The remote node you tried to access is not currently reachable. Try again later.
?CONNECT REJECT - NSP REASON CODE = N The remote node rejected NET's request for a logical link. See Appendix A for a list of the NSP reason code meanings.
?ILLEGAL COMMAND The NET utility does not recognize the command you typed in response to a nodename::NET> prompt. Type HELP or ? to see a display of valid NET commands.
?NETWORK NOT AVAILABLE NET is unable to make a logical link to the node you requested. For example, the node may be shut down for maintenance, or the physical links may be out of order. Try again later.

Chapter 4

NFT: Network File Transfer Utility

The main function of the Network File Transfer utility (NFT) is to move files from one node to another (see Figure 4-1). You can run NFT from a terminal to perform the following functions:

- o Copy files from a storage device at one node to a storage device or output device at another node.
- o Copy and append one or more files stored at one node to a file stored at another node.
- o Copy and submit a command file stored at one node to another node's batch processor or DCL command interpreter for execution. On RSTS/E nodes, NFT or FAL submits such files to the batch processor.
- o Execute an existing command file at a remote node.
- o Delete files stored on disk at a remote node.
- o Copy and spool print files from one node to another node's printer.
- o Display a directory listing of files stored on disk at a remote node.

NFT: Network File Transfer Utility

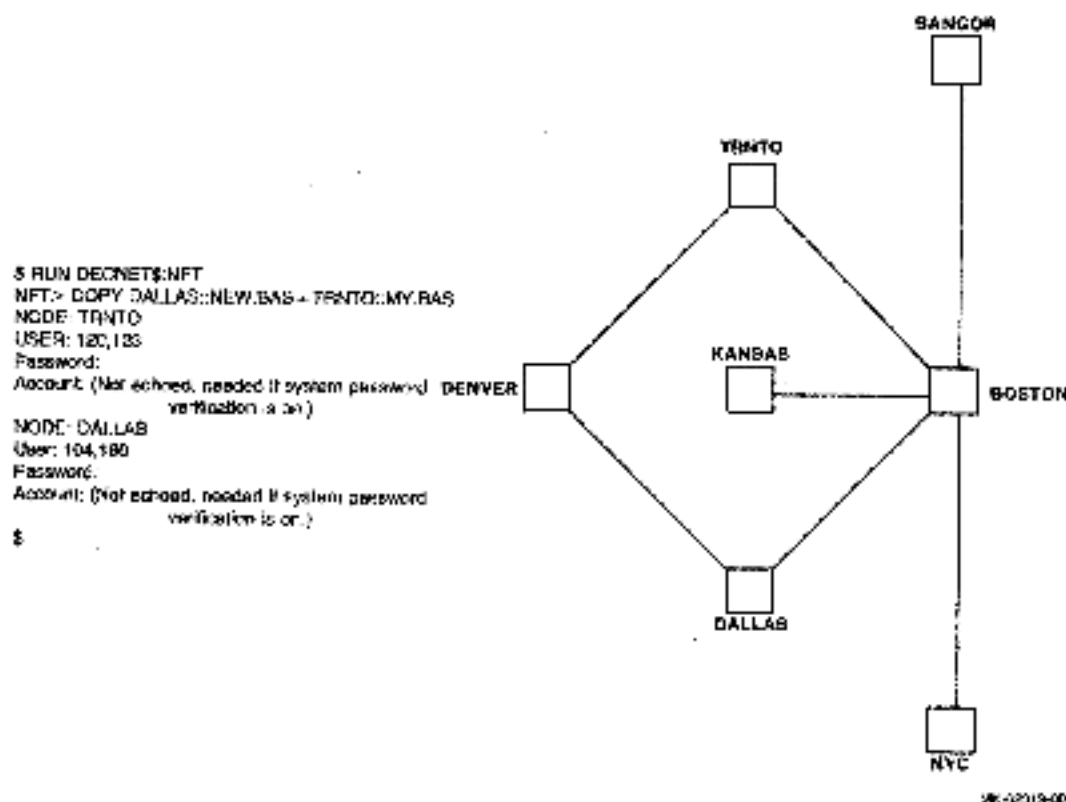


Figure 4-1: NFT Transfers Files to, from, and between Remote Nodes

Note

Although NFT is written to conform to the Data Access Protocol (DAP), it supports only the executable bit during file transfers; it does not pass file protection codes. Thus, files created as a result of NFT commands have a protection code of either <60> or <124>, depending on the presence of the executable bit.

The Relationships Between NFT, FAL, and RMS

It is easier to understand what NFT does by beginning with a brief description of how DECnet/E implements the file transfer capability.

The NFT functions listed at the beginning of this chapter are the result of the relationship between:

- o NFT
- o File Access Listener (FAL)
- o Record Management Services (RMS)

NFT establishes a logical link with the FAL program at the remote node. NFT then accepts commands typed at the local node and transmits them to FAL. The FAL program, a part of DECnet/E, executes the commands. NFT and FAL communicate with each other in a language defined by the Data Access Protocol (DAP). Thus, even though the user commands can be different for various DECnet/E implementations of NFT, all DECnet/E systems having NFT and FAL can communicate to do file handling.

You must make sure that the account from which NFT and FAL run have detached job and total job quotas of one or more. Your system manager sets these quotas.

RMS is an operating system file service feature used by NFT and FAL at DECnet/E nodes to execute input and output to the local I/O devices. This is of interest mainly because non-RSTS/E nodes can use other means to do their local file I/O and can require some of the NFT switches (see the "NFT Switches" section). These switches ensure that NFT transfers files between nodes having different operating systems and file handlers in a usable format. Figure 4-2 shows the interaction between NFT, FAL, and RMS at two DECnet/E nodes.

NFT: Network File Transfer Utility

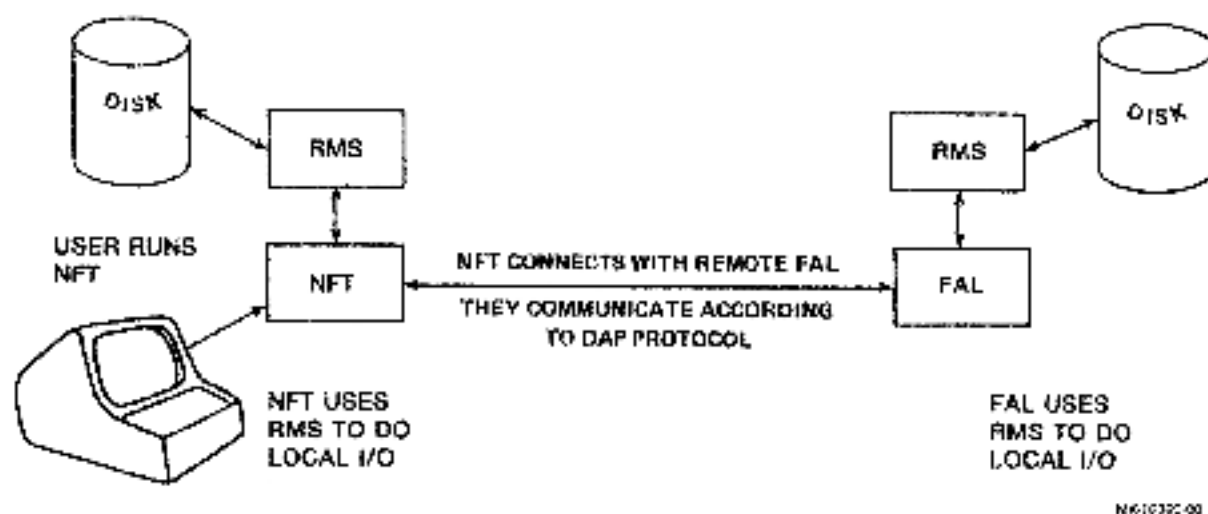


Figure 4-2: At DECnet/E Nodes, NFT, FAL, and RMS Work Together to Allow File Operations

FAL Access to ISAM Files

You can access ISAM (RMS-indexed) files on nodes running DECnet/E (see the RMS-11 documentation). FAL can perform the following:

- c File access operations:
 - Create an indexed file
 - Open an indexed file for read or write access
- c Record access operations:
 - Delete the current record
 - Find a record using keyed or sequential access
 - Write (flush) out all modified buffers
 - Unlock (free) all locked records
 - Get a record using keyed access
 - Put a record using keyed or sequential access
 - Update the current record

FAL has a limited amount of memory for Key Definition Attribute messages and Allocation Attribute messages. If FAL runs out of memory

for message storage (probably due to the file having too many keys defined), FAL issues a file open error message.

FAL does not support multiple data streams on a single file. The maximum record length that FAL can process is 512 bytes. FAL cannot use record access for files that have larger records. However, you can copy larger records from one RSTS/E system to another by using the NFT /BLOCK switch (see the "NFT Switches" section).

Running NFT

You can run NFT from any terminal in a DECnet/E system by typing:

```
RUN DECNET$NFT
```

If your system manager has installed NFT as a Concise Command Language (CCL) command, you can type:

```
NFT
```

In either case, NFT responds with the NFT> prompt and accepts one command per line until you press CTRL/Z or type EXIT in response to the NFT> prompt:

```
NFT> command-1
NFT> command-2
.
.
.
NFT> command-n
NFT> ^Z
```

If your system manager has installed NFT as a CCL command and you want to use a single NFT command, you can type NFT and the NFT command on one line:

```
NFT command
```

NFT executes the command and returns control to the RSTS/E monitor.

Note

If you are running under the DCL Run-Time System, you can duplicate the NFT functions by issuing the appropriate DCL commands (for example, COPY, DELETE, PRINT, and so forth). Because the DCL command syntax differs significantly from the NFT command syntax (see Table 4-1), check the RSTS/E Quick Reference Guide and the RSTS/E System User's Guide for details about using DCL commands.

NFT: Network File Transfer Utility

To protect files from unauthorized use or destruction, you must supply log-in information for any node referred to in an NFT command. There are four methods for supplying this information:

- o You can use the NODESPECIFICATION command.
- o You can supply the information as an extension of the remote node specification.
- o You can use the /IDENTIFY switch to request that NFT prompt for the information on a one-time basis.
- o You can do nothing and force NFT to prompt for the information. NFT prompts you if you refer to a remote node in a command and you have not specified the log-in information in any other way.

In two of these cases (NODESPECIFICATION command and NFT prompting), once you specify log-in information for a particular node, NFT uses that information throughout your current NFT session, each time you reference that node. NFT goes on using the log-in information unless you change the information, it is overridden, or the remote node rejects it.

In the other two cases (node name extension and the /IDENTIFY switch), NFT uses the information you specify to override the existing log-in information for one command only.

No matter how you specify the log-in information, NFT uses the information to gain access to the remote file system before executing any command affecting a file at the remote node. If the information you specify is valid, you can access any files that you could normally access if you were directly logged in to the remote node using the same log-in information.

General Format of NFT Commands

A DECnet/E NFT command consists of a command keyword (defining an action) and variable parameters (defining the nodes and files involved in the action). In addition, you can use switches to further define the action. For example:

```
COPY DENVER::FILE.LST/SU-DALLAS::MYFILE.TXT
```

COPY is the command keyword, indicating that you want NFT to copy the file MYFILE.TXT at node DALLAS, specified at the right of the equal sign, to node DENVER and name the file FILE.LST. The /SU switch indicates that if a file called FILE.LST already exists at node DENVER, you want NFT to supersede the file.

You can abbreviate command and switch keywords. In general, you need to specify only the first two characters of the keyword (for example, CO for COPY, DI for DIRECTORY, TY for TYPE). Some switches, however, require four characters to identify the switch (for example, /NOHE for /NOHEADING, /NOSU for /NOSUPERSEDE, /NODE for /NODELETE). See the "NFT Commands" and "NFT Switches" sections for more information.

Most of the arguments you specify identify nodes and file specifications, which can contain wildcards and quoted strings. The following sections describe these arguments.

Node Names

You can use DECnet/E NFT to manipulate files to, from, and between nodes. NFT also works with files at the local node, although it is more efficient to use RSTS/E utilities and DCL commands to manipulate files at the local node. See the RSTS/E Utilities Reference Manual and the RSTS/E Quick Reference Guide for more information.

This chapter uses the sample network shown in Figure 4-3 as an example. To illustrate file transfers between DECnet nodes having different operating systems: BOSTON and TRNTO are DECnet/E nodes, DALLAS is a DECnet/llm node (RSX-llm operating system), and KANSAS is a DECnet/RT node (RT-11 operating system).

NFT: Network File Transfer Utility

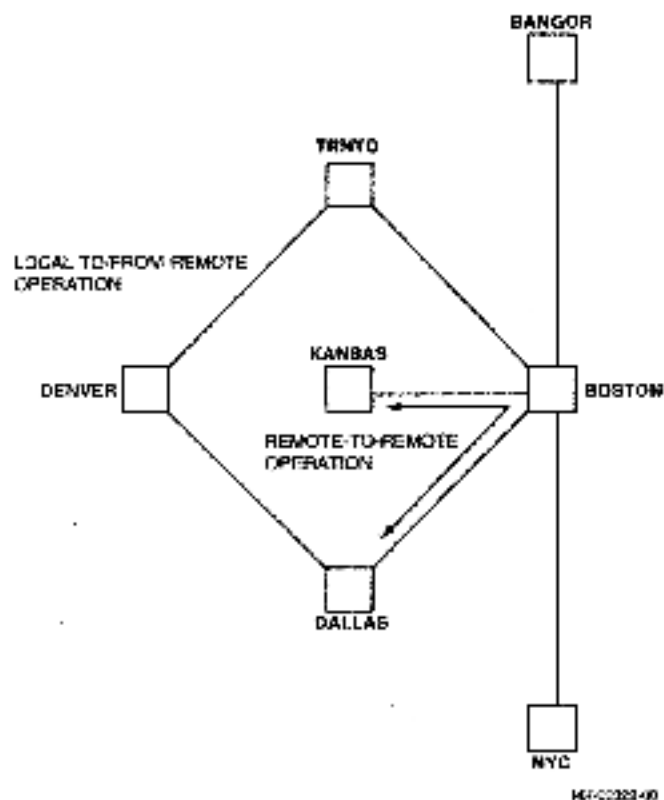


Figure 4-3: Sample Network Showing Transfers Between Nodes with DECnet/E NFT

To identify a remote node, specify a one- to six-character, alphanumeric node name. Use a double colon (::) to separate the node name from the file specification. You can determine the names of currently active nodes in the network by using the NCP command `SHOW ACTIVE NODES` (see the DECnet/E System Manager's Guide). If you specify the local node name in an NFT command, FAL is run at the local node.

As mentioned in the section "Running NFT," you can specify the log-in information required by NFT to access files at the remote node as an extension of the node name. Enter this information in the following format:

```
nodename"[user-id] [password] [sys-password]"::
```


where:

nodename is the name of the node at which you want to be validated as an established user. If you omit the node name, NFT lists the nodes you have already specified (if any), along with the log-in information you specified for the nodes.

If you specify a node name only, NFT prompts for log-in information.

user-id is the project-programmer number or PPN (sometimes called a user identification code or UIC). It can be up to sixteen characters long.

If the remote node permits alphanumeric named directories in addition to or in place of numeric PPNs, you can specify the user-id in that form.

password is the 6- to 14-alphanumeric character string that is your password.

sys-password is the system password for the remote node. It can be up to 16 characters long.

" " are double quotation marks that must enclose the log-in information.

NFT removes the quotation marks and passes the characters they enclose directly to the remote node without examination. In some instances, this means that the remote node processes the log-in information differently than if you specify the information using the NODESPECIFICATION command or through prompting. For example, consider the case where the remote node is running the VMS operating system. Because VMS does not permit non-octal user identification codes, NFT normally converts the specified user-id to the VMS-compatible octal equivalent. However, if you include the log-in information with the node name enclosed in quotation marks, NFT does not perform the conversion and errors may result.

You must separate the user-id, the password, and the sys-password by either a space or a tab.

File Specifications

The way you identify files depends on the operating system at the node where the file is located. For file specification formats used with

NFT: Network File Transfer Utility

non-RSTS/E nodes, see the documentation for the appropriate operating system.

For files at RSTS/E nodes, the file specification is a subset of the full RSTS/E file specification (see the RSTS/E System User's Guide). The file specification used is:

dev:[PPN]filename.typ

where:

dev: is the device designation. If you omit the device, NFT assumes the system disk (SY:).

Although you can refer to specific devices at the local or remote nodes, you are responsible for ensuring that someone properly mounts and positions magnetic tapes and performs other necessary functions. If a device is busy, NFT returns an error message (see Table 4-1). You can use the DCL ASSIGN and DEASSIGN commands at the local node to reserve local devices.

Note

Use NFT to move files between disk structures (either public or normally mounted private disks) at each node. If you want to transfer files from one type of device to another type (from a disk on one node to a magnetic tape on your local node for example), first use NFT to transfer the file to the local node and then use PIP to transfer the file to magnetic tape (see the RSTS/E Utilities Reference Manual.).

[PPN] is the project-programmer number identifying the directory where the file is located ({100,201} for example). If you specify a PPN, you must include the square brackets (or parentheses).

Some DIGITAL systems refer to the PPN as a user identification code, or UIC. If you omit the PPN, NFT uses the user-id you specified during the NFT log-in sequence.

Access to another PPN depends on your privilege at the remote node.

Some DIGITAL systems allow users to assign alphanumeric directory names that are equivalent to the PPN. If the remote system permits such names, you can use them in the file specification.

filename.typ is the file name and type. For files on RSTS/E nodes, the filename can be up to six alphanumeric characters, and the type can be up to three alphanumeric characters. If a command accepts both an input and an output file specification and you omit the output specification, NFT uses the input file specification. If you omit the file type, NFT does not supply a default.

Wildcards

NFT permits wildcard characters in file specifications. You can use wildcards to select a set of files for sequential file retrieval, file deletion, file spooling, directory listings, or command file execution. The format of the wildcard syntax depends on the capabilities of the remote node to process the file specification.

In an input file specification, any valid wildcard that is acceptable to the remote or local system is acceptable. For example, if the remote system is a RSTS/E system, an input file specification of A?????.LST selects as input all those files with an type of .LST and a file name beginning with the letter A. You can enter the PPN in an input specification as either [*], for named directories, or {*,*}, for numeric project-programmer numbers.

The only valid wildcard character you can use in an output file specification, however, is the asterisk (*). You can use an asterisk to replace one entity in the file specification. Thus, A?????.LST in the previous example is invalid as an output file specification but *.LST is valid.

When you use wildcards in the output file specification of certain commands, NFT duplicates file names on a one-to-one basis during the operation depending on the wildcards you use. For example, consider the following COPY commands:

```
COPY *.DAT=DALLAS::[COWBOY]*.LST
COPY *.DAT=DENVER::[1,34]FILEA.LST,[3,12]FILEB.LST
```

On receiving the first command, NFT copies every file from the directory COWBOY on node DALLAS with the type .LST to a file on the local node in the current directory with the same file name but with a type of .DAT.

The second command causes NFT to copy FILEA.LST from directory [1,34] on node DENVER to FILEA.DAT in the current directory on the local node and FILEB.LST from directory [3,12] on node DENVER to FILEB.DAT, also in the current directory on the local node.

In most cases where the input file specification contains wildcards

NFT: Network File Transfer Utility

but the output file specification does not, NFT concatenates the input files into one output file. For example, consider the previous example, slightly modified:

```
COPY MASTER.DAT=DALLAS::[COWBOY]*.LST
```

This command causes NFT to copy every file with the type .LST from directory COWBOY on node DALLAS to a single file MASTER.DAT in the current directory on the local node. In sequence, NFT appends each file located with the type .LST into a single output file named MASTER.DAT.

When you use wildcard characters in input file specifications as in the previous example, you must be aware of the attributes of the individual input files that NFT selects. For example, concatenating an executable file and an ASCII file can cause unpredictable results, depending on how you use the output file. NFT may or may not return an error message when you concatenate files having different attributes.

If you know you are going to have this problem, you can include the /INQUIRY switch in the command line. As NFT locates each input file, NFT prompts you and gives you the option of copying or not copying that file.

Quoted Strings

You can use quoted characters in commands to tell the local NFT not to examine the characters enclosed by the quotes. In general, when NFT finds paired double quotes ("text") or single quotes ('text'), it removes the quotes and passes the characters they enclose to the remote system without examination. This is useful when you must include characters in a file specification for a remote, non-RSTS system that the local NFT would recognize as part of a command.

For example, suppose you want to copy a file to a DECnet/RT node and you want to use the RT-11 file system's switch for allocating a specific amount of storage space for the file. This switch has the format /B:n. To prevent the local NFT from interpreting the slash and trying to process the switch as one of its own, you can enclose the file specification in quotes as follows:

```
COPY RTNODE::"FILE.LST/B:200"=LOCAL.L
```

The local RSTS/E NFT also treats pairs of square brackets [], parentheses (), and angle brackets <> in a similar manner. RSTS/E NFT does not remove these pairs. However, it passes them on, along with the characters they enclose, without examination. You can use these characters for access to either the local or a remote node. At any DECnet/E node, NFT passes the enclosed characters directly to RMS.

NFT Commands

Table 4-1 lists the eleven NFT commands available. The sections following discuss these commands.

Table 4-1: NFT Commands

Command	Meaning
AP[PEND]	Append file(s) from one node to a file on another.
CO[PY]	Copy file(s) from one node to another.
DE[LETE]	Delete file(s) at a remote node.
DI[RECTORY]	Displays a directory listing of files at a remote node.
EXI[T]	Exit from NFT.
HE[LP]	Obtain information on available commands and switches.
NOD[ESPECIFICATION]	Enter or change remote node log-in information.
PR[INT]	Spool file(s) from one node to the line printer of another node.
SU[BMIT]	Submit file(s) from one node for execution on another node.
TY[PE]	Display file(s) on the terminal console.
VE[RSION]	Display the local NFT or remote FAL version number.

Note

You cannot use NFT to access a file that another user has open for read/write access. NFT shares read access only (not read/write access) and returns the error message ?File is protected, skipping -- (see Table 4-3).

NFT: Network File Transfer Utility

APPEND

AP[PEND] Command

Use the APPEND command to transfer one or more files from one node to another, appending them to the end of an existing file at the target node.

Note

When you use the APPEND command, note the attributes of the files you want to append and make sure that the files are compatible. For example, appending a variable-length record file to one with fixed-length records can have unpredictable results. You may or may not receive an error message.

Format

```
AP[PEND] [dstnode::]outfile=[srcnode::]file[,...file]
```

Command Parameters

dstnode::

Specifies the name of the destination node containing the file to which NFT appends the other files. If you omit the destination node name, NFT assumes the local node. You must include two colons after the destination node.

outfile

Specifies the file specification of an existing file to which NFT appends the other files.

· Separates the destination and source specifications.

srcnode::

Specifies the name of the source node from which NFT copies the other files. If you omit the source node name, NFT assumes the local node.

file

Specifies the file specifications for the files NFT transfers and appends. NFT appends the files, in the order specified, to the end of outfile. For example:

NFT> APPEND DALLAS::ACCT.BAS=AACT.BAS,BACCT.BAS

NFT copies files AACCT.BAS and BACCT.BAS from the local node and appends them to the file ACCT.BAS at node DALLAS (see Figure 4-4).

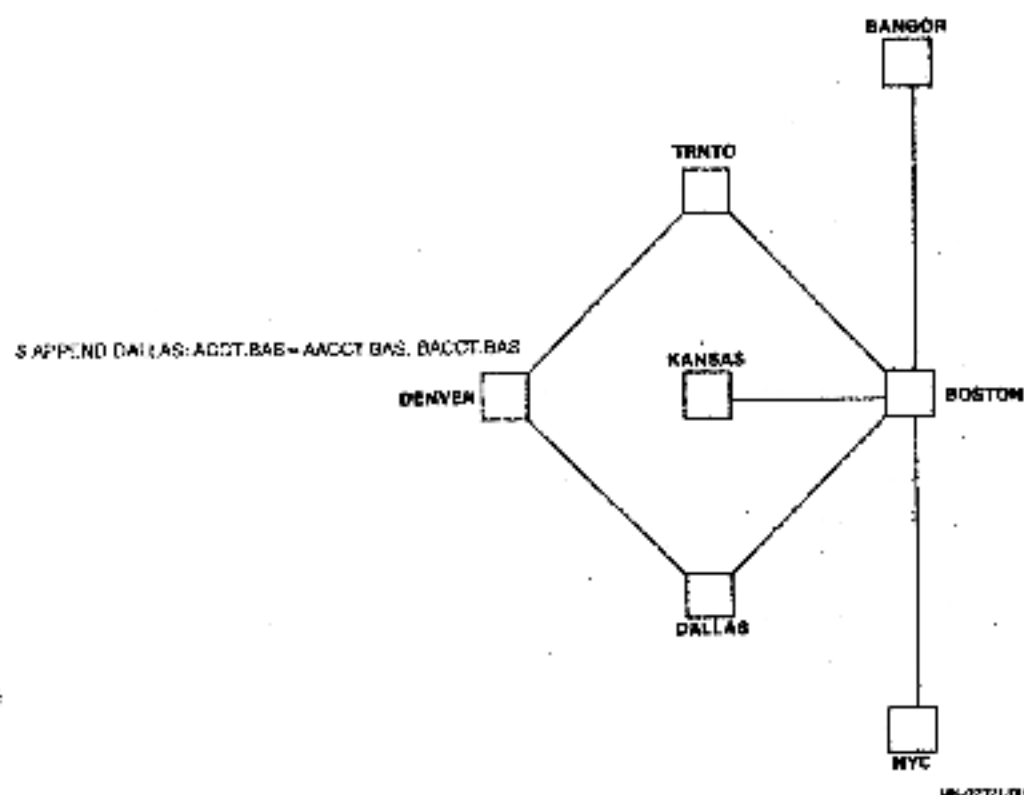


Figure 4-4: Example of the NFT APPEND Command

NFT: Network File Transfer Utility

COPY

CO[PY] Command

Use the COPY command to copy one or more files from one node to one or more files at another node. NFT does not delete the files at the source node. COPY is the default command for NFT; you can omit the command keyword COPY entirely.

Format

```
[CO[PY]] [dstnode::][outfile][=][srcnode::]file[,...file]
```

Command Parameters

dstnode::

Specifies the node name of the destination node to which NFT transfers the file(s). If you omit the destination node name, NFT assumes the local node.

outfile

Specifies the file specification NFT gives the transferred file(s) at the destination node. If you do not use wildcards, NFT concatenates the input files, in the order specified, to outfile. If you use wildcards, each output file NFT creates at the destination node has the outfile specification you specified with the wildcards in the input file names.

NFT creates the outfile using RMS, thus the outfile always has RMS file attributes. You can use PIP to remove the file attributes (see the RSTS/E Utilities Reference Manual).

Separates the destination and source specifications. You need to include the equal sign (=) only if you include a dstnode or outfile specification in the command. If you omit all three items, NFT copies the input files to your terminal. If you omit dstnode and outfile but you include the equal sign (=), NFT assumes *.* for outfile.

srcnode::

Specifies the node name of the source node from which NFT copies the file(s). If you omit the source node name, NFT assumes the local node.

file

Specifies the file specifications of the files NFT copies from the source node. For example:

```
NFT> AFIL.BAS=TRNTO::AFIL.BAS
```

```
NFT> BFIL.BAS=TRNTO::BFIL.BAS
```

```
NFT> CO DK1:MYFILE.BAS=TRNTO::AFIL.BAS,BFIL.BAS
```

```
NFT> COPY DALLAS::NEWFIL.BAS=TRNTO::AFIL.BAS,BFIL.BAS
```

In all these examples, NFT copies files AFIL.BAS and BFIL.BAS from the node named TRNTO (see Figure 4-5). In the first two commands, NFT copies the files to the local node's system disk as separate files named AFIL.BAS and BFIL.BAS, the same names they had at node TRNTO. In the third command, NFT copies and concatenates the files to form a file named MYFILE.BAS on disk DK1 at the local node. In the fourth command, NFT copies the files to form a file named NEWFIL.BAS on the public disk at node DALLAS.

NFT: Network File Transfer Utility
COPY

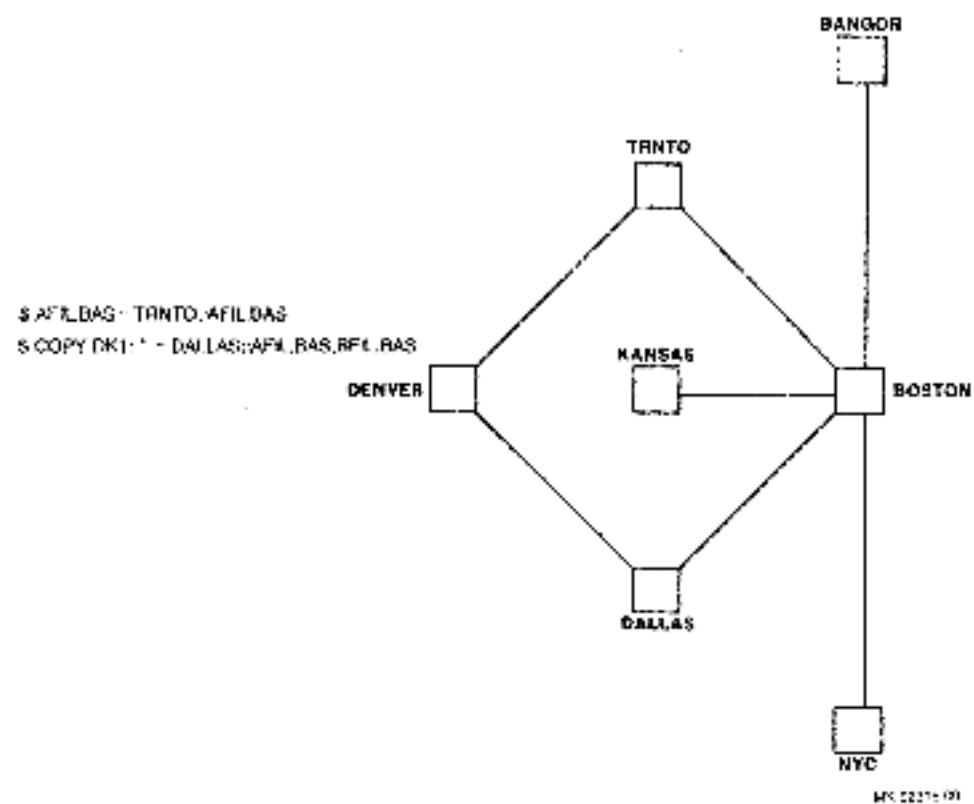


Figure 4-5: Example of the NFT COPY Command

DE[LETE] Command

Use the DELETE command to delete the specified disk files on remote nodes or the local one. You cannot delete tape files.

Format

DE[LETE] {dstnode::}file[,...file]

Command Parameters

dstnode::

Specifies the node name of the remote node at which NFT deletes the files. If you omit the node name, NFT assumes the local node.

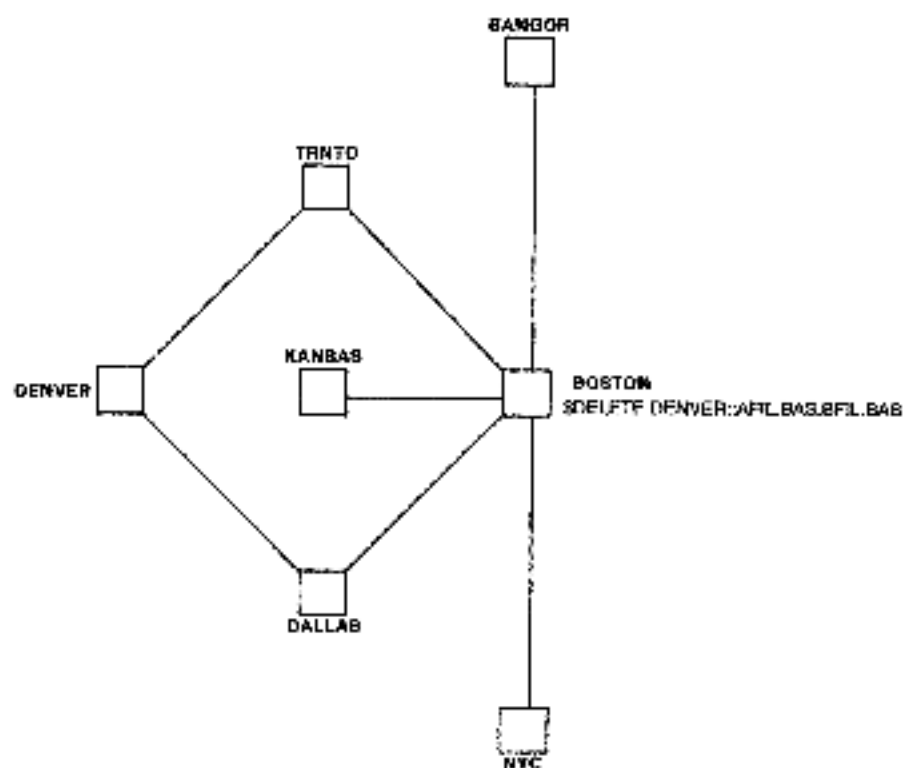
file

Specifies the file specifications for the disk files NFT deletes. You cannot delete tape files. For example:

NFT> DELETE DENVER::FIL.RNO,BINFIL.SAV,RM.BAS

NFT deletes files FIL.RNO, BINFIL.SAV, and RM.BAS from the public disk at node DENVER (see Figure 4-6).

NFT: Network File Transfer Utility
DELETE



MS 02207-00

Figure 4-6: Example of the NFT DELETE Command

DI[RECTOR] Command

Use the DIRECTORY command to display a directory listing of files located at a remote node.

Format

```
DI[RECTOR] [dstnode::][outfile][=][srcnode::]file[,...file]
```

Command Parameters

dstnode::

Specifies the name of the node on which NFT creates the directory listing. If you omit the node name, NFT assumes the local node.

outfile

Specifies the file specification of the file to which NFT writes the directory listing. If you omit outfile, NFT sends the listing to the terminal at the specified node. If the file currently exists, NFT overwrites its contents, unless you use the /APPEND switch.

Separates the destination and source specifications. Use the equal sign (=) only if you include a destination node or outfile specification in the command. If you omit the destination node, the outfile, and the equal sign (=), NFT sends the listing to your terminal.

srcnode::

Specifies the name of the node where the files you want to list are located.

file

Specifies the file specifications of the files on the source node you want to list. For example:

```
DIRECT KANSAS::[7,214)*.*
```

NFT displays a complete listing of all files in account [7,214] on the system disk of node KANSAS on the local terminal (see Figure 4-7).

NFT: Network File Transfer Utility
DIRECTORY

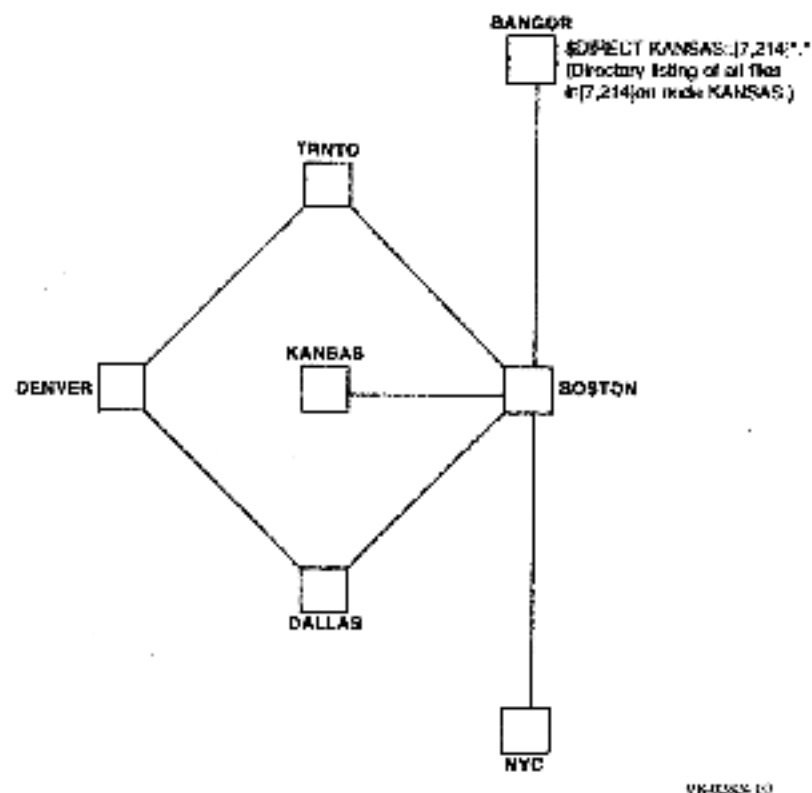


Figure 4-7: Example of the NFT DIRECTORY Command

EXI[T] Command

Use the EXIT command to terminate execution of NFT. It is equivalent to pressing CTRL/Z.

Format

EXI[T]

NFT: Network File Transfer Utility
HELP

HE[LP] Command

Use the HELP command to display information about the commands and switches available with NFT. The HELP command does not involve network access.

Format

HE[LP] [qualifier]

Command Qualifiers

COMMANDS

SWITCHES

Command Qualifiers

COMMANDS

Specifies that you want to display the help text for all NFT commands. If you do not include any qualifier, NFT displays the help text for both commands and switches.

SWITCHES

Specifies that you want to display the help text for all NFT switches.

If you do not include any qualifier, NFT displays the help text for both commands and switches.

NOD[ESPECIFICATION] Command

Use the NODESPECIFICATION command to enter or change the log-in information required to establish you as a valid user at a remote node. NFT keeps this information for use in later references to the node during your current NFT session.

You do not have to use this command to define initial log-in values. However, NFT prompts for log-in information each time you refer to a node for which you have not yet given any log-in information. NFT keeps this log-in information during your current NFT session or until you change it.

You can alter log-in information on a one-time basis for a single command by using either the node name extension or the /IDENTIFY switch.

Format

NOD[ESPECIFICATION] {nodename::}

Command Parameters

nodename::

Specifies the name of the node where you want the system to validate you as an established user. If you omit the node name, NFT lists the nodes you have already specified, if any, along with the log-in information you gave for the nodes.

If you specify a node name, NFT prompts for log-in information, after first displaying the node you specified:

```

      Node:  nodename
      User:  [PPN]
      Password:  password
System Password:  sys-password
  
```

where:

nodename is the name of the node you specified.

[PPN] is the project-programmer number (sometimes called a user identification code or UIC) used for logging in at the remote node. It can be up to 16 characters long.

NFT: Network File Transfer Utility
NODESPECIFICATION

If the remote node allows alphanumeric named directories in addition to or in place of numeric PPNs, you can specify the PPN in that form.

password is a 6- to 14-alphanumeric character string that is your password.

sys-password is the 6- to 14-character system password for the remote node. If the remote node does not require a system password, press RETURN. For example:

```
NFT> DIR BOSTON::  
  
Node: BOSTON  
User: 4,120  
Password:  
System Password: <RET>  
NFT>
```

After accepting the log-in information, NFT is ready to accept another command.

PR[INT] Command

Use the PRINT command to copy one or more disk files at one node to one or more temporary files at another node for input to the line printer. After printing, NFT deletes the temporary file from the destination node, unless you use the /NODELETE switch.

If the Print/Batch Services package (PBS) is active on the destination RSTS/E node, the PRINT command sends the print job to the NET\$PRINT queue, rather than sending it to the LPO queue or the default queue. If PBS is not active, the PRINT command sends the job to the CPSEB-based print spooler.

Format

```
PR[INT] [dstnode::][outfile][=][srcnode::]file[,...file]
```

Command Parameters

dstnode::

Specifies the name of the node to which NFT copies the files for printing. If you omit the node name, NFT assumes the local node.

outfile

Specifies the name of the temporary file at the destination node. If you do not use wildcards in the outfile specification, NFT concatenates the input files into a single output file before printing. If you do use wildcards, NFT creates multiple temporary files, depending on the wildcards you use.

If the destination node is a RSTS/E node, outfile must be a disk file.

=

Separates the destination and source specifications. Use the equal sign (=) only if you include a destination node or outfile specification in the command.

srcnode::

Specifies the name of the source node from which NFT copies the files. If you omit the source node name, NFT assumes the local node.

NFT: Network File Transfer Utility
PRINT

file

Specifies the file specifications of the disk files NFT copies from the source node. For example:

```
NFT> PRINT TRNTO::TEXT.DOC=BOSTON::BOOK?.DOC
```

NFT takes all files at node BOSTON that have file names beginning with BOOK and the .DOC type, and copies them to a temporary file named TEXT.DOC at node TRNTO (see Figure 4-8).

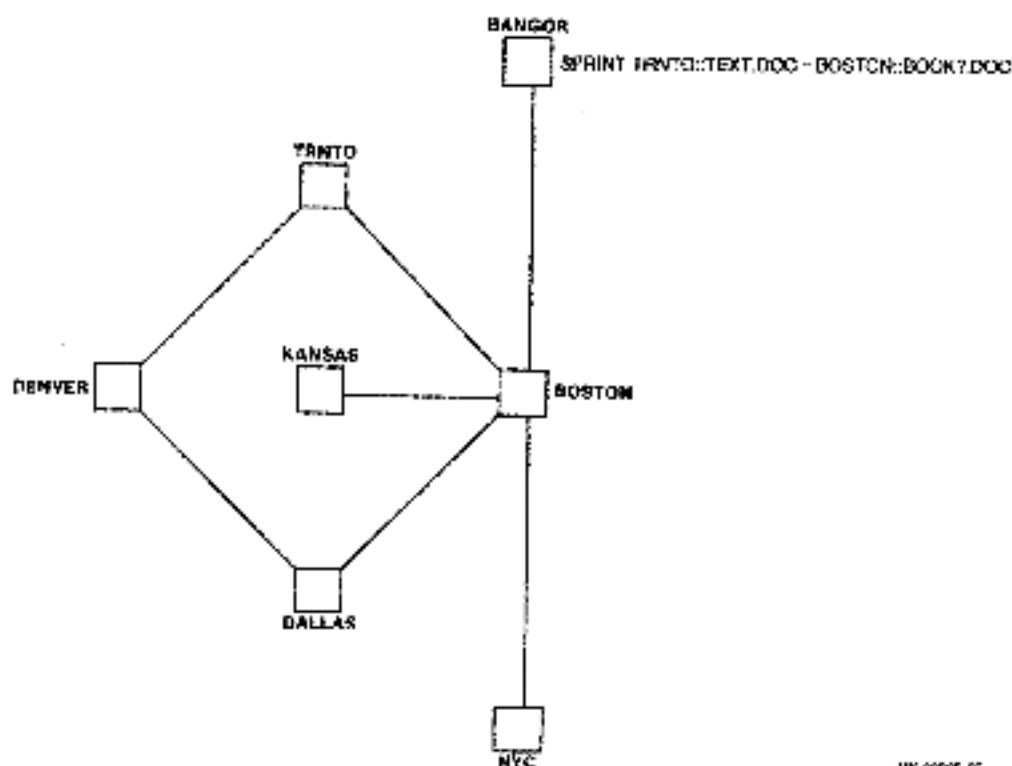


Figure 4-8: Example of the NFT PRINT Command

If you omit the destination node, the outfile, and the equal sign (=), NFT sends the input files to the printer on your local node. In this case, NFT does not delete the file after printing, unless you use the /DELETE switch. For example:

```
NFT> PRINT BOSTON::BOOK?.DOC
```

NFT takes all files at node BOSTON that have file names beginning with BOOK and prints them at node BOSTON. It does not transfer them to a temporary file and it does not delete them.

SU[BMIT] Command

Use the SUBMIT command to copy one or more files from one node to one or more temporary files at another node for execution. After execution, the system deletes the command file from the destination node unless you use the /NODELETE switch. The destination node must be able to execute command files and the files must be in the proper format for execution at that node. For RSTS/E nodes, the system submits the files to BATCH, the batch processor. See the RSTS/E Utilities Reference Manual for a discussion of the command file format for BATCH.

If the file type is .COM and the Print/Batch Services package (PBS) is active on the destination RSTS/E node, the SUBMIT command sends the file to the PBS batch server and to the NET\$BATCH queue. Otherwise, it sends the file to the OPSER-based batch processor.

Format

```
SU[BMIT] [dstnode::][outfile][=][srcnode::]file[,...file]
```

Command Parameters

dstnode::

Specifies the name of the destination node to which NFT copies the command files for execution and deletion. If you omit the node name, NFT assumes the local node.

outfile

Specifies the name of the temporary file NFT creates at the destination node. If you do not use wildcards in the file specification, NFT concatenates the files into a single output file. If you use wildcards, NFT creates multiple temporary files at the destination node, with file names depending on the wildcards you use.

Separates the destination and source specifications. Use the equal sign (=) only if you include a destination node or outfile specification in the command.

srcnode::

Specifies the name of the source node where the files are located.

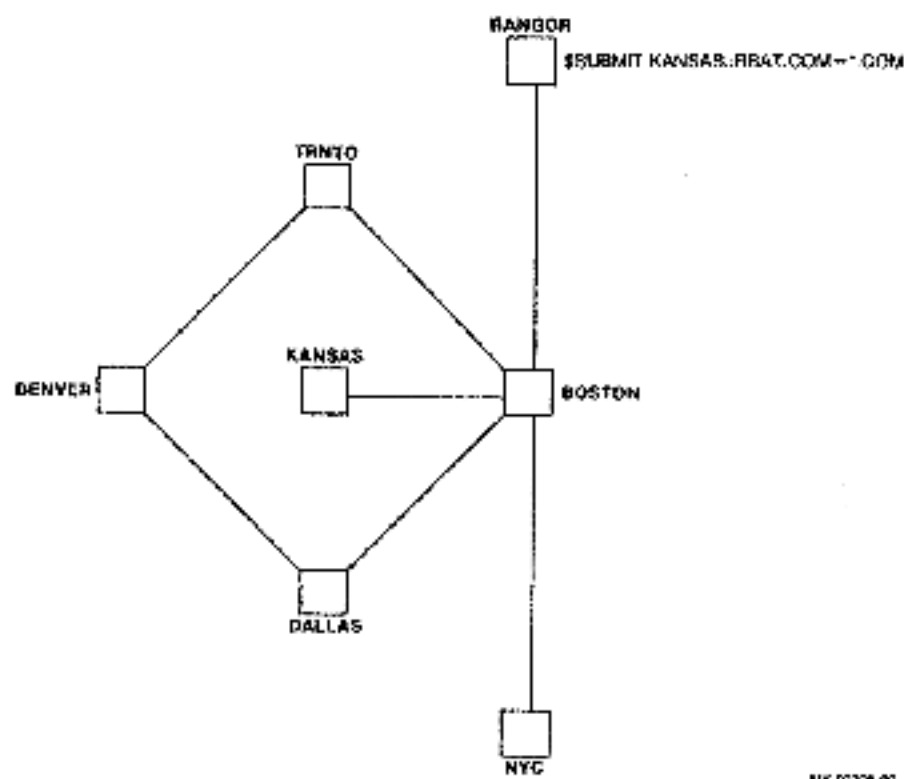
NFT: Network File Transfer Utility
SUBMIT

If you omit the source node name, NFT assumes the local node.
file

Specifies the file specifications of the input files. For example:

```
NFT> SUBMIT KANSAS::RBAT.CTL=*.CTL
```

NFT copies into a temporary file called RBAT.CTL at node KANSAS all files at the local node having a .CTL type. NFT then submits and deletes the files (see Figure 4-9).



NK 02322-01

Figure 4-9: Example of the NFT SUBMIT Command

If you omit the destination node, the outfile, and the equal sign (=), NFT submits the files to the batch processor on your source node. In this case, NFT does not delete the files after execution, unless you use the /DELETE switch. For example:

```
NFT> SUBMIT KANSAS::RBAT.CTL
```

NFT takes the file RBAT.CTL at the node KANSAS and submits it to the batch processor on KANSAS. It does not delete the file afterward.

TY[PE] Command

Use the TYPE command to display the input files from the source node on your terminal. The TYPE command is equivalent to the COPY command when you omit everything on the left side of (and including) the equal sign.

```
+-----+
|      Format      |
|                  |
| TY[PE] [srcnode:]file[,...file] |
|                  |
+-----+
```

Command Parameters

srcnode::

Specifies the name of the source node from which NFT retrieves the file(s) it displays. If you omit the source node name, NFT assumes the local node.

file

Specifies the file specifications of the input files. For example:

```
NFT> TY DENVER::DAILY.SUM
```

NFT retrieves the file DAILY.SUM from the public disk on remote node DENVER and displays them on your terminal. See Figure 4-10.

NFT: Network File Transfer Utility
TYPE

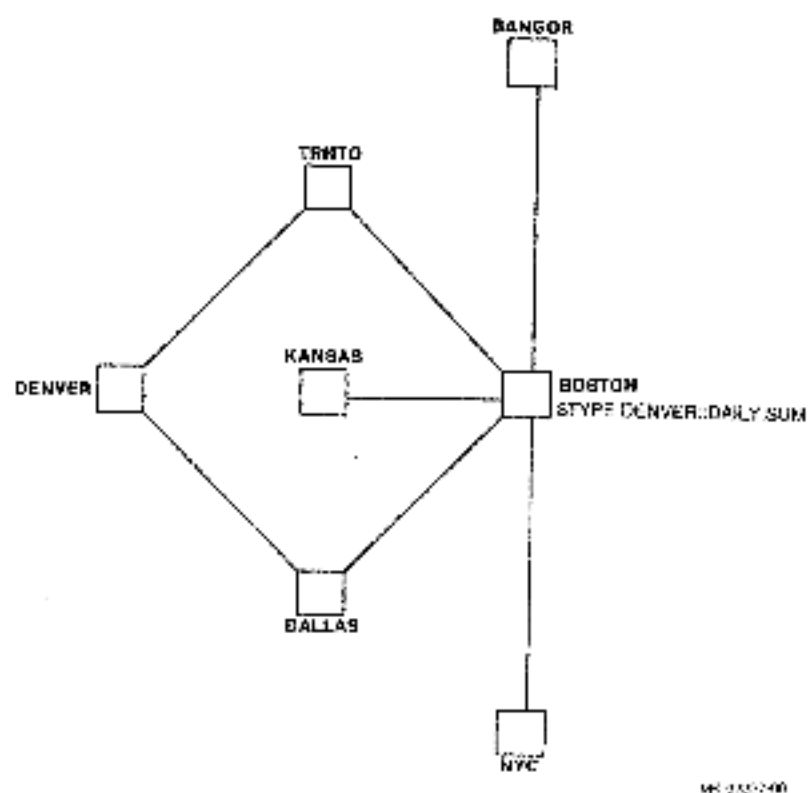


Figure 4-10: Example of the NFT TYPE Command

VE[RSION] Command

Use the VERSION command to display the version numbers of the local DECnet/E NFT program or the remote FAL program, as well as the version number of the Data Access Protocol (DAP) being used.

Format

VE[RSION] [srcnode::]

Command Parameters

srcnode::

Specifies the name of the node at which the version number of FAL is to be examined. If you omit the node name, NFT displays the version numbers of NFT and DAP on the local node.

NFT: Network File Transfer Utility

NFT Switches

Table 4-2 lists the switches you can use in NFT commands. Most of these switches apply only to certain commands; some can apply only to the source or destination file specification. In general, you append a switch to a file specification and it applies to that file only. You can use more than one switch as long as they are mutually compatible. Further description follows Table 4-2.

Table 4-2: NFT Switches

Switch	Meaning	Applicable NFT Commands	Page
/AP[PEND]	Append input file(s) to output file.	COPY, DIRECTORY	4-36
/AS[CII]	Data being transferred is ASCII 7-bit data.	APPEND, COPY, PRINT, SUBMIT, TYPE	4-36
/BL[OCK]	Transfer image data in 512-byte blocks.	APPEND, COPY	4-36
/BR[IEF]	Generate an abbreviated directory listing.	DIRECTORY	4-37
/CO[NFIGUOUS] or /CTG	Allocate space contiguously.	COPY, PRINT, SUBMIT	4-38
/DE[LETE]	Special-case delete for SUBMIT and PRINT commands.	All	4-38
/FU[LL]	Generate an extended directory listing.	DIRECTORY	4-38
/ID[ENTIFY]	Have NFT prompt for log-in information.	All	4-38
/IM[AGE]	Data being transferred is 8-bit data.	APPEND, COPY	4-39
/IN[QUIRY]	Verify input file before function execution.	APPEND, COPY, DELETE, PRINT, SUBMIT, TYPE	4-39
/LI[ST]	Generate a normal directory listing.	DIRECTORY	4-40

Table 4-2: NFT Switches (Cont.)

Switch	Meaning	Applicable NFT Commands	Page
/LO[G]	Output file name after function execution.	APPEND, COPY, DELETE, PRINT, SUBMIT, TYPE	4-40
/MO[RE] or -	Continue command input on next line.	All	4-40
/NA[TIVE]	Reformat file to stream ASCII.	APPEND, COPY, PRINT, SUBMIT, TYPE	4-41
/NOA[TTIBUTES]	Ignore RMS attributes.	COPY	4-42
/NOCO[N]TIGUOUS]	Allocate space noncontiguously.	COPY, PRINT, SUBMIT	4-42
/NOD[ELETE]	Do not delete temporary files.	PRINT, SUBMIT	4-42
/NOH[EADING]	Generate directory listing with no header.	DIRECTORY	4-42
/NOSU[PERSEDE]	Do not supersede an existing file.	COPY, PRINT, SUBMIT	4-43
/POS	Position magnetic tape to end of volume.	COPY	4-43
/RWC	Rewind magnetic tape on close.	COPY	4-43
/RWO	Rewind magnetic tape on open.	COPY	4-43
/SU[PERSEDE]	Supersede an existing file.	COPY, PRINT, SUBMIT	4-44
/TO[TAL]	Output directory size only.	DIRECTORY	4-44
/VA[R]IABLE]	Reformat file to variable format.	APPEND, COPY, PRINT, SUBMIT	4-44

NFT: Network File Transfer Utility

/AP[PEND] Switch

Use the /APPEND switch with the destination file specification in the COPY and DIRECTORY commands. When you use the /APPEND switch with the COPY command, it is the same as using the APPEND command by itself: NFT appends the input files to the output file.

Note

When you use the /APPEND switch with the COPY command, note the attributes of the files to be appended and make sure that the files are compatible. For example, appending a variable-length record file to one with fixed-length records can have unpredictable results. You may or may not receive an error message.

When you use the /APPEND switch with the DIRECTORY command and an output file already exists, NFT appends the new directory listing to the output file. For example:

```
NFT> COPY TEXT.DOC/APP=DALLAS::BOOK77.DOC
NFT> DIRECTORY DIRECT.LST/AP=[7,214]*.BAS
```

/ASCII Switch

Use the /ASCII switch to indicate that a transmitted file contains 7-bit ASCII data. You can use the /ASCII switch with the source file specification of the APPEND, COPY, PRINT, SUBMIT, or TYPE commands. For example:

```
NFT> COPY NYC::MEMO.DOC=[7,214]NOTES.TXT/ASC
```

/BL[OCK] Switch

Use the /BLOCK switch when transferring image (binary) files using the APPEND or COPY commands. You can use the /BLOCK switch with the source file specification so NFT transfers the files in 512-byte blocks.

In general, the /BLOCK switch only applies to systems that fully support binary block transfers.

You must use the /BLOCK switch when transferring non-ASCII files that do not have recorded attributes. On RSTS/E systems, this includes:

- o Files compiled by BASIC-PLUS (.BAC file type)
- o Record-I/O files
- o Virtual array data files
- o Binary files (.OBJ and .SAV file types)

On RSTS/E systems this includes:

- o Files generated by the MACRO assembler
- o Compiled FORTRAN IV files
- o Compiled DIBOL-11 files
- o Files generated by the RT-11 librarian
- o Files generated by the RT-11 linker

You cannot use the /BLOCK switch to transfer ASCII stream files or files that contain valid RMS attributes (for example, object modules created using MAC.TSK). Using the /BLOCK switch with these files results in a BLOCK image copy of the file, with the original attributes being changed to RMS attributes for a stream file of zero record length and fully used blocks. For example:

```
NFT> COPY STORBN.BAC=BOSTON::COMP.BAC/BL
```

Note

Do not use the /BLOCK switch with either the /NATIVE switch or the /VARIABLE switch.

/BR[BRIEF] Switch

Use the /BRIEF switch at the end of the DIRECTORY command to display a brief directory listing including device, directory, file name, and file type. This listing is similar to the listing the RSTS/E PIP program generates. For example:

```
NFT> DIR QUICK.DIR=NYC::[7,214]*.DOC/BR
```

NFT: Network File Transfer Utility

/CO[NTIGUOUS] or /CTG Switch

Use the /CONTIGUOUS switch with the destination file specification in the COPY, PRINT, or SUBMIT commands to allocate contiguous space for the destination file(s). If NFT cannot allocate contiguous space, the command fails. For example:

```
NFT> COPY DALLAS::FILE1.TSK/CO=MYFILE.TSK
```

In this example, NFT allocates contiguous space at node DALLAS (an RSX-11M node) for the binary file FILE1.TSK, consisting of the file MYFILE.TSK copied from the local node.

/DE[LETE] Switch

Use the /DELETE switch with any file specification in any command to delete that file after the command is done.

The primary use of this switch applies to the source file specification of the PRINT and SUBMIT commands when you do not supply a destination file specification -- that is, when the files to be printed or submitted to the batch processor already exist at the desired node. With this form of these two commands, the files are not automatically deleted since they are, in fact, the source files and not temporary destination files. Using the /DELETE switch in this case causes the files to be deleted at the conclusion of the operation. For example:

```
NFT> PRINT ROUGH.DRF/DELETE
```

/FU[LL] Switch

Use the /FULL switch at the end of the DIRECTORY command to display a directory listing that includes device, directory, file name, file type, file size, protection code, creation date and time, as well as a listing of symbolic attributes. This listing is similar to the full directory listing that the RSTS/E PIP program generates. For example:

```
NFT> DIR DALSYS.DIR=DALLAS::SY:[1,2]*.* /FULL
```

/ID[ENTIFY] Switch

Use the /IDENTIFY switch once with both the source and destination specifications of any command to cause NFT to prompt for log-in information. NFT uses the information you give in response to this prompting only for the current operation. NFT does not keep the

information for the remainder of the session. After the execution of a command with this switch, the log-in information goes back to what was previously specified for the node, if any.

If you use the /IDENTIFY switch with both the source and destination specifications, NFT prompts for information for the source node first. NFT uses a second set of prompts to obtain information concerning the destination node. For example:

```
NFT> COPY BOSTON::FILE1.TXT/ID=DALLAS::FILE3.TXT,FILE4.TXT/ID
```

```
      Node:    DALLAS
      User:    120,150
      Password:
System Password:
```

```
      Node:    BOSTON
      User:    30,117
      Password:
System Password:
NFT>
```

NFT uses the log-in information to copy the files and then deletes the log-in information from its internal records. Previously specified log-in information for BOSTON and DALLAS, if any, is restored for use in later commands.

/IM[AGE] Switch

Use the /IMAGE switch to indicate that a transmitted file contains 8-bit image data. You can use the /IMAGE switch with the source file specification of the APPEND and COPY commands. For example:

```
NFT> APPEND DALLAS::TOTALS.DAT=BOSTON::[1,34]DAILY.SUM/IMAGE
```

/IN[QUIRY] Switch

Use the /INQUIRY switch with the source file specification of the APPEND, COPY, DELETE, PRINT, SUBMIT, and TYPE commands. Use the /INQUIRY switch to request NFT to display the name of each input file for verification before it opens the file and performs the indicated function. After NFT displays the name, you can respond to the prompt with either YES or NO to indicate whether or not NFT performs the operation on that particular file. If you respond with NO, NFT skips the file and displays the next file name.

The /INQUIRY switch is useful when you do not use wildcards in the destination file specification and you want to concatenate the source

NFT: Network File Transfer Utility

files into one file. In this circumstance, if the source files represented by the wildcard specification have incompatible attributes, unpredictable results can occur, depending on how you use the final destination file. NFT may or may not generate an error message.

By using the /INQUIRY switch, you can make the decision, on a file-by-file basis, as to whether you want to include the file in the operation. For example:

```
NFT> DELETE NYC::[7,214]*.DOC/INQ
```

/LI[ST] Switch

Use the /LIST switch at the end of the DIRECTORY command to display a directory listing that includes: device, directory, file name, file type, file size, protection code, and creation date and time. This listing is similar to the directory listing that the RSTS/E PIP program generates.

If you do not use any switch with the DIRECTORY command, /LIST is the default. For example:

```
NFT> DIRECT DENVER::[100,57]*.*/LIST
```

/LO[G] Switch

Use the /LOG switch with the source file specification in the APPEND, COPY, DELETE, PRINT, SUBMIT, and TYPE commands to request that NFT log the name of each input file on your terminal after NFT performs the indicated function. The /LOG switch is useful when you use wildcards in the input file specification. For example:

```
NFT> COPY BOSTON::[1,230]*.*=DENVER::[3,45]*.MAC/LOG
```

NFT also supports the /NOLO[G] switch, which is the default, and therefore, is not actually required.

/MO[RE] or - Switch

Use the /MORE or the hyphen character (-) switch at the end of a terminal line to indicate that the command continues on the next line. NFT then prompts for continuation lines with MORE>. If you press CTRL/Z while typing a continuation line, NFT deletes the whole command

and displays the NFT> prompt. You can use the /MORE switch as many times as you want. For example:

```
NFT> DALLAS::DK0:[120,153]FILE.RNO=BOSTON::[30,205]DK1/MORE
MORE> :FILE.RNO,-
MORE> FILE3.RNO
NFT>
```

You cannot use the /MORE switch when you include a command on the same line as the NFT CCL command.

/NA[TIVE] Switch

Use the /NATIVE switch with the destination file specification in the APPEND, COPY, PRINT, SUBMIT, or TYPE commands when the destination node is a RSTS/E node. The /NATIVE switch converts files stored at the source node (in either of two ASCII formats) to stream ASCII files at the destination node. The two source formats are:

- o RMS-11 variable-length records with implied carriage control
- o Variable-length records with embedded carriage control characters

This conversion is useful when you transfer ASCII files from an RSX or VAX/VMS node that does not process stream ASCII to a RSTS/E node that does. For example:

```
NFT> COPY TEST.MAC/NA=DALLAS::FIL.MAC
```

You can also use the /NATIVE switch with the source file specification in the APPEND, COPY, PRINT, SUBMIT, or TYPE commands when the source node is either a TOPS-20 or an RT-11 node and you want to transfer the files as stream ASCII files.

Because RT-11 and TOPS-20 files do not contain file attributes, NFT at the source node has no way of knowing that the file is stream ASCII unless you use the /NATIVE switch. For example:

```
NFT> COPY PROGA.MAC=NYC::PROGA.MAC/NA
```

Note

You cannot use the /NATIVE switch with the /BLOCK switch.

NFT: Network File Transfer Utility

/NOATTRIBUTES] Switch

Use the /NOATTRIBUTES switch when you do a RSTS-to-RSTS copy and you want to ignore the RMS attributes. Use the /NOATTRIBUTES switch on the destination or source file specification, or both. However, if you use the /NOATTRIBUTES switch on the destination file specification, RMS creates the output file without RMS attributes. If you use the switch on the source file specification, RMS opens the source file but ignores the attributes if any exist for that file. For example:

```
NFT> COPY TEST.MAC=DALLAS::FIL.MAC/NOA
```

In this example, NFT creates TEST.MAC with RMS attributes. It also opens file FIL.MAC on node DALLAS and ignores the RMS attributes if the file has any.

/NOCONTIGUOUS] Switch

Use the /NOCONTIGUOUS switch with the destination file specification in the COPY, PRINT, or SUBMIT commands to allocate noncontiguous space for the destination file(s). For example:

```
NFT> COPY DALLAS::FILE1.TSK/NOCO=MYFILE.TSK
```

In this example, NFT allocates noncontiguous space at node DALLAS (an RSX-11M node) for the binary file FILE1.TSK, which consists of the file MYFILE.TSK.

/NODELETE] Switch

Use the /NODELETE switch with the destination file specification in the PRINT or SUBMIT commands to suppress deletion of the temporary file(s) created at the destination node. For example:

```
NFT> SUBMIT BOSTON::RBAT.CTL/NOD=*.CTL  
NFT> PRINT TEXT.DOC/NODELETE=DALLAS::BOOK??,DOC
```

/NOHEADING] Switch

Use the /NOHEADING switch at the end of the DIRECTORY command to obtain a listing that does not include the header line across the top. The directory listing includes a complete file specification for each

entry (device, FPN, number, file name and file type). The /NOHEADING switch is useful in batch command files. For example:

```
NFT> DIRECT LIBRY.CTL=DALLAS::[2,173]*.SAV/NOHEAD
```

/NOSUPERSEDE Switch

Use the /NOSUPERSEDE switch with the destination file specification in the COPY, SUBMIT, or PRINT commands. If a file already exists at the destination node with the same name as that given in the command, the command fails.

When NFT finds that the specified output file name already exists, it prompts you, asking whether you want to supersede the file. The /NOSUPERSEDE switch disables this automatic feature. This is useful in batch command files that cannot handle unexpected prompting. For example:

```
NFT> COPY FILE.LST/NOSU=BOSTON::FILE.TXT
```

/POS Switch

Use the /POS switch with the destination file specification in the COPY command when the device is a magnetic tape. Using the /POS switch causes NFT to position the file at the current end of volume before making the transfer. For example:

```
NFT> COPY MT2:HFFIL.BAC/POS=BOSTON::HPRTN.BAC
```

/RWC Switch

Use the /RWC switch with any file name in the COPY command when the device is magnetic tape. Using the /RWC switch causes NFT to rewind the tape after making the transfer. For example:

```
NFT> COPY MT1:RMAX.BAS/RWC=BOSTON::RMAX.BAS
```

/RWO Switch

Use the /RWO switch with any file name in the COPY command when the device is magnetic tape. Using the /RWO switch causes NFT to rewind the tape before making the transfer. For example:

```
NFT> COPY MT0:FILEX.RNO/RWO=BOSTON::TEXT.RNO
```

NFT: Network File Transfer Utility

/SU[PERSEDE] Switch

Use the /SUPERSEDE switch with the destination file specification in the COPY, SUBMIT, or PRINT commands. If a file already exists at the destination node with the same name you specify in the command, NFT replaces the existing file with the new file. For example:

```
NFT> COPY FILE.LST/SU~BOSTON::FILE.TXT
```

/TO[TAL] Switch

Use the /TOTAL switch at the end of the DIRECTORY command to display the total size of the directory. In this instance, NFT does not list the individual directory entries. For example:

```
NFT> DIR NYC::[1,2]*.* /TOTAL
```

/VA[R]IABLE Switch

Use the /VARIABLE switch with the destination file specification in the APPEND, COPY, PRINT, or SUBMIT commands to convert a file from stream ASCII at the source node to RMS-11 variable-length record format with implied carriage control. Because RSX and VMS nodes do not process stream ASCII files, you mainly use the /VARIABLE switch to transfer stream ASCII files from a DECnet/E or DECnet/RT node to an RSX or VAX/VMS node. For example:

```
NFT> COPY DALLAS::FIL.MAC/VA=TEST.MAC
```

In this example, NFT copies a MACRO file from a DECnet/E node to the RSX node.

Note

You cannot use the /VARIABLE switch with the /BLOCK switch.

NFT Error Messages

Table 4-3 lists the error messages that NFT prints when you make a mistake typing an NFT command (syntax errors) or when an error occurs during NFT initialization.

Table 4-3: NFT Error Messages

	Message and Meaning
?NFT -- ATTEMPTED CONTINUATION IS ILLEGAL	You cannot continue a one-line CCL command with the /MORE or - switch.
?NFT -- BREAK BUFFER OVERFLOW	There are too many quotes, commas, underscores, and equal signs in the command. The current limit on these special characters per command is 40.
?NFT -- FILE PROTECTED, SKIPPING	The file you attempted to access is open for read/write access by another user.
?NFT -- FILE SPECIFICATION TOO LONG	The file specification you typed is too long to fit in an internal buffer used by NFT.
?NFT -- GMCR\$ DIRECTIVE FAILURE	Tell your DIGITAL Software Support Specialist that the RSX-11M directive used to read the CCL command line has failed. Then retype the command, starting NFT with RUN \$NFT.
?NFT -- INVALID COMMAND FORMAT	Can be caused by one of several errors: - o Trying to type something other than a node name in a NODESPECIFICATION command. - o Typing a double colon (::) in a NODESPECIFICATION command without any node name. - o Using any switch other than /IDENTIFY or /INQUIRY in a DELETE or EXECUTE command. - o Not typing an equal sign (=) in an APPEND command. - o Typing an equal sign (=) in a DELETE, TYPE, or EXECUTE command. - o Typing anything but COMMANDS or SWITCHES after a HELP command. - o Not typing a node name in an EXECUTE command.
?NFT -- INVALID NETWORK NODE NAME	Valid node names are from one- to six characters that are uppercase, alphanumeric, and must contain at least one letter.
?NFT -- INVALID NODE NAME SPECIFICATION	Either the node name is not in the right place or it contains nonalphanumeric characters.

Table 4-3: NFT Error Messages (Cont.)

Message and Meaning	
?NFT -- MORE THAN ONE INPUT NODE SPECIFICATION	You cannot type more than one node name on the right side of an equal sign.
?NFT -- MORE THAN ONE OUTPUT NODE SPECIFICATION	You cannot type more than one node name on the left side of an equal sign.
?NFT -- NO NODE NAME GIVEN FOR /IDENTIFY	No node name was given for the /IDENTIFY switch to operate on.
?NFT -- RMS INITIALIZATION FAILURE	An error occurred in initializing RMS. If the error persists, call a software specialist.
?NFT -- TOO MANY /IDENTIFY SWITCHES	More than one /IDENTIFY switch was given on one side of an equal sign.
?NFT -- TOO MANY OUTPUT FILE SPECIFICATIONS	More than one file name was given on the left side of an equal sign.
?NFT -- TOO MANY = SIGNS	The command typed contains more than one equal sign (=).
?NFT -- UNMATCHED QUOTE CHARACTERS	There are an odd number of quote characters in the command line. Quote characters must be paired.
?NFT -- UNRECOGNIZABLE COMMAND	Check the spelling of the command keyword (see Table 4-1) and retype the command.
?NFT -- UNRECOGNIZABLE SWITCH	Check the spelling of the command switch (see Table 4-2) and retype the command.

Chapter 5

NETCPY: Network Copy Between Devices

The NETCPY utility allows you to copy all of the information on a DECtape, magnetic tape, flexible diskette, or disk from one DECnet/E node to a like device on another DECnet/E node. See Figure 5-1. One of the nodes must be the local node. NETCPY is a DECnet/E utility; there is no general DECnet utility to copy the contents of devices between nodes.

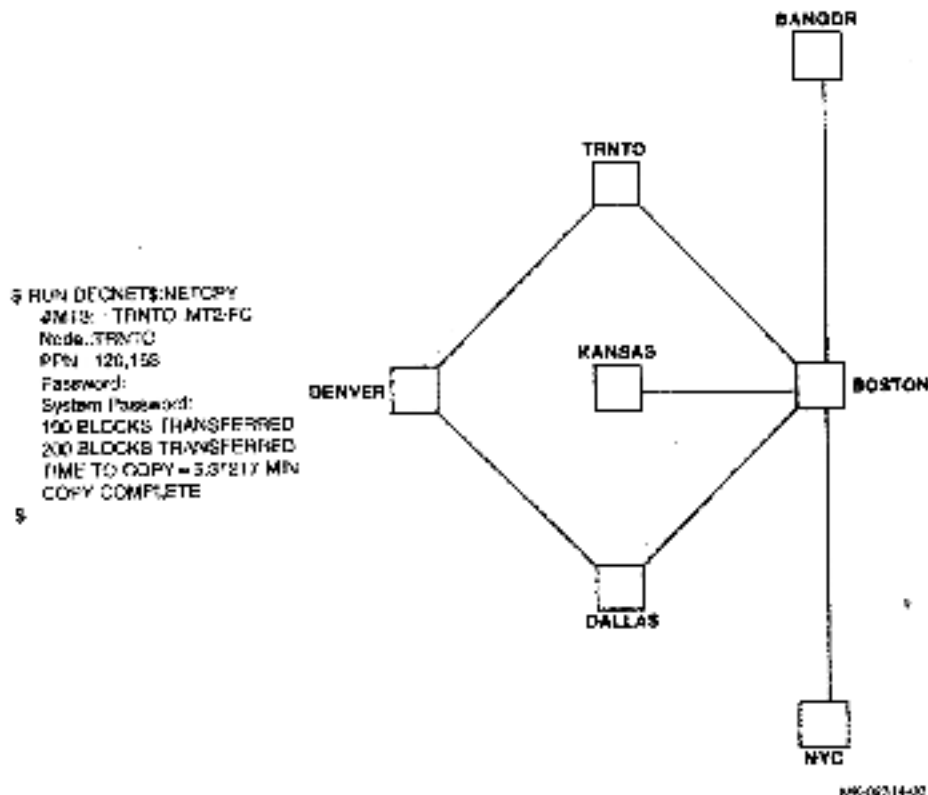


Figure 5-1: NETCPY Copies an Entire Device from One DECnet/E Node to Another DECnet/E Node

NETCPY: Network Copy Between Devices

To protect devices from unauthorized use, you must supply log-in information for the specified remote node. NETCPY prompts for the information and checks its validity before allowing the transfer to proceed. Then you can use the TLK utility to ask the operator at the remote DECnet/E node to mount tapes or do other preliminary preparations (see Chapter 2).

Running NETCPY

You can run NETCPY from any terminal in a DECnet/E system by typing:

```
$RUN DECNET$:NETCPY
NETCPY V4.0 RSIS V9.3 DECNET/E
+command
Node:
PPN:
Password:
System Password:
```

where:

+command:	is described in the following section.
PPN:	is the Project-Programmer Number you use to log in to the remote node.
Password:	is the 6- to 14-alphanumeric character password you use to log in to the remote node.
System Password:	is the system password (if required)

General Format of NETCPY Command

The general format of the NETCPY command is:

```
[dstnode::]dstdevice:[dstsw]=[srcnode::]srcdevice:[/FC][srcsw]
[/NC]
```

where:

dstnode::

is the one- to six-character node name of the destination node to which the contents of the device are to be copied. The node name must begin with a letter. If you omit the node name, NETCPY assumes your local node. If dstnode is not the local node, then srcnode must be the local node. Dstnode must be a DECnet/E node.

dstdevice:

is the destination device. Allowable devices are:

MTn: Magnetic tape unit n
 DXn: Flexible diskette unit n
 DKn: Disk unit n
 DTn: DECTape unit n

The destination device must be the same type as srcdevice. You can use NETCPY to copy larger disks but the speed of most transmission lines usually makes such operations impractical for remote transfers.

dstsw

is either or both of the optional /PA and /DE switches. See the following section.

=

separates the destination description from the source description.

srcnode:

is the node from which the information is to be copied. If you omit srcnode, NETCPY assumes your local node. If srcnode is not the local node, then dstnode must be the local node. Srcnode must be a DECnet/E node.

srcdevice

is the source device. You must specify the same device that you specified for dstdevice.

/PC or /NC

specifies whether you want to copy (/PC) or verify only (/NC). You must specify one of the switches, which are described in the following section.

srcsw

is one of the optional switches:

/VERIFY
 /BLOCK
 /PARITY
 /DENSITY

See the following section.

NETCPY: Network Copy Between Devices

NETCPY Switches

Table 5-1 lists the seven optional NETCPY switches. The sections following discuss these switches.

Note

You can abbreviate all NETCPY switches to the first two characters.

Table 5-1: NETCPY Switches

Switch	Meaning	Page
/BL OCK	Specifies the block size for magnetic tape.	5-5
/DE NSITY	Specifies the density for magnetic tapes.	5-5
/FC	Specifies that you want to copy a device.	5-6
/HE LP	Displays help for NETCPY.	5-6
/NC	Specifies that you do not want to copy a device but do want to verify the information on a device.	5-7
/PA RITY	Specifies the parity for a magnetic tape.	5-7
/VE RIFY	Specifies that you want to verify the information on a device.	5-7

/BLOCK]:n Switch

Use the /BLOCK switch to specify the block size that NETCPY uses to allocate buffers for magnetic tapes written with nonstandard record sizes. The default block size is 2048-byte blocks. Block sizes must be in multiples of 512 bytes. NETCPY buffers srcdevice and writes and/or verifies dstdevice at the specified block size. For example:

```
#MT0:-BOSTON::MT1:/BL:4096/FC
```

This command causes NETCPY to copy magnetic tape unit 1 at node BOSTON to magnetic tape unit 0 at the local node in 4096-byte blocks.

/DENSITY]:n Switch

Use the /DENSITY switch to set the density for either source or destination magnetic tapes. Depending on the tape drive you use, you can set the density for either 800, 1600, 6250, 6667, or 8000 bpi. For example:

```
#MT0:/DENSITY:1600=BOSTON::MT1:/DENSITY:800/FC
```

NETCPY reads a 9-track magnetic tape on unit 1 at node BOSTON at 800 bpi and writes in 9-track format on magnetic tape unit 0 at the local node at 1600 bpi.

NETCPY: Network Copy Between Devices

/FC Switch

Use the /FC switch to request a device copy. If you do not include the /FC switch, then you must include the /NC switch. If you omit both switches, NETCPY returns an error message (see Table 5-2).

NETCPY lists the progress of the copy in 100-block units for disk and in 100-record units for tape. When the copy is complete, NETCPY lists the time it took and the total number of blocks or records transferred. For example:

```
$RUN DECNET$;NETCPY
NETCPY V4.0 RSTS V9.3 DECNET/E
#BOSTON::DT0:*DT2:/FC
PPN:1,222
Password:

100 BLOCKS TRANSFERRED
200 BLOCKS TRANSFERRED
300 BLOCKS TRANSFERRED
400 BLOCKS TRANSFERRED
500 BLOCKS TRANSFERRED
TIME TO COPY = 5.11111 MIN
COPY COMPLETE
BLOCKS TRANSFERRED = 578
S
```

/HE[LP] Switch

Type /HELP in response to the number sign (#) prompt to get help about NETCPY. For example:

```
$RUN DECNET$;NETCPY
NETCPY V4.0 RSTS V9.3 DECNET/E
#/HE
.
.
.
(Help Display:
.
.
.
#
```

/NC Switch

Use the /NC switch to request no copy. You always use this switch with the /VE (verify) switch for a verify-only run. If you do not include the /NC switch you must include the /FC switch after the srcdevice specification. The /NC switch has no effect by itself.

/PA[RITY]:n Switch

Use the /PARITY switch when copying magnetic tapes to set the parity for either the source or destination device. The parity can be either ODD or EVEN. The default is whatever was set by the system manager for that device at the specified node. For example:

```
#MT0:/DENSITY:1600/PARITY:EVEN=BOSTON::MT1:/DENSITY:600/FC
```

/VE[RIFY] Switch

Use the /VERIFY switch with either the /FC or /NC switch to verify that the information on a device unit has been copied exactly. NETCPY performs a block-by-block verification, taking as much time as the copy operation. When you use the /FC switch, NETCPY first copies the information from the source device to the destination device, and then sends the information back and verifies it. When used with the /NC switch, NETCPY only verifies the information.

NETCPY: Network Copy Between Devices

NETCPY lists the progress of the verification procedure. For example:

```
$RUN DECNET$:NETCPY
NETCPY V4.0 RSTS V9.3 DECNET/E
#BOSTON::DTG:=DT2:/NC/VE
PPN:1,222
Password:

BEGINNING VERIFICATION PASS
 100 BLOCKS VERIFIED
 200 BLOCKS VERIFIED
 300 BLOCKS VERIFIED
 400 BLOCKS VERIFIED
VERIFICATION COMPLETE 0 BAD BLOCKS
OUT OF 492 BLOCKS VERIFIED
$
```

If the information has not been copied correctly, NETCPY prints the decimal number of blocks or records in which inconsistencies appear. The numbers printed correspond to magnetic tape record numbers and to disk and DECTape blocks of 512 bytes. For example:

```
#DX1:=BOSTON::DX2/FC/VE
PPN:102,50
Password:
 100 BLOCKS TRANSFERRED
 200 BLOCKS TRANSFERRED
 300 BLOCKS TRANSFERRED
 400 BLOCKS TRANSFERRED
TIME TO COPY - 7.56667 MIN
COPY COMPLETE
BLOCKS TRANSFERRED = 492

BEGINNING VERIFICATION PASS
THE FOLLOWING BLOCKS ARE BAD
17
31
89
 100 BLOCKS VERIFIED
 200 BLOCKS VERIFIED
 300 BLOCKS VERIFIED
 400 BLOCKS VERIFIED
```

```
VERIFICATION COMPLETE 3 BAD BLOCKS
OUT OF 492 BLOCKS VERIFIED
$
```

NETCPY Error Messages

Table 5-2 lists the NETCPY error messages.

Table 5-2: NETCPY Error Messages

Message and Meaning
CANNOT SPECIFY BOTH /FC AND /NC You must specify either the /FC or /NC switch, but not both.
ERROR IN SPECIFYING DENSITY The density you specified is not one of the valid values of 900, 1600, 6250, 6667, 8000, or DUMP.
ERROR IN SPECIFYING PARITY The parity you specify must be either ODD or EVEN.
/FC OR /NC MUST BE SPECIFIED You must specify either the /FC or /NC switch.
ILLEGAL BLOCK SIZE The block size you select must be a multiple of 512.
INVALID NODE SPECIFICATION Either the nodes you specified did not include the local node, or NETCPY did not recognize the node name.
MUST HAVE SAME TYPE DEVICE Both the source device and the destination device must be either MTn: (magnetic tapes), DTn: (DECtape), DXn: (flexible diskettes), or DRn: (disks).
(NSF COMMAND) -- ABORT NETCPY sets up a logical link to the remote node to effect the transfer requested. If any network error occurs during execution, NETCPY aborts the program and displays this message. The NSF command shown is the one NETCPY was trying to execute when the error occurred. For example, if NETCPY was trying to receive a message from the remote node, NETCPY displays the message "RECEIVE NETWORK ERROR -- PROGRAM ABORT."
TYPE /HE FOR HELP NETCPY does not recognize the command you typed. Use the /HE switch for help.

Chapter 6

NETOFF: Network Shutdown Utility

The NETOFF utility lets you shut down your network in an orderly fashion without shutting down the entire RSTS/E operating system. It causes a gradual shutdown rather than an immediate disconnect. You must have the SHUTUP privilege to use NETOFF.

NETOFF performs a similar operation to the NCP SET EXECUTOR STATE SHUT command (see the DECnet/E System Manager's Guide). Both permit existing logical links to complete before ending network operations, but only NETOFF warns users that the network will close down in some determined period of time. This gives network users an opportunity to respond to the warning message by finishing up network processing or by requesting that network operations be extended.

You can run NETOFF from any terminal in a DECnet/E system by typing:

```
$ RUN DECNET$:NETOFF
```

NETOFF responds by asking how many minutes it should wait before new network links are prohibited (that is, until it effects a SET EXECUTOR STATE SHUT command) and how many minutes it should wait until actual network shutdown.

After this, the following sequence of operations occurs:

1. NETOFF broadcasts a message to all system terminal users. For example:

```
THE NETWORK IS SHUTTING DOWN IN [variable] MINUTES
```

2. NETOFF then stops execution for the appropriate number of minutes, as determined in the dialogue. When it resumes execution, it sends the SET EXECUTOR STATE SHUT message to NSP. If no logical links are open, NSP terminates network operations and the NETOFF program completes. Otherwise, NSP prohibits the forming of any new logical links, but allows normal operations to continue on existing links.

NETOFF: Network Shutdown Utility

3. NETOFF again stops execution, this time for the remainder of the period before final network shutdown. When it resumes execution, it sends the "Set executor state off" message to NSP.
4. At this point, NSP aborts any existing logical links and sends a Network Abort message to any local programs with links open. NSP then shuts down Transport, thus ending network operations.
5. Finally, NETOFF sends a STOP command to the Event Logger program and displays the following message on the terminal:

SHUTDOWN COMPLETE

Note

You can interrupt this sequence at any point by pressing CTRL/C. NETOFF resumes execution and sends the SET EXECUTOR STATE ON command to NSP, aborting the shutdown procedure and resuming normal network operations.

APPENDIX A

NSP REASON CODES FOR CONNECT REJECT AND LINK ABORT

Table A-1 lists the Network Service Protocol reason codes and their associated meanings.

Table A-1: NSP Reason Codes

Code	Reason
000	No error -- user-initiated reject or abort
001	Resource allocation failure
002	Destination node does not exist
003	Node shutting down
004	Destination program does not exist
005	Invalid destination name or source name field
006	Destination program's queue full
007	Unspecified error condition
008	Third party aborted logical link
009	User-initiated link abort
010-020	Reserved
021	Invalid destination address in Connect Initiate Message

NSP REASON CODES FOR CONNECT REJECT AND LINK ABORT

Table A-1: NSP Reason Codes (Cont.)

Code	Reason
022	Invalid destination address in Connect Confirm Message
023	Source address zero in Connect Initiate or Connect Confirm Message
024	Flow control violation -- invalid request count in Link Service Message
025-031	Reserved
032	Too many connects to node
033	Too many connects to destination program
034	Access not permitted
035	Logical link services mismatch
036	Invalid accounting information
037	Segment size too small
038	User aborted, timed out, or canceled link
039	No path to node
040	Flow control failure -- data received when request count zero
041	No current link (cannot recognize destination address)
042	Confirmation from remote system of Disconnect message
043	Image data field too long

APPENDIX B

ERROR MESSAGES GENERATED BY RMS AND DAP

This appendix contains non-NFT error messages that can appear while using the Network File Transfer (NFT) program. These messages can appear for conditions diagnosed by the Record Management Services (RMS), used by NFT and FAI to do local input and output, or for violations of the Data Access Protocol (DAP), used to do remote file transfers across the network.

NFT Error Messages from RMS-11

NFT prints the following error message in response to detected RMS-11 errors:

```
?NFT -- RMS ERROR = nnnnnn
```

The errors that can occur when you use NFT should be understandable, such as error 176440, File not found. However, Table B-1 lists all possible RMS errors. Some descriptions will be obscure to all but those familiar with RMS. If you get one of these errors repeatedly, call a DIGITAL Software Support Specialist and report the error.

Note

The following list reflects errors generated by RMS Version 1.8. See the RMS-11 MACRO-11 Reference Manual for more descriptions.

ERROR MESSAGES GENERATED BY RMS AND DAP

Table B-1: RMS Error Codes

Octal Value (nnnnnn)	Description
177760	Operation aborted: Stack save area exhausted or in-core data structures corrupted.
177740	Files-11 ACP could not access the file.
177720	File activity precludes action (for example, attempting to close a file with outstanding asynchronous record operation).
177700	Bad area identification number (AID) field in allocation XAB (out of sequence).
177660	Invalid value in alignment boundary type (ALN) field of allocation XAB.
177640	Value in allocation quantity (ALQ) field in FAB (or allocation XAB) exceeds maximum or, during an explicit \$EXTEND operation, equals zero.
177620	Records in a file on ANSI-labeled magnetic tape are variable length but not in ANSI-C format.
177600	Invalid value in allocation options (AOP) field in allocation XAB.
177560	Invalid operation at AST level: Attempting to issue a synchronous operation from an asynchronous record operation completion routine.
177540	Read error on file header attributes.
177530	Invalid file ID.
177520	Write error on file header attributes.
177500	Bucket size (BKS) field in FAB contains value exceeding maximum.
177460	Bucket size (BKZ) field in allocation XAB contains value exceeding maximum.
177440	Block length (BLN) field in a FAB, RAB, or XAB is incorrect.

Table B-1: RMS Error Codes (Cont.)

Octal Value (nnnnnn)	Description
177430	Beginning of file detected on \$SPACE operation to magnetic tape file.
177420	Private buffer pool address not a double word boundary.
177400	Private buffer pool size not a multiple of 4.
177360	Internal error detected in RMS-11. No recovery possible; contact a software specialist.
177340	Cannot connect RAB (only one record access stream permitted for sequential files).
177320	\$UPDATE attempting to change a key field that does not have the change attribute.
177300	Index file bucket check-byte mismatch. The bucket has been corrupted. Recovery can be attempted by: - o Moving disk pack to another device and trying the process again. - o Recreating file using either RMSIFL or RMSCNV utility. - o Restoring file from last backup.
177260	\$CLOSE function failed.
177240	Invalid COB field in XAB or XAB type is invalid for the organisation or operation.
177220	Files-11 ACP could not create file.
177200	No current record: Operation not immediately preceded by a successful \$GET or \$FIND.
177160	Files-11 ACP deaccess error during \$CLOSE.
177140	Invalid area number in DAN field of key definition XAB.
177120	Record accessed by RPA access mode has been deleted.

ERROR MESSAGES GENERATED BY RMS AND DAF

Table B-1: RMS Error Codes (Cont.)

Octal Value (nnnnnn)	Description
177100	Any of the following: - o Syntax error in device name. - o No such device. - o Inappropriate device for operation (for example, attempting to create an indexed file on magnetic tape).
177070	Files-11 ACP could not write bucket. RMS-11 deferred the I/O operation until it needed the I/O buffer for another bucket because the user program specified deferred writes.
177060	Syntax error in directory name.
177040	Dynamic memory exhausted: Insufficient space in central space pool or private buffer pool.
177020	Directory not found.
177000	Device not ready.
176770	Device positioning error.
176760	DTP field invalid (STV=0XAB).
176740	Duplicate key detected, "duplicates allowed" attribute not set for one or more key fields.
176720	Files-11 ACP enter function failed.
176700	Environment error: Operation or file organization not specified in ORG\$ macro.
176660	End of file.
176640	Expanded string area in NAM block too short.
176630	File expiration date not reached.
176620	File extend failure.
176600	Not a valid FAB: BID field does not contain FB\$BID.

Table B-1: RMS Error Codes (Cont.)

Octal Value (nnnnnn)	Description
176560	Any of the following: o Record operation attempted was not declared in FAC field of FAB at open time. o Invalid contents in FAC field. o FBSPUT not present in FAC for \$CREATE operation.
176540	File already exists (attempted \$CREATE operation).
176530	Invalid file ID.
176520	Invalid combination of values in FLG field of key definition XAB (for example, "no duplicates" and "changeable keys").
176500	File locked by another user: Cannot access the file because sharing specification cannot be met.
176480	Files-11 ACP \$FIND function failed.
176440	File not found during \$OPEN.
176420	Syntax error in file name.
176400	Invalid file options.
176370	System error during FNA/DNA string parse (STV = system error code).
176360	Device full: Cannot create or extend file.
176340	Invalid area number in IAK field of key definition XAB.
176320	Index not initialized. This code can only occur in the STV field when STS contains ERSRNF.
176300	Invalid RFI field in FAB.
176280	Maximum number (254) of key definition or allocation XABs exceeded or multiple summary, protection, or date XABs present during operation.
176240	\$INIT or \$INITIF macro call never issued.

ERROR MESSAGES GENERATED BY RMS AND DAP

Table B-1: RMS Error Codes (Cont.)

Octal Value (nnnnnn)	Description
176220	Invalid operation. Examples include: - o Attempting a STRUNCATE operation to a nonsequential file. - o Attempting an \$ERASE or \$EXTEND operation to a magnetic tape file. - o Issuing a block mode operation (for example, \$READ or \$WRITE) to a stream not connected for block operations. - o Issuing a record operation (for example, \$GET, \$PUT) to a stream connected for block mode operations.
176200	Invalid record encountered in sequential file: Invalid count field.
176160	Invalid internal stream identifier (ISI) field in RAB (field may have been altered by user) or \$CONNECT never issued for stream.
176140	Key buffer address (KBF) field equals 0.
176120	Record identifier (the 4-byte location addressed by KBF) for random operation to relative file is 0 or negative.
176100	Invalid key of reference (RRF) in RAB: - o During random \$GET or \$FIND operation, or - o During \$CONNECT or \$REWIND. In this case, ERRSRF is returned for the first record operation following the \$CONNECT or \$REWIND.
176060	Key size equals zero or too large (indexed file) or not equal to 4 (relative file).
176040	Invalid area number in IAN field of key definition XAB.
176020	Magnetic tape is not labeled in accordance with ANSI standards.
176000	Logical channel busy.
175760	Invalid value in logical channel number (LCH) field of FAB.

Table B-1: RMS Error Codes (Cont.)

Octal Value (nnnnnn)	Description
175750	Attempt to extend an area containing an unused extent.
175740	Invalid value in LOC field of allocation XAB.
175720	In-core data structures (for example, I/O buffers) corrupted. This code can only occur in the STV field when STS contains ER\$ABO.
175700	Files-11 ACP could not mark file for deletion.
175660	Either of the following: - o Maximum record number field contains a negative value during \$CREATE of relative file. - o Record identifier (pointed to by KBF) for random operation to relative file exceeds maximum record number specified when file created.
175640	Maximum record size (MRS) field contains 0 during \$CREATE operation and: - o Record format is fixed, or - o File organization is relative.
175620	Odd address in Name Block address (NAM) field in FAB on \$OPEN, \$CREATE, or \$ERASE.
175600	Not at end-of-file: Attempting a \$PUT operation to a sequential file when stream is not positioned to EOF.
175560	Cannot allocate internal index descriptor: insufficient room in space pool while attempting to open an indexed file.
175540	No primary key definition XAB present during \$CREATE of indexed file.
175520	\$OPEN function failed.
175500	XABs in chain not in correct order: - o Allocation or key definition XABs not in ascending (or densely ascending) order. - o XAB of another type intervenes in key definition or allocation XAB sub-chain.
175460	Invalid value in file organization (ORG) field of FAB.

ERROR MESSAGES GENERATED BY RMS AND DAF

Table B-1: RMS Error Codes (Cont.)

Octal Value (nnnnnn)	Description
175440	Error in file's prologue: File is corrupted; recovery can be attempted by: - o Moving disk pack to another device. - o Recreating file using either the RMSIFL or RMSCNV utility. - o Restoring file from latest backup.
175420	Key position (POS) field in key definition XAB contains a value exceeding maximum record size.
175400	File header contains bad date and time information (retrieved by RMS-11 because a date and time XAB is present during an \$OPEN or \$DISPLAY operation). File may be corrupted.
175360	Privilege violation: Access to the file denied by the operating system.
175340	Not a valid RAB: BID field does not contain RB\$BID.
175320	Either of the following: - o Invalid values in record access mode (RAC) field of RAB. - o Illogical value in RAC field (for example, RB\$KEY with a sequential file).
175300	Either of the following: - o Invalid values in record attributes (RAT) field of RAB during \$CREATE. - o Illogical combination of attributes (for example, FB\$CR and FB\$FTN) in RAC field during \$CREATE.
175260	Record address (RBF) field in RAB contains an odd address (block mode access only).
175240	Files-11 ACP error: - o In record processing -- read failure on file block. - o In block I/O -- VBN=0.
175220	Record already exists: During a \$PUT operation in random mode to a relative file, an existing record found in the target record position.

Table B-1: RMS Error Codes (Cont.)

Octal Value (nnnnnn)	Description
175200	Invalid RFA in RFA field of RAB during RFA access.
175160	Either of the following: - o Invalid record format in RFX field of FAS during \$CREATE. - o Specified record format is invalid for file organization.
175140	Target bucket locked by another task or another stream in the same program.
175120	Files-11 ACP \$REMOVE function failed.
175100	Record identified by KBF/KSZ fields of RAB for random \$GET or \$FIND operation does not exist in relative or indexed file (for indexed files only, STV may contain ERSIDX). Record may never have been written or may have been deleted.
175060	\$FREE operation issued but no bucket was locked by stream.
175040	Record options (ROP) field contains invalid values or illogical combination of values.
175020	Error while reading prologue.
175000	Invalid RRV record encountered in indexed file. File may be corrupted.
174760	Record stream active: In asynchronous environment, attempting to issue a record operation to a stream that has a request outstanding.
174740	Record size specified in RSZ of RAB during \$PUT or \$UPDATE is invalid: - o RSZ equals zero. - o RSZ exceeds maximum record size (MRS) specified when file created. - o RSZ not equal to size of Current Record for \$UPDATE operation to a sequential file on disk. - o RSZ does not equal MRS (for fixed format records). - o RSZ not large enough to contain Primary Key of indexed file.

ERROR MESSAGES GENERATED BY RMS AND DAP

Table B-1: RMS Error Codes (Cont.)

Octal Value (nnnnnn)	Description
174720	Record too big for user's buffer: RMS-11 could not move entire record retrieved by \$GET operation to user work area (UBF/USZ). Note that this error does not destroy the current context of the stream. Rather, the stream's context is updated as if the operation had been completely successful and as much of the record as possible is moved to user buffer.
174710	RRV update error on insert.
174700	During \$PUT operation, key of record to be written is not equal to or greater than key of previous record (and RAC field contains RBSSEQ).
174660	Illogical value in \$RR field of RAB (for example, FBSWRI specified for sequential file).
174640	Invalid SIZ field in key definition XAB during \$CREATE (for example, specified size exceeds maximum record size).
174620	During asynchronous record operation, RMS-11 has found that the stack is too big to be saved. This code can only occur in the STV field when STS contains ER\$ABO.
174600	System directive error.
174560	Index tree error: indexed file is corrupted.
174540	Syntax error in file type (for example, more than 3 characters specified).
174520	Invalid address in user buffer (UBF) field of RAB: o UBF contains 0, or o UBF not word aligned (for block mode access only).
174500	Invalid user buffer size (USZ) field in RAB (for example, USZ contains 0).
174440	Invalid VOL field in allocation XAB (for example, VOL does not contain 0).
174430	Wildcard encountered during FNA/DNA string parse.

Table B-1: RMS Error Codes (Cont.)

Octal Value (nnnnnn)	Description
174420	File write error.
174410	Device is write locked.
174400	Error while writing prologue.
174360	XAB field in FAB (or NXT field in XAB) contains an odd address.
174340	Extraneous field detected during FNA/DNA string parse.

NFT Error Messages from DAP

NFT prints the following error message in response to an error in NFT/FAL communication, violating the Data Access Protocol:

```
?NFT -- DAP ERROR = aannnn
```

The aa value gives the macro or functional group reason for the error. The nnnn value gives the micro, or specific, reason for the error.

Macro (aa) Field Values

Table B-2 lists the possible values for the macro (aa) field.

Note

This list reflects the errors generated as the result of a violation of DAP Version 5.6, as described in the DIGITAL Network Architecture, "Data Access Protocol Functional Specification," released in October, 1980.

ERROR MESSAGES GENERATED BY RMS AND DAP

Table B-2: Macro (aa) Field Values

Octal Value (aa)	Name	Meaning
0	Pending	Operation in progress.
1	Successful	Returns information that indicates success.
2	Unsupported	This implementation of DAP does not support the specified request.
3		Reserved.
4	File Open	Errors that occur before a file is successfully opened.
5	Transfer Error	Errors that occur after opening a file and before closing that file.
6	Transfer Warning	For operations on open files, indicates the operation completed, but not with complete success.
7	Access Termination	Errors associated with terminating access to a file.
10	Format	Error in parsing a message. Format is not correct.
11	Invalid	Field of message is invalid. For example, bits that are meant to be mutually exclusive are set, an undefined bit is set, a field value is out of range, or an invalid string is in a field.
12	Sync	DAP message received out of synchronization.
13-15		Reserved.
16-17		User-defined STATUS message MACCODE field.

Micro (nnnn) Field Values

Table B-3 lists the micro (nnnn) values for use with macro (aa) values of 2, 10, and 11 octal. These refer to the macro Unsupported, Format, and Invalid categories.

Note

Micro (nnnn) Format: Bits 6-11 specify the DAP message type number. Bits 0-5 specify the DAP message field number.

Table B-3: Micro (nnnn) Field Values

Code (Octal)	Reason
Miscellaneous errors	
00 00	Unspecified DAP message error (catch all)
00 10	DAP message type field (TYPE) error
CONFIGURATION message errors by field:	
01 00	Unknown field
01 10	DAP message flags field (FLAGS)
01 11	Data stream identification field (STREAMID)
01 12	Length field (LENGTH)
01 13	Length extension field (LEN256)
01 14	Bit count field (BITCNT)
01 20	Buffer size field (BUFSIZ)
01 21	Operating system type field (OSTYPE)
01 22	File system type field (FILESYS)
01 23	DAP version number field (VERNUM)
01 24	ECO version number field (ECONUM)
01 25	USER protocol version number field (USRNUM)
01 26	DEC software release number field (SOFTVER)
01 27	User software release number field (USRSOFT)
01 30	System capabilities field (SYSCAP)

ERROR MESSAGES GENERATED BY RMS AND DAP

Table B-3: Micro (nnnn) Field Values (Cont.)

Code (Octal)	Reason
ATTRIBUTES message errors by field:	
02 00	Unknown field
02 10	DAP message flags field (FLAGS)
02 11	Data stream identification field (STREAMID)
02 12	Length field (LENGTH)
02 13	Length extension field (LEN256)
02 14	Bit count field (BITCNT)
02 15	System specific field (SYSPEC)
02 20	Attributes menu field (ATTMENU)
02 21	Data type field (DATATYPE)
02 22	File organization field (ORG)
02 23	Record format field (RFM)
02 24	Record attributes field (RAF)
02 25	Block size field (BLS)
02 26	Maximum record size field (MRS)
02 27	Allocation quantity field (ALQ)
02 30	Bucket size field (BKS)
02 31	Fixed control area size field (FSZ)
02 32	Maximum record number field (MRN)
02 33	Run-time system field (RUNSYS)
02 34	Default extension quantity field (DEQ)
02 35	File options field (FOP)
02 36	Byte size field (BSZ)
02 37	Device characteristics field (DEV)
02 40	Spooling device characteristics field (SDC)
02 41	Longest record length field (LRL)
02 42	Highest virtual block allocated field (HBK)
02 43	End of file block field (EBK)
02 44	First free byte field (FFB)
02 45	Starting LBN for contiguous file (SBN)
ACCESS message errors by field:	
03 00	Unknown field
03 10	DAP message flags field (FLAGS)
03 11	Data stream identification field (STREAMID)
03 12	Length field (LENGTH)
03 13	Length extension field (LEN256)
03 14	Bit count field (BITCNT)
03 15	System specific field (SYSPEC)

Table B-3: Micro (nnnn) Field Values (Cont.)

Code (Octal)	Reason
ACCESS message errors by field:	
03 20	Access function field (ACCFUNC)
03 21	Access options field (ACCOPT)
03 22	File specification field (FILESPEC)
03 23	File access field (FAC)
03 24	File sharing field (SHR)
03 25	Display attributes request field (DISPLAY)
03 26	File password field (PASSWORD)
CONTROL message errors by field:	
04 00	Unknown field
04 10	DAP message flags field (FLAGS)
04 11	Data stream identification field (STREAMID)
04 12	Length field (LENGTH)
04 13	Length extension field (LEN256)
04 14	Bit count field (BITCNT)
04 15	System specific field (SYSPEC)
04 20	Control function field (CTLFUNC)
04 21	Control menu field (CTLMENU)
04 22	Record access field (RAC)
04 23	Key field (KEY)
04 24	Key of reference field (KRF)
04 25	Record options field (ROP)
04 26	Hash code field (HSH)
04 27	Display attributes request field (DISPLAY)
CONTINUE TRANSFER message errors by field:	
05 00	Unknown field
05 10	DAP message flags field (FLAGS)
05 11	Data stream identification field (STREAMID)
05 12	Length field (LENGTH)
05 13	Length extension field (LEN256)
05 14	Bit count field (BITCNT)
05 15	System specific field (SYSPEC)
05 20	Continue transfer function (CONFUNC)

ERROR MESSAGES GENERATED BY RMS AND DAP

Table B-3: Micro {nnnn} Field Values {Cont.}

Code {Octal}	Reason
ACKNOWLEDGE message errors by field:	
06 00	Unknown field
06 10	DAP message flags field {FLAGS}
06 11	Data stream identification field {STREAMID}
06 12	Length field {LENGTH}
06 13	Length extension field {LEN256}
06 14	Bit count field {BITCNT}
06 15	System specific field {SYSPEC}
ACCESS COMPLETE message errors by field:	
07 00	Unknown field
07 10	DAP message flags field {FLAGS}
07 11	Data stream identification field {STREAMID}
07 12	Length field {LENGTH}
07 13	Length extension field {LEN256}
07 14	Bit count field {BITCNT}
07 15	System specific field {SYSPEC}
07 20	Access complete function field {CMPFUNC}
07 21	File options field {FOP}
07 22	Checksum field {CHECK}
DATA message errors by field:	
10 00	Unknown field
10 10	DAP message flags field {FLAGS}
10 11	Data stream identification field {STREAMID}
10 12	Length field {LENGTH}
10 13	Length extension field {LEN256}
10 14	Bit count field {BITCNT}
10 15	System specific field {SYSPEC}
10 20	Record number field {RECNUM}
10 21	File data field {FILEDATA}

Table B-3: Micro (nnnn) Field Values (Cont.)

Code (Octal)	Reason
STATUS message errors by field:	
11 00	Unknown field
11 10	DAP message flags field (FLAGS)
11 11	Data stream identification field (STREAMID)
11 12	Length field (LENGTH)
11 13	Length extension field (LEN256)
11 14	Bit count field (BITCNT)
11 15	System specific field (SYSPEC)
11 20	Macro status code field (MACCODE)
11 21	Micro status code field (MICCODE)
11 22	Record file address field (RFA)
11 23	Record number field (RECNUM)
11 24	Secondary status field (STV)
KEY DEFINITION message errors by field:	
12 00	Unknown field
12 10	DAP message flags field (FLAGS)
12 11	Data stream identification field (STREAMID)
12 12	Length field (LENGTH)
12 13	Length extension field (LEN256)
12 14	Bit count field (BITCNT)
12 15	System specific field (SYSPEC)
12 20	Key definition menu field (KEYMENU)
12 21	Key option flags field (FLG)
12 22	Data bucket fill quantity field (DFL)
12 23	Index bucket fill quantity field (IFL)
12 24	Key segment repeat count field (SEGCNT)
12 25	Key segment position field (POS)
12 26	Key segment size field (SIZ)
12 27	Key of reference field (REF)
12 30	Key name field (KNM)
12 31	Null key character field (NUL)
12 32	Index area number field (IAN)
12 33	Lowest level area number field (LAN)
12 34	Data level area number field (DAN)
12 35	Key data type field (DTP)
12 36	Root VBN for this key field (RVB)
12 37	Hash algorithm value field (HAL)
12 40	First data bucket VBN field (DVB)

Table B-3: Micro (nnnn) Field Values (Cont.)

Code (Octal)	Reason
ALLOCATION message errors by field:	
12 41	Data bucket size field (DBS)
12 42	Index bucket size field (IBS)
12 43	Level of root bucket field (LVL)
12 44	Total key size field (TKS)
12 45	Minimum record size field (MRL)
13 00	Unknown field
13 10	DAP message flags field (FLAGS)
13 11	Data stream identification field (STREAMID)
13 12	Length field (LENGTH)
13 13	Length extension field (LEN256)
13 14	Bit count field (BITCNT)
13 15	System specific field (SYSPEC)
13 20	Allocation menu field (ALLMENU)
13 21	Relative volume number field (VOL)
13 22	Alignment options field (ALN)
13 23	Allocation options field (AOP)
13 24	Starting location field (LOC)
13 25	Related file identification field (RFI)
13 26	Allocation quantity field (ALQ)
13 27	Area identification field (AID)
13 30	Bucket size field (BKZ)
13 31	Default extension quantity field (DEQ)
SUMMARY message errors by field:	
14 00	Unknown field
14 10	DAP message flags field (FLAGS)
14 11	Data stream identification field (STREAMID)
14 12	Length field (LENGTH)
14 13	Length extension field (LEN256)
14 14	Bit count field (BITCNT)
14 15	System specific field (SYSPEC)
14 20	Summary menu field (SUMENU)
14 21	Number of keys field (NOK)
14 22	Number of areas field (NOA)
14 23	Number of record descriptors field (NOR)
14 24	Prologue version number (PVN)

ERROR MESSAGES GENERATED BY RMS AND DAP

Table B-3: Micro (nnnn) Field Values (Cont.)

Code (Octal)	Reason
DATE AND TIME message errors by field:	
15 00	Unknown field
15 10	DAP message flags field (FLAGS)
15 11	Data stream identification field (STREAMID)
15 12	Length field (LENGTH)
15 13	Length extension field (LEN256)
15 14	Bit count field (BITCNT)
15 15	System specific field (SYSPEC)
15 20	Date and time menu field (DATMENU)
15 21	Creation date and time field (CDT)
15 22	Last update date and time field (RDT)
15 23	Deletion date and time field (EDT)
15 24	Revision number field (RVN)
PROTECTION message errors by field:	
16 00	Unknown field
16 10	DAP message flags field (FLAGS)
16 11	Data stream identification field (STREAMID)
16 12	Length field (LENGTH)
16 13	Length extension field (LEN256)
16 14	Bit count field (BITCNT)
16 15	System specific field (SYSPEC)
16 20	Protection menu field (PROTMENU)
16 21	File owner field (OWNER)
16 22	System protection field (PROTSYS)
16 23	Owner protection field (PROTOWN)
16 24	Group protection field (PROTGRP)
16 25	World protection field (PROTWLD)
NAME message errors by field:	
17 00	Unknown field
17 10	DAP message flags field (FLAGS)
17 11	Data stream identification field (STREAMID)
17 12	Length field (LENGTH)
17 13	Length extension field (LEN256)
17 14	Bit count field (BITCNT)
17 15	System specific field (SYSPEC)
17 20	Name type field (NAMETYPE)
17 21	Name field (NAMESPEC)

Table B-3: Micro {nnnn} Field Values (Cont.)

Code (Octal)	Reason
ACCESS CONTROL LIST message errors by field: (reserved for future use)	
20 00	Unknown field
20 10	DAP message flags field (FLAGS)
20 11	Data stream identification field (STREAMID)
20 12	Length field (LENGTH)
20 13	Length extension field (LEN256)
20 14	Bit count field (BITCNT)
20 15	System specific field (SYSPEC)
20 20	Access control list repeat count field (ACLCNT)
20 21	Access control list entry field (ACL)

Table B-4 lists the micro {nnnn} values for use with macro {aa} values of 0, 1, 4, 5, 6, and 7 octal.

Note

Micro {nnnn} Format: Bits 0-11 contains error code number. Symbolic status codes, where supplied, see the corresponding RMS status codes. They are included here for ease of reference only -- they have no meaning for DAP.

Table B-4: Micro {nnnn} Format

Value (Octal)	Symbolic Status Code	Reason
0000		Unspecified error
0001	ERSABO	Operation aborted (STV=ERSSTE/MAP)
0002	ERSACC	File-ACP could not access file (STV=sys err code)
0003	ERSACT	File activity precludes operation
0004	ERSAID	Bad area ID
0005	ERSALN	Alignment options error
0006	ERSALO	Allocation quantity too large or equal to 0
0007	ERSANI	Not ANSI-D format

ERROR MESSAGES GENERATED BY RMS AND DAP

Table B-4: Micro (nnnn) Format (Cont.)

Value (Octal)	Symbolic Status Code	Reason
0010	ER\$AOP	Allocation options error
0011	ER\$AST	Invalid (for example, synch) operation at AST level
0012	ER\$ATR	Attribute read error
0013	ER\$ATW	Attribute write error
0014	ER\$BKS	Bucket size too large
0015	ER\$BKZ	Bucket size too large
0016	ER\$BLN	BLN length error
0017	ER\$BOF	Beginning of file detected
0020	ER\$BPA	Private pool address not multiple of 4
0021	ER\$BPS	Private pool size not multiple of 4
0022	ER\$BUG	Internal RMS error condition detected
0023	ER\$CCR	Cannot connect RAB
0024	ER\$CHG	\$UPDATE changed a key without having attribute of XBSCHB set
0025	ER\$CHK	Bucket format check-byte failure
0026	ER\$CLS	RSTS/E close function failed
0027	ER\$COD	Invalid or unsupported COD field
0030	ER\$CRE	File-ACP could not create file (STV=sys err code)
0031	ER\$CUR	No current record (operation not preceded by \$GET/\$FIND)
0032	ER\$DAC	File-ACP deaccess error during close
0033	ER\$DAN	Data area number invalid
0034	ER\$DEL	RFA-Accessed record was deleted
0035	ER\$DEV	Bad device or inappropriate device type
0036	ER\$DIR	Error in directory name
0037	ER\$DME	Dynamic memory exhausted
0040	ER\$DNF	Directory not found
0041	ER\$DNR	Device not ready
0042	ER\$DPE	Device has positioning error
0043	ER\$DTP	DTP field invalid
0044	ER\$DUP	Duplicate key detected, XBSDUP not set
0045	ER\$ENT	RSX-File-ACP enter function failed
0046	ER\$ENV	Operation not selected in ORG\$ macro
0047	ER\$EOF	End of file
0050	ER\$ESS	Expanded string area too short
0051	ER\$FXP	File expiration date not yet reached
0052	ER\$EXT	File extend failure
0053	ER\$FAB	Not a valid FAB (BID field not = FB\$BID)

Table B-4: Micro (nnnn) Format (Cont.)

Value (Octal)	Symbolic Status Code	Reason
0054	ER\$PAC	Invalid PAC for REC-OP: 0, or FB\$PUT not set for \$CREATE
0055	ER\$PEX	File already exists
0056	ER\$FID	Invalid file ID
0057	ER\$FLG	Invalid flag-bits combination
0060	ER\$FLK	File is locked by other user
0061	ER\$FND	R\$X-FllACP find function failed
0062	ER\$FNF	File not found
0063	ER\$FNM	Error in file name
0064	ER\$FOP	Invalid file options
0065	ER\$FUL	Device/file full
0066	ER\$IAN	Index area number invalid
0067	ER\$IFI	Invalid IFI value or unopened file
0070	ER\$IMX	Maximum NUM (254) areas/key XABS exceeded
0071	ER\$INI	\$INIT macro never issued
0072	ER\$IOP	Operation unknown or invalid for file organization
0073	ER\$IRC	Invalid record encountered (sequential files only)
0074	ER\$ISI	Invalid ISI value on unconnected RAB
0075	ER\$KBF	Bad key buffer address (KBF=0)
0076	ER\$KEY	Invalid key field (KEY=0 or negative)
0077	ER\$KRF	Invalid key-of-reference (\$GET/\$FIND)
0100	ER\$KSZ	Key size too large (IDX/NOT-:(REL))
0101	ER\$LAN	Lowest level index area number invalid
0102	ER\$LBL	Not ANSI labeled tape
0103	ER\$LBY	Logical channel busy
0104	ER\$LCH	Logical channel number too large
0105	ER\$LEX	Logical extend error, prior extend still valid
0106	ER\$LOC	LOC field invalid
0107	ER\$MAP	Buffer mapping error
0110	ER\$MKD	Fll-ACP could not mark file for deletion
0111	ER\$MRN	MRN value = negative or relative key > MRN
0112	ER\$MRS	MRS value = 0 for fixed length records or 0 for relative files
0113	ER\$NAM	NAM block address invalid (NAM = 0 or not accessible)
0114	ER\$NEF	Not positioned to EOF (sequential files only)
0115	ER\$NID	Cannot allocate internal index descriptor

ERROR MESSAGES GENERATED BY RMS AND DAP

Table B-4: Micro (nnnn) Format (Cont.)

Value (Octal)	Symbolic Status Code	Reason
0116	ER\$NPK	No primary key defined for indexed file
0117	ER\$OPN	RSTS/E open function failed
0120	ER\$ORD	XABs not in correct order
0121	ER\$ORG	Invalid file organization value
0122	ER\$PLG	Error in file's prologue (reconstruct file)
0123	ER\$POS	POS field invalid (POS>MRS, STV=XAB indicator)
0124	ER\$PRM	Bad file date field retrieved
0125	ER\$PRV	Privilege violation (OS denies access)
0126	ER\$RAB	Not a valid RAB (BID field not = RB\$BID)
0127	ER\$RAC	Invalid RAC value
0130	ER\$RAT	Invalid record attributes
0131	ER\$RBF	Invalid record buffer address (odd, or not word-aligned if BLK-ID)
0132	ER\$RER	File read error (STV=sys err code)
0133	ER\$REX	Record already exists
0134	ER\$RFA	Bad RFA value (RFA=0)
0135	ER\$RFM	Invalid record format
0136	ER\$RLK	Target bucket locked by another stream
0137	ER\$RMV	RSX-FilACP remove function failed
0140	ER\$RNF	Record not found
0141	ER\$RNL	Record not locked
0142	ER\$ROP	Invalid record options
0143	ER\$RPL	Error while reading prologue
0144	ER\$RRV	Invalid RRV record encountered
0145	ER\$RSA	RAB stream currently active
0146	ER\$RSZ	Bad record size (RSZ>MRS, or not=MRS if fixed length records)
0147	ER\$RTB	Record too big for user's buffer
0150	ER\$SEQ	Primary key out of sequence (RAC=RB\$SEQ for \$PUT)
0151	ER\$SHR	SHR field invalid for file (cannot share sequential files)
0152	ER\$SIZ	SIZ field invalid
0153	ER\$STK	END_ROW Stack too big for save area
0154	ER\$SYS	System directive error
0155	ER\$TRE	Index tree error
0156	ER\$TYP	Error in file type; extension on FNS too big
0157	ER\$UBF	Invalid user buffer address (0, odd, or if BLK-ID not word-aligned)

Table B-4: Micro (nnnn) Format (Cont.)

Value (Octal)	Symbolic Status Code	Reason
0160	ER\$USZ	Invalid user buffer size (USZ=0)
0161	ER\$VER	Error in version number
0162	ER\$VOL	Invalid volume number
0163	ER\$WER	File write error (STV=sys err code)
0164	ER\$WLK	Device is write locked
0165	ER\$WPL	Error while writing prologue
0166	ER\$XAB	Not a valid XAB (_@XAB=odd, STV=XAB indicator)
0167	BUGDD1	Default directory invalid
0170	CAA	Cannot access argument list
0171	CCF	Cannot close file
0172	COA	Cannot deliver AST
0173	CHN	Channel assignment failure (STV=sys err code)
0174	CNTRLO	Terminal output ignored due to Control-O
0175	CNTRLY	Terminal input aborted due to Control-Y
0176	DNA	Default file name string address error
0177	DVI	Invalid device ID field
0200	ESA	Expanded string address error
0201	FNA	File name string address error
0202	FSZ	FSZ field invalid
0203	IAL	Invalid argument list
0204	KFF	Known file found
0205	LNE	Logical name error
0206	NOD	Node name error
0207	NORMAL	Operation successful
0210	OK_DUP	Record inserted had duplicate key
0211	OK_IDX	Index update error occurred; record inserted
0212	OK_RLK	Record locked but read anyway
0213	RFAOK_RRV	Record inserted in primary, but may not be accessible by secondary keys or RFA
0214	CREATE	File was created, but not opened
0215	PBF	Bad prompt buffer address
0216	PNDING	Asynchronous operation pending completion
0217	QUO	Quoted string error
0220	RHE	Record header buffer invalid
0221	RLF	Invalid related file
0222	RSS	Invalid resultant string size
0223	RST	Invalid resultant string address
0224	SQO	Operation not sequential

ERROR MESSAGES GENERATED BY RMS AND DAP

Table B-4: Micro (nnnn) Format (Cont.)

Value (Octal)	Symbolic Status Code	Reason
0225	SUC	Operation successful
0226	SPRSED	Created file superseded existing version
0227	SYN	File name syntax error
0230	TMO	Time-out period expired
0231	ER\$BLK	FB\$BLK record attribute not supported
0232	ER\$BSZ	Bad byte size
0233	ER\$CDR	Cannot disconnect RAB
0234	ER\$CGJ	Cannot get JFN for file
0235	ER\$COF	Cannot open file
0236	ER\$JFN	Bad JFN value
0237	ER\$PEF	Cannot position to end of file
0240	ER\$TRU	Cannot truncate file
0241	ER\$UDF	File is currently in an undefined state; access is denied
0242	ER\$XCL	File must be opened for exclusive access
0243		Directory full
0244		Handler not in system
0245		Fatal hardware error
0246		Attempt to write beyond EOF
0247		Hardware option not present
0250		Device not attached
0251		Device already attached
0252		Device not attachable
0253		Shareable resource in use
0254		Invalid overlay request
0255		Block check or CRC error
0256		Caller's nodes exhausted
0257		Index file full
0260		File header full
0261		Accessed for write
0262		File header checksum failure
0263		Attribute control list error
0264		File already accessed on LUN
0265		Bad tape format
0266		Invalid operation on file descriptor block
0267		Two different devices specified on a rename
0270		New file name specified in rename already in use
0271		Cannot rename old file system
0272		File already open

Table B-4: Micro (nnnn) Format (Cont.)

Value (Octal)	Symbolic Status Code	Reason
0273		Parity error on device
0274		End of volume detected
0275		Data over-run
0276		Bad block on device
0277		End of tape detected
0300		No buffer space for file
0301		File exceeds allocated space; no blocks
0302		Specified task not installed
0303		Unlock error
0304		No file accessed on LUN
0305		Send/receive failure
0306	SPL	Spool or submit command file failure
0307	NMF	No more files
0310	CRC	DAP file transfer checksum error
0311		Quota exceeded
0312	BUGDAP	Internal network error condition detected
0313	CNTRL C	Terminal input aborted due to Control-C
0314	DFL	Data bucket fill size > bucket size in XAB
0315	ESL	Invalid expanded string length
0316	IBF	Invalid bucket format
0317	IBK	Bucket size of LAN not = LAN in XAB
0320	IDX	Index not initialized
0321	IFA	Invalid file attributes (corrupt file header)
0322	IPL	Index bucket fill size > bucket size in XAB
0323	KNM	Key name buffer not readable or writable in XAB
0324	KSI	Index bucket will not hold two keys for key of reference
0325	MBC	Multibuffer count invalid (negative value)
0326	NET	Network operation failed at remote node
0327	CK_AIK	Record is already locked
0330	OK_DEL	Deleted record successfully accessed
0331	OK_LIM	Retrieved record exceeds specified key value
0332	OK_NOP	Key XAB not filled in
0333	OK_RNF	Nonexistent record successfully accessed
0334	PLV	Unsupported prologue version
0335	REF	Invalid key-of-reference in XAB
0336	RSL	Invalid resultant string length
0337	RVU	Error updating RRVs; some paths to data may be lost

ERROR MESSAGES GENERATED BY RMS AND DAP

Table B-4: Micro (nnnn) Format (Cont.)

Value (Octal)	Symbolic Status Code	Reason
0340	SEG	Data types other than string limited to one segment in XAB
0341		Reserved
0342	SUP	Operation not supported over network
0343	WBE	Error on write behind
0344	WLD	Invalid wildcard operation
0345	WSF	Working set full (cannot lock buffers in working set)
0346		Directory listing; error in reading volume-set name, directory name, or file name
0347		Directory listing; error in reading file attributes
0350		Directory listing; protection violation in trying to read the volume-set, directory or file name
0351		Directory listing; protection violation in trying to read file attributes
0352		Directory listing; file attributes do not exist
0353		Directory listing; unable to recover directory list after Continue Transfer (Skip)
0354	SNE	Sharing not enabled
0355	SPE	Sharing page count exceeded
0356	UPI	UPI bit not set when sharing with BRO set
0357	ACS	Error in access control string
0360	TNS	Terminator not seen
0361	BES	Bad escape sequence
0362	PES	Partial escape sequence
0363	WCC	Invalid wildcard context value
0364	IDR	Invalid directory rename operation
0365	STR	User structure (FAB/RAB) became invalid during operation
0366	FTM	Network file transfer mode precludes operation
6000 to 7777		User defined errors

Table B-5 lists the micro (nnnn) values for use with macro (aa) value of 12 octal. This refers to the macro Synchronization category.

Note

Micro (nnnn) Format: Bits 0-11 contains message type number.

Table B-5: Micro (nnnn) Values for Macro (aa) Value Octal 12

Value (Octal)	Reason
0000	Unknown
0001	Configuration
0002	Attributes
0003	Access
0004	Control
0005	Continue transfer
0006	Acknowledge
0007	Access complete
0010	Data
0011	Status
0012	Key definition attributes extension
0013	Allocation attributes extension
0014	Summary attributes extension
0015	Date and time attributes extension
0016	Protection attributes extension
0017	Name
0020	Access control list extended attributes

INDEX

-A-

Active node, 2-3
 Active nodes, 4-8
 Adjacent node, 1-3
 APPEND command
 NFT, 4-12, 4-14, 4-15f
 /APPEND switch
 NFT, 4-12, 4-36
 Appending files, 4-14, 4-36
 /ASCII switch
 NFT, 4-36

-B-

BATCH
 submitting files, 4-29
 Binary files, 4-36
 /BLOCK switch
 NETCPY, 5-5
 NFT, 4-36
 Block transfer
 verification, 5-7
 /BRIEF switch
 NFT, 4-37

-C-

Carriage control, 4-41
 CCL command formats, 2-4
 NFT, 4-5
 TLK, 2-3, 2-6
 Commands, 3-3t, 4-13t, 5-4t
 Concise Command Language
 See CCL
 Contiguous files
 allocation, 4-38
 /CONTIGUOUS switch
 NFT, 4-38
 CONTINUE command
 NET, 3-6
 COPY command
 NFT, 4-16, 4-18f
 Copying devices, 5-1
 Copying files, 4-16
 CTG switch
 NFT, 4-38

CTRL/P command
 NET, 3-7

-D-

DAP, 4-2, 4-3
 version number, 4-33
 DCL Run-Time System, 3-4, 4-5
 DECnet/E
 definition, 1-1
 node, 1-2
 sample network, 1-3f
 utilities, 1-1
 DELETE command
 NFT, 4-19, 4-20f
 /DELETE switch
 NFT, 4-38
 Deleting files, 4-42
 /DENSITY switch
 NETCPY, 5-5
 DETACH command
 NET, 3-8
 Device specification
 NETCPY, 5-2, 5-3
 NFT, 4-10
 Devices
 busy, 4-10
 copying, 5-1
 magnetic tape, 4-10, 4-43, 5-7
 parity, 5-7
 specification (TLK), 2-3, 2-4,
 2-5
 verification of transfer, 5-7
 Dialog mode (TLK), 2-5
 DIRECTORY command
 NFT, 4-21, 4-22f
 Directory listings
 NFT, 4-37, 4-38, 4-40, 4-42,
 4-44
 Directory names, 4-10, 4-26

-E-

Error messages
 DAP, 8-11
 NET, 3-13
 NETCPY, 5-9t

Error messages (Cont.)

NFT, 4-44t
RMS, B-1, B-11t
TLR, 2-10
EXIT command
NET, 3-9
NFT, 4-23

-F-

FAL, 4-3
/FC switch
NETCPY, 5-3, 5-6, 5-7
File Access Listener
See FAL
File specification
NFT, 4-9
Files
appending with NFT, 4-14, 4-36
attributes, 4-12, 4-14, 4-36,
4-39, 4-41, 4-42, 4-44
binary, 4-36
block transfer, 4-36
contiguous allocation, 4-38
copying with NFT, 4-16
deleting, 4-19, 4-42
directory listings, 4-21
image
See Binary files
name and type, 4-11
noncontiguous allocation, 4-42
printing with NFT, 4-27
stream ASCII, 4-41, 4-44
submitting to BATCH, 4-29
superseding existing, 4-43,
4-44
variable-length records, 4-41,
4-44
virtual array data, 4-37
/FULL switch
NFT, 4-38

-H-

HELP command
NET, 3-10
NFT, 4-24
/HELP switch
NETCPY, 5-6

-I-

/IDENTIFY switch
NFT, 4-6, 4-38
Image files
See Binary files
/IMAGE switch
NFT, 4-39
/INQUIRY switch
NFT, 4-12
INQUIRY switch
NFT, 4-39

-L-

Line input
NET, 3-11
Link
logical, 4-3
physical, 1-3
/LIST switch
NFT, 4-40
Local node, 1-3
/LOG switch
NFT, 4-40
Log-in information
NETCPY, 5-2
NFT, 4-6, 4-8, 4-10, 4-25, 4-38
Logical link, 4-3
LSN, 2-1, 2-4, 2-7

-M-

Macro field values, B-11t
Magnetic tape, 4-10, 4-43
Micro field values, B-13t
Micro format, B-20t
Micro values, B-28t
/MORE or - switch
NFT, 4-40

-N-

Named directories, 4-10, 4-26
/NATIVE switch
NFT, 4-41
/NC switch
NETCPY, 5-7
NC switch
NETCPY, 5-7
NCP, 4-8

NET

- commands, 3-5t
- providing access, 3-2f

NET utility, 3-1

- capabilities, 3-1
- commands, 3-4
- CONTINUE command, 3-6
- CTRL/P command, 3-7
- DETACH command, 3-8
- error messages, 3-13
- EXIT command, 3-9
- HELP command, 3-10
- line input, 3-11
- NORMAL command, 3-11
- ODT command, 3-12
- running, 3-2
- single-character input, 3-11

NETCPY

- allowable devices, 5-3
- capabilities, 5-1
- command format, 5-2
- device specification, 5-2, 5-3
- error messages, 5-9t
- log-in information, 5-2
- node specification, 5-2, 5-3
- running, 5-2
- switches, 5-4t
- switches /BLOCK switch, 5-5
- switches /DENSITY switch, 5-5
- switches /FC switch, 5-3, 5-6, 5-7
- switches /HELP switch, 5-6
- switches /NC switch, 5-7
- switches /PARITY switch, 5-7
- switches /VERIFY switch, 5-7
- verifying transfer, 5-7

NETOFF, 6-1

Network Pseudo Keyboard Driver

See NPKDVR

Network Services Protocol

See NSP

Network shutdown

See NETOFF

NFT

- commands, 4-13t
- error messages, 4-44t
- error messages from DAP, B-11
- error messages from RMS, B-1
- sample network, 4-8f
- switches, 4-34t

NFT utility, 4-1

- APPEND command, 4-12, 4-14
- appending files, 4-14, 4-36
- capabilities, 4-1
- CCL command format, 4-5
- command format, 4-6
- commands, 4-13
- continuing command line, 4-40
- COPY command, 4-16
- copying files, 4-16
- DELETE command, 4-19
- deleting files, 4-42
- device specification, 4-10
- DIRECTORY command, 4-21
- directory listings, 4-21, 4-37, 4-38, 4-40, 4-42, 4-44
- EXIT command, 4-23
- file specification, 4-9
- HELP command, 4-24
- log-in information, 4-6, 4-8, 4-10, 4-25, 4-38
- node specification, 4-7
- NODESPECIFICATION command, 4-25
- PRINT command, 4-27
- printing files, 4-27
- quoted strings, 4-12
- running, 4-5
- SUBMIT command, 4-29
- switches
 - /APPEND, 4-12, 4-36
 - /ASCII, 4-36
 - /BLOCK, 4-36
 - /BRIEF, 4-37
 - /CONTIGUOUS, 4-38
 - /CTG, 4-38
 - /DELETE, 4-38
 - /FULL, 4-38
 - /IDENTIFY, 4-6, 4-38
 - /IMAGE, 4-39
 - /INQUIRY, 4-12, 4-39
 - /LIST, 4-40
 - /LOG, 4-40
 - /MORE or -, 4-40
 - /NATIVE, 4-41
 - /NOATTRIBUTES, 4-42
 - /NOCONTIGUOUS, 4-42
 - /NODELETE, 4-42
 - /NOHEADING, 4-42
 - /NOSUPERSEDE, 4-43
 - /POS, 4-43
 - /RWC, 4-43

NFT utility
 switches (Cont.)
 /RWO, 4-43
 /SUPERSEDE, 4-44
 /TOTAL, 4-44
 /VARIABLE, 4-44
 system password, 4-26
 TYPE command, 4-31
 VERSION command, 4-33
 wildcards, 4-11
 /NOATTRIBUTES switch
 NFT, 4-42
 /NOCONTIGUOUS switch
 NFT, 4-42
 Node
 active, 2-3, 4-8
 adjacent, 1-3
 local, 1-3
 remote, 1-3
 specification (NETCPY), 5-2
 specification (NFT), 4-7
 specification (TLK), 2-3, 2-4,
 2-5
 Node specification
 NETCPY, 5-3
 /NODELETE switch
 NFT, 4-42
 NODESPECIFICATION command
 NFT, 4-25
 NODESPECIFICATION command (NFT),
 4-6
 /NOHEADING switch
 NFT, 4-42
 Noncontiguous files
 allocation, 4-42
 NORMAL command
 NET, 3-11
 /NOSUPERSEDE switch
 NFT, 4-43
 NPKDVR, 3-1, 3-9
 NSP
 reason codes, A-1t

-O-

Octal Debugging Tool
 See ODT
 ODT (Octal Debugging Tool), 3-12
 ODT command
 NET, 3-12
 single-character input, 3-12

One-line mode (TLK), 2-2
 Operator Services program
 See OPSER
 OPSER, 2-4, 2-8

-P-

PARITY switch
 NETCPY, 5-7
 Password, 4-9, 4-26
 system, 4-26
 Peripheral Interchange Program
 See PIP
 Physical link, 1-3
 PIP, 4-37, 4-38, 4-40
 /POS switch
 NFT, 4-43
 PPN, 4-10, 4-25
 wildcarding, 4-11
 PRINT command
 NFT, 4-27, 4-28f
 Protocols
 DAP, 4-2
 Pseudo keyboard driver
 See NPKDVR

-Q-

Quote characters
 NFT, 4-12
 Quoted strings
 NFT, 4-12

-R-

Record Management Services
 See RMS
 Remote node, 1-3
 RMS, 4-3, 4-41, 4-44
 /RWC switch
 NFT, 4-43
 /RWO switch
 NFT, 4-43

-S-

SHOW ACTIVE NODES, 4-8
 Single-character input
 ODT command, 3-12
 Single-character input (NET),
 3-11

Stream ASCII, 4-41, 4-44

Strings

quoted

NFT, 4-12

SUBMIT command

NFT, 4-29, 4-30f

/SUPERSEDE switch

NFT, 4-44

Superseding files, 4-43, 4-44

Switches

NETCPY, 5-4

NFT, 4-34

System password, 4-9, 4-26

-T-

Terminal Listener

See LSN

TLK, 2-1

capabilities, 2-1

CCL command format, 2-3, 2-6

device specification, 2-3, 2-4,
2-5

dialog mode, 2-5

error messages, 2-10t

node specification, 2-3, 2-4,
2-5

one-line mode, 2-2

terminal listener (LSN), 2-1,
2-4, 2-7

utility with LSN, 2-2f

/TOTAL switch

NFT, 4-44

TYPE command

NFT, 4-32f

-U-

Utility

NET

introduction, 3-1

NFT

CCL command format, 4-5

introduction, 4-1

TLK

CCL command format, 2-3

introduction, 2-1

-V-

/VARIABLE switch

NFT, 4-44

Variable-length record files,
4-41, 4-44

/VERIFY switch

NETCPY, 5-7

VERSION command

NFT, 4-33

Version number

DAP, 4-33

Virtual array data files, 4-37

-W-

Wildcards, 4-39

NFT commands, 4-11

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA
and Puerto Rico
call **800-258-1710**

In Canada
call **800-267-6146**

In New Hampshire,
Alaska or Hawaii
call **603-884-6660**

DIRECT MAIL ORDERS (U.S. and Puerto Rico*)

DIGITAL EQUIPMENT CORPORATION
P.O. Box CS2008
Nashua, New Hampshire 03061

DIRECT MAIL ORDERS (Canada)

DIGITAL EQUIPMENT OF CANADA LTD.
940 Belfast Road
Ottawa, Ontario, Canada K1G 4C2
Attn: P&SG Business Manager

ELECTRONIC ORDERING

Dial 800-DEC-DEMO with any VT100 or VT200
compatible terminal and a 1200/2400 baud modem.
If you need assistance, call 800-DEC-INFO.

INTERNATIONAL

DIGITAL EQUIPMENT CORPORATION
P&SG Business Manager
c/o Digital's local subsidiary
or approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital
Equipment Corporation, Northboro, Massachusetts 01532

*Any prepaid order from Puerto Rico must be placed
with the Local Digital Subsidiary:
809-754-7575

Reader's Comments

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement. _____

Did you find errors in this manual? If so, specify the error and the page number. _____

Please indicate the type of user/reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code
or Country _____

Do Not Tear - Fold Here and Tape

digital



No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN: Office Systems Documentation MK21-2/E02
DIGITAL EQUIPMENT CORPORATION
Continental Boulevard
Merrimack N.H. 03054



Do Not Tear - Fold Here and Tape

Cut Along Dotted Line