DECnet–RSX

digital

Guide to User Utilities

Order Number: AA–H223F–TC

DECnet–RSX

Guide to User Utilities

October 1989

This manual explains how to use the DECnet–RSX user utilities and describes the user commands.

Supersession/Update Information:	Revised for new covers.
Operating System and Version:	RSX–11M V4.5 RSX–11S V4.5 RSX–11M–PLUS V4.3 Micro/RSX V4.3
Software Version:	DECnet–11M V4.5 DECnet–11S V4.5 DECnet–11M–PLUS V4.3 DECnet–Micro/RSX V4.3

Order Number: AA–H223F–TC

AA–H223F–TC
October 1989

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

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

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

The following are trademarks of Digital Equipment Corporation:

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

This manual was produced by Networks and Communications Publications.

Contents

Preface

1 Introduction

2 The Network Control Program (NCP)

3 Network File Operations from DCL

4 The Network File Transfer (NFT) Utility

5 The File Transfer Spooler (FTS) Utility

6 The Network Command Terminal (NCT) Utility

7 The Remote Terminal (RMT) Utility

8 The Phone Communications (PHO) Utility

9 The Terminal Communications (TLK) Utility

A Status and Error Messages

B FTS Log File Messages

C QIO Functions Supported by the HT: Driver

Figures

Tables

Preface

The *DECnet–RSX Guide to User Utilities* describes how to display information about node, line, and circuit states, set alias node names, transfer files, queue file operations, access the resources of remote DECnet nodes, and communicate with other terminal users; it also includes information on QIO functions for the HT: driver.

Intended Audience

This manual is for you if you use a DECnet–RSX terminal. It assumes that you are familiar with the RSX operating system.

Structure of This Manual

Chapter 1 Outlines the DECnet–RSX user utilities and includes a sample session that shows how you can use the utilities described in this manual.

Chapter 2 Describes the Network Control Program (NCP). NCP lets you display node, line, and circuit information and set alias node names.

Chapter 3 Describes how to use the Digital Command Language to perform file operations such as printing, deleting, and copying files on a local or remote node.

Chapter 4 Describes how to use the Network File Transfer (NFT) utility to perform file operations from the MCR command language interpreter. NFT performs file operations such as printing, deleting, and copying files on a local or remote node.

Chapter 5 Describes the File Transfer Spooler (FTS) utility, which queues file operations. You can use FTS only from MCR.

Chapter 6 Describes the Network Command Terminal (NCT) utility. NCT lets you log on to and use any host operating system that supports network command terminals.

Chapter 7 Describes the Remote Terminal (RMT) utility. RMT lets you access the resources of a remote DECnet–RSX node.

Chapter 8 Describes the Phone Communications (PHO) utility. PHO lets you have an interactive dialog with a remote user on a DECnet–RSX, VAX, or PRO/DECnet system. PHO is available with RSX–11M–PLUS and Micro/RSX.

Chapter 9 Describes the Terminal Communications (TLK) utility. TLK lets you send single-line or dialog messages to other terminal users.

Appendix A Lists and explains error and status messages displayed by the user utilities.

Appendix B Explains FTS user log file entries.

Appendix C Describes QIO functions that can be executed over an RMT line with the HT: driver.

Associated Documents

To supplement the *DECnet–RSX Guide to User Utilities*, refer to the following DECnet–RSX and RSX–11M/M–PLUS documents:

- *DECnet–RSX User's Pocket Guide*
- *DECnet–RSX Programmer's Reference Manual*
- *DECnet–RSX Programmer's Pocket Guide*
- *DECnet–RSX Network Management Concepts and Procedures*
- *DECnet–RSX Guide to Network Management Utilities*
- *IAS/RSX–11 I/O Operations Reference Manual*

Graphic Conventions

Convention	Meaning
`Special type`	This special type shows examples of user input (in) or system output (in black).
UPPERCASE	Uppercase letters indicate characters to type exactly as shown. You can type the text in upper- or lowercase. You can abbreviate uppercase words to the first three or more unique characters.
lowercase italics	Lowercase italics indicate variables for which you specify or the system supplies the actual values.
[ ]	Square brackets enclose optional data. If the brackets enclose a vertical list of options, you can specify only one option. Do not type the brackets when you enter the command.
{ }	Braces enclose options, from which you must choose one and only one. Do not type the braces when you enter the command.
( )	Parentheses enclose a set of options. You must specify both or neither of them. Do not type the parentheses when you enter the command.
key	This symbol indicates a key to press. CTRL/*x* indicates that you press the CONTROL key and the key represented by *x* together.
> NCP>	Most examples in this manual represent terminals in MCR mode. However, you can type a utility name at an MCR or DCL prompt to invoke the utility and have it prompt you for input.

1 Introduction

1.1 Capabilities of DECnet–RSX User Utilities

The DECnet–RSX user utilities let you do the following:

- Examine node, line, and circuit states. The Network Control Program (NCP) lets you display node, line, and circuit information at your terminal. The local node information includes the node's name, address, and current state. The remote node information includes the node's type and current status, and identifies its associated lines and circuits. NCP also provides information on the state of a specific line or circuit. Chapter 2 explains NCP.

- Specify alias node names. NCP also lets you assign or change alias node names. Alias node names are abbreviated pseudonyms that you use for simplifying a node name or node name and access control specification (see Section 2.7).

- Transfer and manipulate files. The Network File Transfer (NFT) and the File Transfer Spooler (FTS) utilities let you transfer files to another user on the same node or a remote node. NFT and FTS also have a variety of file transfer and manipulation capabilities, such as transferring local command files to a remote node for execution, spooling files to a line printer, and appending files to an existing file. NFT executes user commands immediately, while FTS can queue requests for later processing.

  If your command language interpreter is DCL, you use the commands in Chapter 3 to invoke NFT file operations. If your command language interpreter is MCR, you use the NFT and FTS operations described in Chapters 4 and 5, respectively.

- Use the resources of remote host nodes. The Network Command Terminal (NCT) utility lets you log on to remote nodes that support the Terminal Services Architecture (TSA).Once logged on to a remote system, you communicate with the host operating system and programs running under that operating system as if your terminal were physically connected to it. NCT is available on RSX-11M-PLUS systems, RSX-11M/RSX-11S V4.2 (or later) systems, and Micro/RSX systems. Chapter 6 explains the NCT utility.

  The Remote Terminal (RMT) utility lets you use the resources of remote DECnet-RSX nodes. RMT logically connects your terminal to a remote node that supports the RMT host facility. You can then use your terminal as though it were physically connected to the remote host node. Chapter 7 explains the RMT utility.

- Communicate with DECnet-RSX, DECnet-VAX, and PRO/DECnet users. The Phone Communications (PHO) utility lets you have an interactive dialog with up to five other users. PHO lets you "phone" any user on your local node or on a remote DECnet-RSX, DECnet-VAX, or PRO/DECnet node. PHO has many advanced features that are not available with the TLK utility (described next). PHO is available only on RSX-11M-PLUS and Micro/RSX systems. Chapter 8 explains the PHO utility.

  The Terminal Communications (TLK) utility lets you send a single-line message or have a dialog with another terminal user. TLK lets you communicate with another user on your local DECnet-RSX node or on a remote node that supports the TLK utility. Chapter 9 explains the TLK utility.

Table 1-1 summarizes the DECnet-RSX utilities that this manual describes. The utilities are listed in chapter order. The first column in the table names the utility and the second column summarizes the utility's function. Note that certain chapters in this manual are for users of a particular command language interpreter (CLI). For information on file operations, DCL users should read the DCL chapter and MCR users should read the NFT and FTS chapters. For information on the specific RSX systems that support each utility, see the introductory sections of each chapter.

Table 1-1: DECnet-RSX User Utilities

Utility	What It Does
Network Control Program (NCP)	Displays information on node, line, and circuit states; sets and displays alias node names.
Network File Transfer (NFT)	Performs a variety of file operations, such as transferring, copying, and queuing files. If DCL is your command language interpreter, you invoke NFT operations with DCL commands; if MCR is your command language interpreter, you use NFT commands.
File Transfer Spooler (FTS)	Performs the same file operations as NFT, but can also queue operations. FTS is only available from MCR.
Network Command Terminal (NCT)	Lets you log on to and use another host node in the network.
Remote Terminal (RMT)	Lets you log on to and use another RSX host node in the network.
Phone Communications (PHO)	Lets you have an interactive dialog with up to five other users. PHO simulates telephone features, such as setting up conference calls, putting callers on hold, and so forth.
Terminal Communications (TLK)	Lets you send single-line messages or have a dialog with another user.

1.2 Sample Session with the DECnet-RSX User Utilities

The following sample session shows a sequence of operations that use the utilities described in this manual. In this session, HOME is your local node name, and UTICA is the remote node name. You are user Jeff on node HOME. MCR is the command line interpreter.

Initially, you use NCP to confirm that remote node UTICA is ''reachable'' from your node.

```
>
NCP>

Node summary as of 14-JAN-86 13:12

Remote                        Active                      Next
Node               State      Links     Delay    Circuit  Node

4.19 (UTICA)       Reachable  1         5        DMC-1    8.22 (MAYHEM)

NCP>
>
```

Then, you use PHO to see if user Smith is available on node UTICA. You request a directory of users on node UTICA to see if Smith is available, and if so, obtain his TT*n* (target terminal identification).

```
>

               RSX-11M PLUS Phone Facility     15-AUG-86

%DIRECTORY UTICA RET
Press any key to cancel directory listing and continue.
-------------------------------------------------------------

Process Name          User Name      Terminal     Phone Status

BROWNSTONE            BROWNSTONE     unusable     ---
Allen                 GINSBURG       TT2          available
Joe Smith             SMITH          TT4          available
WOOLF                 WOOLF          TT7          available
Gina Simone           SIMONE         TT9          available
IAN                   IAN            TT12         /nobroadcast
Walt Whitman          WHITMAN        TT18         available
Ray Rivera            RIVERA         TT13         available

8 persons listed


>
```

After typing CTRL/Z to exit from PHO, you use TLK in dialog mode to converse with user Smith on node UTICA. Another option is to "dial" Smith from the PHO utility; for more information on this, refer to Chapter 7. When TLK successfully connects with UTICA and Smith agrees to "talk" with you, your terminal displays the <TLK> – START OF DIALOG message. You can then have a dialog with Smith, instructing him to load pack FRED onto drive DR0:.

```
>

<TLK> - START OF DIALOG
TLK>

<TLK> YES. YOU CAN USE DR0:.

TLK>
TLK>
<TLK> - END OF DIALOG
```

Now you use NCT to logically connect your terminal to UTICA. NCT lets you log on to the UTICA host operating system and use your terminal as if it were physically connected to UTICA. You can then mount pack FRED on UTICA.

```
>

Connected to UTICA, System type = RSX-11M-PLUS
System ID: Distributed Systems 11/70

>
ACCOUNT OR NAME:
PASSWORD:

RSX-11M-PLUS V4.0 [1,54] SYSTEM

GOOD MORNING
14-SEP-86 11:18 LOGGED ON TERMINAL RTO:

>
```

Note that the password does not echo on your screen.

You can then use NCP to set the alias (H) for your original node name, HOME, and your account and password combination, JEFF/SECRET. Using the alias will simplify the NFT command line for transferring files to UTICA.

```
>
```

Next, you use NFT to transfer files from your original node, HOME, to your current node, UTICA, for task building. You need not specify an output node name or directory; the output files will default to your current node and directory. You then check the UTICA directory for the special library files used for assembling and task building files.

```
>
NFT>
NFT>

>

DIRECTORY DRO:[1,1]
14-JAN-86 13:18

SPCLIB.OLB      31.      C 22-NOV-86
```

Now you are ready to assemble and task build the files. Remember that NCT logically connects your terminal to UTICA.

```
>
>
```

When the job is done, you dismount the disk, DR0: and log off of the UTICA operating system. This causes NCT to exit, returns control to your local operating system, and completes the session.

```
>
DMO -- RT0: DISMOUNTED FROM DR0: *** FINAL DISMOUNT ***
>
CONNECT TIME:   1  MINS.
CPU TIME USED:  1  SECS.
TASK TOTAL:     7

NCT -- Control returned to node "HOME"
>
```

2 The Network Control Program (NCP)

The Network Control Program (NCP) displays node, line, and circuit information and sets alias node names. NCP is a DECnet utility program that system managers use to control the network. This chapter describes some NCP commands supported on RSX-11M/M-PLUS systems that you might find useful. If you use RSX-11S, refer to Chapter 1 of the *DECnet-RSX Guide to Network Management Utilities* for a description of RSX-11S support for NCP. The *DECnet-RSX Network Management Concepts and Procedures* manual fully describes all of NCP's capabilities.

2.1 Invoking and Exiting NCP

To invoke NCP, type NCP at the MCR or DCL prompt. NCP will prompt you for command input:

```
>NCP RET
NCP>command
```

Where *command* is any NCP command verb or command string.

To exit from NCP, enter CTRL/Z or the EXIT command:

```
NCP> CTRL/Z                   NCP>EXIT RET
>                             >
```

2.2 Abbreviating NCP Commands

Most NCP commands consist of a command name, a component upon which the command acts, and selected parameters for that component. When you enter NCP commands, you can abbreviate any keyword to the first three letters if three letters make the abbreviation unique. For example, the format for the NCP command for displaying node information is:

SHOW NODE *node-id*

or

SHO NOD *node-id*

where *node-id* specifies the name or address of the node. The node ID cannot be abbreviated in NCP commands (see Section 2.4.2).

2.3 Getting Help on NCP Commands

For information about NCP commands, use the HELP command. For a summary of all NCP commands, type

```
NCP>
```

You can also use the HELP command for information about a specific NCP command or component:

```
NCP>
```

```
NCP>
```

The HELP command requires that the NCP HELP files are located on LB:[1,2]. If the files are missing, you get the following message:

```
NCP - No HELP Available
```

If you get this message, ask your system manager to put the HELP files in the correct directory.

For additional assistance, NCP provides command prompting. At any point where you can type a space and where another argument must follow, type a carriage return instead. NCP will prompt you for the required information, as follows:

```
NCP>
Show component:
Node identification:
```

For more information on your choices, press the ESCAPE key in response to any of the NCP prompts.

2.4 Displaying Node Information

To display local or remote node information, use the appropriate NCP SHOW command, as described in the next sections.

All NCP SHOW commands display output at your terminal.However, you can send command output to a file at your local node, as the next section explains.

2.4.1 Sending Command Output to a File

To send NCP command output to a local file, append the keyword TO and the output file specification to the command. For example, to send the SHOW NODE UTICA command output to the file UTICA.LST, type

```
NCP>
```

If the specified output file does not already exist, NCP creates a new file. If the file already exists, NCP appends the output to that file. NCP does not create a new version of the existing output file.

You can also use the Network Display (NTD) utility to display information about local or remote RSX nodes. The NTD utility is described in the *DECnet–RSX Guide to Network Management Utilities*.

2.4.2 Node Identification

Node names are unique names that your system uses to refer to other nodes. A node name is a 1- to 6-character alphanumeric string containing at least 1 alphabetic character. You cannot abbreviate node names in NCP commands. However, you can create aliases to simplify node name specification, as Section 2.7.1 describes. The format for a node address depends on whether or not a node has multi-area support. If a node has multi-area support, the node address consists of an area number identifying that node's area within the multi-area network. A unique decimal integer is also assigned to the node at network generation (for example, 4.19 UTICA). If a node does not have multi-area support, the node address consists of an area number of 1 and a unique decimal integer (for example, 1.18 MIAMI). Multi-area support is described in the *DECnet–RSX Network Management Concepts and Procedures* manual.

NOTE

> Most of the examples in this chapter contain node addresses for single area network nodes. However, for users with multi-area networks, the examples for the NCP SHOW ACTIVE NODES and SHOW KNOWN NODES commands contain multi-area node addresses.
>
> In NCP commands, if you give a node address for a multi-area node without a specific area number for that node, the returned area number defaults to the area of the executor node to which the command is issued.

To obtain information about your own node (the executor node), follow the instructions given in Section 2.4.4. To obtain information about remote nodes, refer to Sections 2.4.5, 2.4.6, and 2.4.7.

2.4.3 Node States

The executor (local) node can be in one of three states:

ON — Communication paths (logical links) can be established with other nodes. This is the normal operating state.

SHUT — When the system manager or another privileged user sets the executor to this state, it refuses new logical links but lets existing links complete. When all links are complete and disconnected, the node is set to the OFF state.

OFF — Your node is isolated from the network. No logical links can exist and any previous links are terminated.

2.4.4 Displaying Local Node Information

To display the name of the node on which you are operating (the executor node), the node address and state, and the system identification, use the SHOW EXECUTOR command, as the following example shows:

```
NCP>

Node summary as of 23-NOV-86 10:36:01

Executor node = 1.18 (AUSTIN)

State = On, Identification =  AUSTIN DISTRIBUTED SYSTEMS
```

2.4.5 Checking to See If a Node Is Reachable

A node is reachable when a communication path can be established between that node and yours. To find out if a specific node is reachable use the NCP SHOW NODE command. All SHOW NODE commands display node state information for remote nodes (see the following text and Section 2.4.6). The format for the NCP SHOW NODE command is:

SHOW NODE *node-id*

where *node-id* specifies the name or address of the remote node.

The following example checks to see if node UTICA is reachable.

```
NCP>

Node summary as of 23-NOV-86 10:46:04

Remote                      Active                Next
Node              State     Links     Delay   Circuit    Node

1.19 (UTICA)      Reachable 0         4       DMC-1      1.22 (PIERRE)
```

The display shows that UTICA is reachable; the connection can be attempted.

If your node is a nonrouting node, remote node state information is not available. The SHOW NODE command must be executed at a reachable full-routing node. In a multi-area network, the full-routing node must reside in the same area as the node for which you are requesting information. To execute SHOW NODE at a reachable full-routing node, use the NCP command TELL. The format for TELL is:

TELL *routing-node-id* SHOW NODE *node-id*

where

routing-node-id is the name or address of a reachable full-routing node at which to execute the SHOW NODE command.

node-id is the name or address of the node about which you want information.

In the following example, your node, HOME, is a nonrouting node. To find out whether remote node UTICA is reachable, you execute the SHOW NODE command at node ERIE. ERIE is an accessible full-routing node:

```
NCP>
```

The command response tells you whether UTICA is reachable from your node.

For more information on TELL, see Section 2.6. For information on displaying adjacent nodes and adjacent node types, see Section 2.4.8.

2.4.6 Displaying Information on Reachable Nodes

A reachable node is a node to which the executor node can establish a communication path. NCP displays names, addresses, and other information for all reachable nodes in the network. The SHOW ACTIVE NODES command is useful when you plan to access several nodes (to transfer files, for example). SHOW ACTIVE NODES displays information on all reachable nodes. Using SHOW ACTIVE NODES lets you avoid repeating SHOW NODE for each node. On a large network, however, you might prefer to use SHOW NODE, which limits the display to the node you specify.

The MCR and DCL command syntax for the SHOW ACTIVE NODES command are as follows:

MCR Command Syntax:	**DCL Command Syntax:**
NCP>SHOW ACTIVE NODES RET	$ SHOW NETWORK RET

The following is an example of the SHOW ACTIVE NODES display:

```
Active nodes summary as of 23-NOV-86 11:02:33

Executor node = 6.18 (AUSTIN)

State = On, Identification = AUSTIN DISTRIBUTED SYSTEMS
Active links = 2

Remote                        Active                    Next
Node                State     Links    Delay  Circuit   Node

2.2 (HELENA)        Reachable   2      4      DMC-2     6.17 (REXHUB)
2.3 (UTICA)         Reachable   0      4      DMC-1     8.22 (PIERRE)
4.7 (TACOMA)        Reachable   0      18     DMC-0     4.7 (TACOMA)
4.8 (BRONX)         Reachable                 DMC-0     9.21 (ROCK)
4.9 (LONDON)        Reachable                 DMC-0     11.30 (TOLEDO)
6.10 (ERIE)         Reachable                 DMC-0     8.24 (TAMPA)
9.15 (GENEVA)       Reachable                 DMC-0     11.16 (BOISE)
```

2.4.7 Displaying Information on Known Nodes

You can display information on the names and states of all nodes known to your node. The SHOW KNOWN NODES command lists each node's name and tells whether that node is currently reachable or unreachable. The display is the same as the display for the SHOW ACTIVE NODES command. The following example displays this information for node AUSTIN:

```
NCP>

Known nodes summary as of 23-NOV-86 14:21:22

Executor node = 6.18 (AUSTIN)

  State = On, Identification = AUSTIN DISTRIBUTED SYSTEMS
  Active links = 0

Remote                    Active                   Next
Node              State       Links   Delay  Circuit    Node

2.1 (BUTTE)       Unreachable  0      6                 2.1 (BUTTE)
2.2 (HELENA)      Reachable    2      4      DMC-2      6.17 (REXHUB)
2.3 (UTICA)       Reachable    0      4      DMC-1      8.22 (PIERRE)
4.6 (WACO)        Unreachable                           8.26 (RDVAX)
4.7 (TACOMA)      Reachable    0      18     DMC-0      4.7 (TACOMA)
4.8 (BRONX)       Reachable                  DMC-0      9.21 (ROCK)
4.9 (LONDON)      Reachable                  DMC-0      11.30 (TOLEDO)
9.10 (ERIE)       Reachable                  DMC-0      8.24 (TAMPA)
9.12 (ATHENS)     Unreachable                           8.23 (VAXWRK)
```

Note that nodes 2.1 (BUTTE), 4.6 (WACO), and 9.12 (ATHENS) are unreachable.

2.4.8 Displaying Information on Adjacent Nodes

The SHOW ADJACENT NODES command displays the names of all nodes adjacent to yours. Adjacent nodes are those that are physically connected or on the same Ethernet. The following example shows the adjacent node display:

```
NCP>

Adjacent nodes status as of 10-NOV-86 15:22:07

Remote                 Active                      Next
Node           State     Links    Delay    Circuit    Node

1.6 (HELENA)   Reachable 1        3        UNA-0      1.6 (HELENA)
1.9 (DEL)      Reachable                   UNA-0      1.9 (DEL)
1.20 (NYC)     Reachable                   UNA-0      1.20 (NYC)
1.22 (PIERRE)  Reachable                   DMC-2      1.22 (PIERRE)
1.24 (TAMPA)   Reachable                   UNA-1      1.24 (TAMPA)
1.30 (TOLEDO)  Reachable 1        2        DMC-0      1.30 (TOLEDO)
1.32 (VAX4)    Reachable                   UNA-0      1.32 (VAX4)
```

To display the node type for adjacent nodes, use the NCP command SHOW ADJACENT NODES STATUS, as in the following example:

```
NCP>

Adjacent nodes status as of 10-AUG-86 16:23:07

Remote                         Active
Node              State        Links   Delay   Type           Cost   Hops  Circuit

1.3               Reachable                    Routing IV     3      1     UNA-1
   Next node = 1.3
1.6 (HELENA)      Reachable                    Nonrouting IV  3      1     UNA-0
   Next node = 1.6 (HELENA)
1.8 (BRONX)       Reachable    1       3       Area           3      1     UNA-0
   Next node = 1.8 (BRONX)
1.16 (BOISE)      Reachable                    Routing IV     4      1     DMC-2
   Next node = 1.16 (BOISE)
1.19 (DULUTH)     Reachable    1       2       Nonrouting IV  3      1     UNA-1
   Next node = 1.19 (DULUTH)
1.22 (PIERRE)     Reachable                    Routing IV     4      1     DMC-0
   Next node = 1.22 (PIERRE)
1.24 (TAMPA)      Reachable                    Nonrouting IV  3      1     UNA-0
   Next node = 1.24 (TAMPA)
```

2.5 Displaying Line and Circuit Information

The following sections contain information on DDCMP and Ethernet lines and circuits. For information on LAPB (X.25) lines and circuits, refer to the *DECnet–RSX Network Management Concepts and Procedures* manual.

For information on lines and circuits, you can use the NCP SHOW LINE and SHOW CIRCUIT commands. You can request information for a specific line or circuit, for all active lines and circuits, or for all known lines and circuits. Section 2.5.1 discusses line and circuit identification.

2.5.1 Line and Circuit Identification

Each line and circuit has a unique identifier assigned during network generation. Examples of line or circuit identification are DMC–0, DZ–1–0, and DV–0–3. If you do not know a line or circuit identifier, you can display it by using the NCP SHOW KNOWN LINES or SHOW KNOWN CIRCUITS command. The command response displays the identifiers of all lines or circuits known to your node.

You can also use SHOW ACTIVE LINES and SHOW ACTIVE CIRCUITS to list known lines and circuits that are currently in the ON state. Section 2.5.2 identifies the line and circuit states.

2.5.2 Line and Circuit States

Table 2–1 lists and defines some of the line and circuit states. For a complete discussion of line and circuit states, see the *DECnet–RSX Network Management Concepts and Procedures* manual.

Table 2–1: Line and Circuit States

State	Line	Circuit
ON	At least one of the circuits associated with this line is in the ON or the ON-STARTING state.	The circuit is connected to another active node, available for use.
ON-STARTING	Not applicable.	The circuit is ready to be used at this end, but there is no active node connected at the other end of the circuit.
SERVICE	The line is reserved for testing.	The circuit is reserved for testing.
OFF	None of the circuits associated with the line is in the ON state.	The circuit has not been turned on.
CLEARED	The software line controller databases are not loaded.	Associated software line controller databases are not loaded.

2.5.3 Displaying Line and Circuit States

To display the state of a specific line or circuit, use the appropriate NCP command format, as follows:

SHOW LINE *line-id*

SHOW CIRCUIT *circuit-id*

where *line-id* and *circuit-id* specify the line or circuit whose status you want to display.

In the following examples, the SHOW LINE command displays the state of line DMC–0 and the SHOW CIRCUIT command displays the state of circuit PCL–0.1:

```
NCP>

Line summary as of 23-NOV-86 16:15:29

Line               State
DMC-0              On

NCP>

Circuit summary as of 23-NOV-86 16:43:03

Circuit = PCL-0.1

   State = On
   Adjacent node = 1.8 (SHELOB)
```

You can use the SHOW ACTIVE LINES or SHOW ACTIVE CIRCUITS command to display the state of all lines or circuits known to the executor node and currently working, as the following example shows:

```
NCP>

Active circuits summary as of 23-NOV-86 17:09:00

Circuit = DMC-0

        State = On
        Loopback name = A, Adjacent node = 1.17 (ARWEN)

Circuit = DMC-1

        State = On
        Adjacent node = 1.13 (SAM)
```

To display the state of all lines or circuits known to the executor node, use SHOW KNOWN LINES or SHOW KNOWN CIRCUITS, as in the following example:

```
NCP>

Known lines summary as of 23-NOV-86 17:31:04

Line             State

DMC-0            On
DMC-1            On
DMC-2            On
KDP-0-0          Cleared
DUP-0            Cleared
```

2.6 Executing NCP Commands at Remote Nodes

You can instruct a remote node to execute an NCP command for you. Prefix the command with TELL and give the node ID of the node where you want the command to execute. Your terminal displays the command response. For example, you can learn your own node's network status by displaying a remote node's "view" of your node. Tell the other node to execute the SHOW NODE command, giving your node name as the SHOW NODE parameter. The command response will be the same as it would be if your terminal were at the remote node. The format for using TELL is:

TELL *node-id command*

The following command line causes remote node UTICA to execute the SHOW LINE command:

```
NCP>
```

When node UTICA executes the SHOW LINE command, your terminal displays the resulting information.

The remote node at which you execute the command may require that you specify access control information. Access control information follows the node identification:

TELL *node-id* [*/user-id/passwd/accnt*] *command*

where

user-id is a 1- to 39-character string identifying the user at the remote system.

passwd is a 1- to 39-character password needed for access to the remote system.

accnt is a 1- to 39-character string that specifies an account at the remote system. RSX systems do not use this field.

You can use an alias for the full node name specification, including access control information. The next section describes how to set an alias.

2.7 Using Alias Node Names

An alias is a user-assigned name for a network node. You assign an alias to a node by entering a command at your terminal. You can then use the alias to abbreviate the node name in all references to the node. You can also include access control information in the alias to further abbreviate your references to the node.

Aliases also let you increase the flexibility of your programs. For example, you can create a program that interacts with a different node each time you run it, without requiring modification. Within the program, use an alias for the name of a generic node with which to communicate. Then, before running the program to communicate with a specific node, issue the SET ALIAS command, assigning that node the alias name.

The following sections describe the commands for setting, changing, and removing alias node names and displaying information on aliases. More information on aliases is in the *DECnet–RSX Network Management Concepts and Procedures* manual.

2.7.1 Setting Alias Node Names

To specify an alias for a node name, use the SET ALIAS command. The format for the NCP SET ALIAS command is:

SET ALIAS *alias* DESTINATION *destination-node*

where

alias consists of 1 to 6 alphanumeric characters, with at least 1 alphabetic character.

destination-node is the name of a node to which the alias refers, and, optionally, any network access control parameters to associate with the alias for that node.

Nonprivileged users' aliases apply to their own terminals only. Privileged users can set aliases to apply to all users and tasks. See the *DECnet–RSX Network Management Concepts and Procedures* manual for more details.

Example 1:

The following command lets you refer to node UTICA as ME:

```
NCP>
```

All connect requests to node ME from programs associated with your terminal will now go to node UTICA.

Example 2:

The following example shows NFT command lines for operations to node MEXICO, including access control information, before you set an alias:

```
>
>
>
```

To simplify the command lines, you set the alias M to include the node name and access control information:

```
NCP>
```

Now, using the alias M, the three command lines are shorter:

```
>
>
>
```

Example 3:

Using aliases, you can create a generic command file to apply to any node. You use an alias for the node name in the command file; this example uses REM as an alias. Before executing the command file, you assign that alias to the actual node to which you want the command file to apply. The following command file transfers test files to a remote node and sends a message to the remote node's operator.

```
;TXFER.CMD
;
NFT REM::=DM1:[1,10]TEST1,TEST2,TEST3
TLK REM::'RUN TEST FILES, PLEASE.
.EXIT
```

To transfer the test files to node UTICA, you give node UTICA the alias REM before executing the command file:

```
NCP>
```

2.7.2 Changing and Removing Alias Node Names

To change the current definition of an alias, simply enter a new SET ALIAS command, using the old alias name. The new alias overrides the previous alias. For example, if in Example 3, in Section 2.7.1, you later want to transfer the test files to another node, you can use the same command file without modifying it. To transfer the files to node QUEBEC, assign alias node name REM to QUEBEC.

To remove one or all aliases known to your terminal, use the CLEAR ALIAS command:

CLEAR ALIAS *alias*

For example, to remove the alias ME, enter

```
NCP>
```

To remove all aliases known to your terminal, enter

```
NCP>
```

2.7.3 Displaying Alias Node Names

You can display information about aliases with the NCP command SHOW ALIAS. You can display all of your aliases or all systemwide aliases. You can also display information about a particular alias.

2.7.3.1 Displaying Your Aliases

To display all of your aliases, enter the SHOW KNOWN ALIASES command, as in the following example:

```
NCP>

Known aliases summary as of 27-NOV-86 16:17:40

Alias
Name          Scope           Destination

ME            Terminal TT2:   AUSTIN
```

2.7.3.2 Displaying System Alias Node Names

To display all systemwide aliases, enter SHOW ALL ALIASES, as the following example shows:

```
NCP>

All aliases summary as of 27-NOV-86 16:25:34

Alias
Name          Scope             Destination

ME            Terminal TT2:     AUSTIN
B             Terminal TT4:     HELENA/SHELLEY/...
EF            Terminal TT4:     UTICA/FAL
H             Terminal TT4:     AUSTIN/FRANKLIN/...
W             Terminal TT4:     TACOMA/DECNET/...
AUSTIN        Terminal TT10:    AUSTIN/SHELLEY/...
PIERRE        Terminal TT10:    PIERRE/SHELLEY/...
TOPEKA        Terminal TT10:    TOPEKA/SHELLEY/...
TACOMA        Global            TACOMA/DECNET/...
ATHENS        Global            ATHENS/DECNET/...
GENEVA        Global            GENEVA/DECNET/...
```

The system manager sets global aliases, such as TACOMA, ATHENS, and GENEVA in the display. Global aliases are available to all tasks and users. Nonprivileged users set local aliases, which apply only at the user's terminal. For example, only terminal TT2: can use alias ME.

2.7.3.3 Displaying Information About an Alias

To display information about a particular alias, use the following format:

SHOW ALIAS *alias*

where

alias identifies the alias about which to display information.

The following example displays information about the alias ME:

```
NCP>

Alias summary as of 27-NOV-86 17:07:09

Alias
Name          Scope             Destination

ME            Terminal TT2:     AUSTIN
```

3

Network File Operations from DCL

The Digital Command Language (DCL) lets you perform file operations on accessible network nodes. This chapter describes DCL network file operations. It is organized as follows:

- Section 3.1 describes how you specify files for network operations, use wildcards in file specifications, and specify files on non-RSX nodes. It also describes the file descriptor defaults.

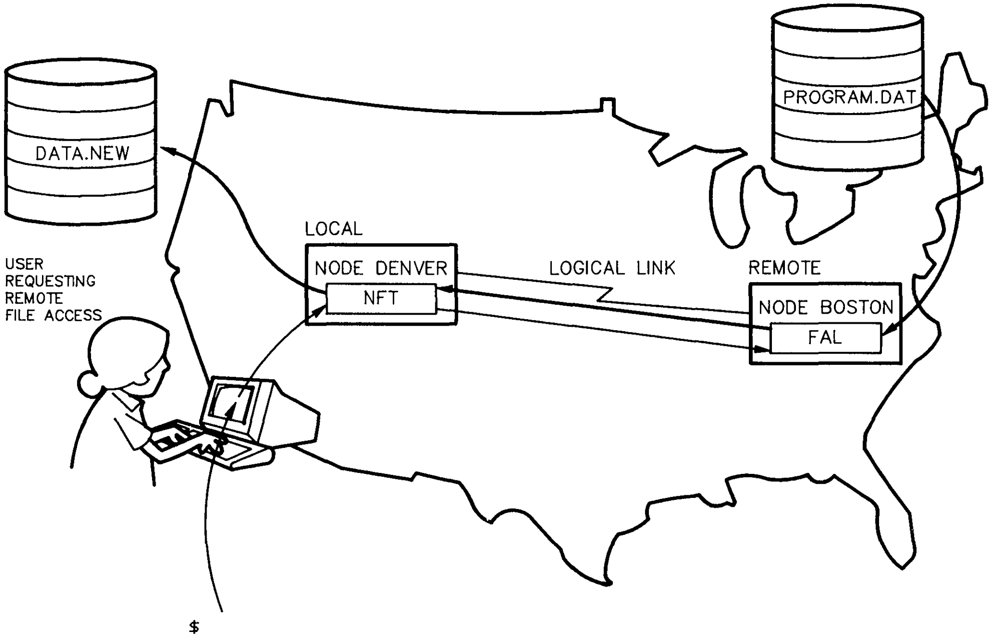
- Section 3.2 describes DCL's network file operations. Table 3.2 summarizes the DCL network file operations.

While you simply enter DCL commands to perform network operations, DCL uses the Network File Transfer (NFT) utility to execute those commands for you. NFT must be installed on your system to perform the operations that this chapter describes. The File Access Listener (FAL) object must be installed on your system to service incoming file requests from NFT at remote nodes (see Figure 3–1).

For information on cross-system file operations, refer to Section 3.1.2.2 and to the *DECnet–RSX Release Notes*.

Figure 3–1 shows a user entering the DCL command COPY. The command copies the remote file PROGRAM.DAT from node BOSTON to local file DATA.NEW. The user includes, as access control information for the remote system, the user name Davis and password SOWHAT.

Figure 3–1: DCL File Copy



LKG–0595

3.1 File Descriptor Syntax

To use DCL file operation commands, you provide descriptors that identify the files, which can be on a local or remote network node. A file descriptor consists of a node and file specification, in the following format:

[*node-specification*::][*file-specification*]

3.1.1 Node Specification

A node specification consists of a node name followed by two colons. You can optionally include access control information for the node. The format is

node-name [*access-control*]::

where

node-name is a 1- to 6-character name that includes at least 1 alphabetic character. If a node name is an alias that includes access control information, you can omit the access control fields. For information on aliases, see Section 2.7.

access-control is a user identification, password, and account specification. You append the access control information to a node name to gain access to the remote file system. You can specify access control information in either of the following formats:

/user-id/passwd/accnt::

or,

"user-id passwd accnt"::

where *user-id* is a 1- to 39-character string that identifies the user at the remote system.

passwd is a 1- to 39-character password for gaining access to the remote file system.

accnt is a 1- to 39-character string that specifies an account at the remote system. RSX systems do not use this field.

To pass access control information to case sensitive systems such as ULTRIX–32, you must use the second format, since the first format passes access control information in uppercase. The embedded spaces in the second format are mandatory.

Different remote systems interpret the access control fields and the access control mechanism differently. If the remote system is a DECnet–RSX node that supports access control verification, the *user-id* field specifies a valid account by last name or UIC. The *passwd* string must be the password for that account, and the *accnt* field is ignored. For more information on access control, see the *DECnet–RSX Network Management Concepts and Procedures* manual.

If the remote node does not support access control verification, its File Access Listener (FAL) permits privileged and nonprivileged access control. For privileged access, use the privileged password specified for FAL during generation of the remote node. The default password is PRIV. Use any desired UIC for the user ID. For nonprivileged access, use the guest UIC and nonprivileged password specified for FAL during generation of the remote node. The default nonprivileged user ID is [200,200] and the default password is GUEST. You can omit the *accnt* field for both privileged and nonprivileged access.

If you have a proxy account on a remote node, you can omit the access control information. If you have multiple proxy accounts, you can include just the user ID in the access control information to identify the account that contains the file.

Examples of Node Specifications:

Examples of valid node specifications follow. You can omit missing fields from the right. For example, you can enter the node specification QUEBEC/[310,2]// as QUEBEC/[310,2]. To denote a missing field on the left, use a slash or blank, depending on the format you use, as in the fourth example.

NOTE

> Brackets ([ ]) are optional around a user ID that has the form of a UIC, as in the last two examples.

```
NODE4/[7,7]/SECRET/ACCNT::
UTICA"5,10 LEFT"::
BOSS/EVERY/ONE::
BOSTON///ACCNTNE::
NODE1"RMES"::
BILBO/1,1/PRIV::
PEORIA/200,200/GUEST::
```

3.1.2 File Specification

A file specification identifies a local or remote file on which to operate. DCL with NFT can parse many foreign file specifications (see Section 3.1.2.2). RSX file specifications have the following format:

dev:[*ufd*]*file-name.type*;*ver*

NOTE

> Enclosing a file specification in quotation marks maintains case sensitivity required for transferring files to and from case sensitive systems such as ULTRIX–32 systems.

For more information on RSX files, refer to the *RSX-11 Utilities Manual*. For information on specifying file names that do not conform to RSX conventions, see Section 4.1.2.2. For information on wildcard characters in file specifiers, see Section 3.1.2.1.

A file in a DCL command can be an input (source) or output (destination) file. Input and output file specifications take different positions in the command line.The node name tells DCL whether the file is local or remote. Files can be:

- Local output files
- Remote output files
- Local input files
- Remote input files

Local or remote input files provide the data for a file operation; local or remote output files receive data from a file operation.

Table 3-1 summarizes defaults for DECnet-RSX file descriptors.

Table 3-1: DECnet-RSX File Descriptor Defaults

Field	Conditions	Default Value
node-name	The file is the first or only file in the input list or the output list.	Local node name.
	The file is a subsequent file in a given list.	Preceding node name specified in the list (including access control information).
user-id	The *user-id* has been assigned using an alias node name. (See Chapter 2 for information on the NCP SET ALIAS command.)	Value of *user-id* specified with the alias node name.
	A node name with access control information has been specified to NFT using the /DF (default) switch.	Value of *user-id* given for the node name.

(continued on next page)

Table 3-1 (Cont.): DECnet-RSX File Descriptor Defaults

Field	Conditions	Default Value
passwd	The *passwd* value has been preassigned using an alias node name. (See Chapter 2 for information on the NCP SET ALIAS command.)	Value of *passwd* specified with the alias node name.
	A node name with access control information has been specified to NFT using the /DF (default) switch.	Value of *passwd* given for the node name.
accnt	The accounting number has been preassigned using an alias node name. (See Chapter 2 for information on the NCP SET ALIAS command.)	Value of *accnt* specified with the alias node name.
	A node name with access control information has been specified to NFT using the /DF (default) switch.	Value of *accnt* given for the node name.
dev	The file is the first or only file in an input list or an output list associated with a particular node.	SY: device associated with the access control given with the specified node name. If no node name is given, it defaults to the user's current SY: device.
	The file is a subsequent file in a given list.	Preceding device specified in the list.
	A device has been specified to NFT using the /DF (default) switch.	Device specified in the /DF command.
ufd	The file is the first or only file in an input list or an output list associated with a particular node.	Directory associated with the access control given with the specified node name. If no node name is given, it defaults to the user's current directory. (RSX-11M-PLUS and Micro/RSX systems now support named directories.)
file-name	The file is the first or only file in the input list.	None.
	The file is a subsequent file in the input list.	Preceding file name specified in the list.
	The file is an output file.	Name of the corresponding input file.

Table 3–1 (Cont.): DECnet–RSX File Descriptor Defaults

Field	Conditions	Default Value
type	The file is the first or only file in the input list.	None.
	The file is a subsequent file in the input list.	Preceding file name specified in the list.
	The file is an output file.	Type of corresponding input file.
ver	The version number is omitted for any input file.	Highest version of file.
	The version number is omitted for an output file.	If a version number was specified for the input file (either wildcard or explicit), the output file will have the corresponding version number. If a version number was not specified for the input file, then no version is used on the output file. On RSX systems this results in the highest version being used.

To use a format that the table does not describe, enclose the foreign file directory and file specification in double quotation marks, as in "*file-specification*". NFT will transmit the file specification to the foreign node without checking syntax or applying defaults for missing fields. File specifications in double quotation marks are not used in determining default values for output files.

Always use quotation marks to enclose file specifications when you transfer files to and from case sensitive systems, such as ULTRIX–32 systems. The quotation marks maintain the upper- and lowercase characters in the file specification.

The conventions of foreign operating systems restrict your use of wildcards in file specifications.The next section describes how you use wildcards in RSX–11 file specifications.

3.1.2.1 Wildcards in NFT File Specifications

NFT lets you use wildcards (asterisks and/or percent signs) in file specifications for existing local or remote files on DECnet–RSX systems. To use a wildcard specifier that NFT does not accept in a remote file specification, enclose it in quotation marks.

An asterisk (*) in a field means that NFT will take any value within that field. These are the wildcard field specifiers:

[*,_m_]_file-name.type_;_ver_	takes all group numbers.
[_g_,*]_file-name.type_;_ver_	takes all member numbers.
[*]_file-name.type_;_ver_	takes all named directories.
[_g_,_m_]*._type_;_ver_	takes all file names.
[_g_,_m_]_file-name_.*;_ver_	takes all file types.
[_g_,_m_]_file-name.type_;*	takes all version numbers.

NFT also accepts wildcard characters for part of a field in the file name and file type, except when you are renaming files. The wildcard rules are as follows:

- An asterisk in a portion of a field means that that portion can be replaced with any characters of any length (including none). For example,

SOME*.TYP will accept:	SOME.TYP SOMETHG.TYP SOMEBODY.TYP SOMEBZ.TYP

- A percent sign (%) in a character position accepts any one character (except a null) in that position. RSTS/E uses the question mark (?) rather than the percent sign as a single position wildcard. For example,

MAIL%.MAI will accept:	MAIL1.MAI MAIL2.MAI MAIL3.MAI MAIL4.MAI
IMV%I%%.TXT will accept:	IMV1I22.TXT IMV2I30.TXT IMV3I45.TXT

You can combine multiple wildcards in a single file specifier, as in these examples:

.;* means all versions of all files under the same UIC.

TEST.*;* means all versions and all types of files named TEST.

TEST*.DAT;* means all versions of files having names beginning with TEST and being of type DAT.

TEST%.* means the most recent version of all file types for files with 5-character names beginning with TEST.

Sample command line:

```
NFT>
```

In this example, NFT transfers to node NEWARK the most recent files

- Located on device DM1: on the local node, and
- Listed under any group number for UFD member 10, and
- With a 5-character file name beginning with the characters FIL, and
- With the file type MAC.

3.1.2.2 Foreign File Formats

A foreign file resides on a non-RSX node. A file operation that includes a foreign file must use a file specification that conforms to the foreign system. NFT can parse the following foreign file specification types:

Directory formats:

[*ufd*]	IAS
[*directory*]	VMS
[*ppn*] or (*ppn*)	RSTS/E
<*directory*>	TOPS–20, VMS
(not applicable)	RT–11
/pathname	ULTRIX–32

File specification formats:

file-name.type;ver	IAS, VMS
file-name.type.ver	TOPS–20, VMS
file-name.type	RT–11, RSTS/E

NOTE

> An ULTRIX–32 directory is specified by a path name, such as /usr/users/whitman where "/" denotes the root. Enclose an ULTRIX–32 file specification in quotation marks to maintain case sensitivity.

These are examples of foreign file specifications:

VMS:

SYS$SYSDISK:[MIDLER.DAP]DAPV70.MEM;69

TOPS–20:

PS:[MIDLER]MAIL.TXT.1439

RSTS/E:

SY:(1,4)SYSTEM.HLP

RT–11:

DU:TECO.SAV

ULTRIX–32:

"/usr/users/whitman/main.c"

3.1.3 Logical Names in File Descriptors

RSX–11M–PLUS and Micro/RSX systems have extended logical name support. You can use logical names for the node and file specification in network file specifiers. For incoming remote files, you can use only global logicals; for outgoing local files, you can use both local and global logicals.

The DCL commands DEFINE and ASSIGN create logical names. Using DEFINE, the following example equates a logical name, ADDR, to a node, disk, and directory name. Then it uses the logical name ADDR to simplify the file specification in a DCL TYPE command.

```
$
$
```

Note that the logical name is always at the leftmost part of the file specification.

Certain network commands require that you supply access control information. A logical name definition can contain access control information, but consider the following suggestions for protecting security:

- Do not put a logical name definition that contains access control information in a command file, unless you heavily protect the file.

- Use aliases in combination with logical names. Once you define an alias, the access control information is protected. The NCP command SHOW KNOWN ALIASES displays information about the alias, but does not echo the access control information on a terminal. In contrast, the DCL command SHOW LOGICAL displays the entire equivalence string.

  To create a secure logical name, first put the node name and access control information in an alias. Then use the alias when defining the logical name. Any SHOW LOGICAL command will then show the alias instead of the node name and access control information.

Suppose you want to do NFT operations that require access to user Snyder's account on node UTICA. The following example uses NCP to set the alias REM equal to node UTICA, user ID SNYDER, and password SECRET. Then it substitutes the alias for those elements in a DCL DEFINE command. DEFINE equates logical name SPECS to disk DB1 and directory [275,10] on UTICA. The DIRECTORY command then uses logical name SPECS for all elements of the file specification other than the file name.

```
$
$
$
```

For more information on using logical names, refer to the *RSX-11M-PLUS Command Language Manual.*

3.2 File Operations for DCL Users

To perform file operations from DCL, you use DCL command verbs. DCL uses the Network File Transfer (NFT) utility in performing file operations, but you interact with DCL, not with NFT. To append the contents of one file to the contents of another, for example, you use the APPEND command. Your DCL command invokes NFT's services when you include a node name in your command line.

Most DCL commands for file operations have optional qualifiers that further define the action of the command verb. For example, APPEND/DATA_TYPE = ASCII appends one file to another in ASCII record mode. Table 3–2 summarizes the DCL file operations. Sections 3.2.4 through 3.2.14 describe each operation in detail.

3.2.1 Specifying DCL Qualifiers in a Command Line

A DCL command line for a file operation consists of a command verb, one or more input and/or output file specifications, and optional qualifiers. These qualifiers can act on commands or file specifications:

1. Command qualifiers apply to a DCL command. You can put a command qualifier anywhere in a command line.

2. File qualifiers apply to one or more files on the command line. On an input file specification, a file qualifier applies only to that file. On an output file, however, the switch becomes global and applies to all input files in the command, unless you override the switch for specific files.

You can also make a file qualifier global by appending it to a command verb.

3.2.2 Using Multiple Input Lines

You can use any number of lines to enter a command. To use more than one line to enter a DCL command, enter a hyphen (-) and then press the RETURN key anywhere in the command line. DCL will prompt you for the remainder of the command with an underscore and angle bracket (_>).

```
_>
```

Table 3–2: Summary of the Network File Operations from DCL

Operation	DCL Command	DCL Qualifiers	Scope of Qualifier
Append: appends files to the end of an existing file.	APPEND	/DATA_TYPE = ASCII	file
		/TRANSFER_MODE = AUTO	file
		/TRANSFER_MODE = BLOCK	file
		/DATA_TYPE = IMAGE	file
		/MACY11	file
		/PROTECTION = *level*	file
		/OWNER = [*uic*]	file
		/PRINT	file
		/TRANSFER_MODE = RECORD	file
		/LOG	cmd
		/NOWARNINGS	cmd
Copy: transfers one or more existing files to one or more files on the local or remote node.	COPY	/DATA_TYPE = ASCII	file
		/TRANSFER_MODE = AUTO	file
		/TRANSFER_MODE = BLOCK	file
		/CONTIGUOUS	file
		/DATA_TYPE = IMAGE	file
		/MACY11	file
		/NEW_VERSION	file
		/NO_SPAN	file
		/PROTECTION = *level*	file
		/OWNER = [*uic*]	file
		/PRINT	file
		/TRANSFER_MODE = RECORD	file
		/REPLACE	file
		/SUBMIT	file
		/CONCATENATE	cmd
		/LOG	cmd
		/NOWARNINGS	cmd
Create: creates an output file that previously did not exist.	CREATE	/CONTIGUOUS	file
		/MACY11	file
		/NEW_VERSION	file
		/NO_SPAN	file
		/PROTECTION = *level*	file
		/OWNER = [*uic*]	file
		/PRINT	file
		/REPLACE	file
		/SUBMIT	file
		/LOG	cmd
Delete: deletes one or more files.	DELETE	/LOG	cmd
		/NOWARNINGS	cmd

(continued on next page)

Table 3-2 (Cont.): Summary of the Network File Operations from DCL

Operation	DCL Command	DCL Qualifiers	Scope of Qualifier
Directory: lists a directory of one or more files.	DIRECTORY	/BRIEF	cmd
		/FULL	cmd
		/ATTRIBUTES	cmd
		/WIDTH = *n*	cmd
		/OUTPUT = *outfile*	cmd
		/NOWARNINGS	cmd
Print: directs files to a line printer for printing. (PRINT/REMOTE prints a file located on a remote node.)	PRINT/ REMOTE	/DATA_TYPE = ASCII	file
		/TRANSFER_MODE = AUTO	file
		/TRANSFER_MODE = BLOCK	file
		/DATA_TYPE = IMAGE	file
		/NEW_VERSION	file
		/TRANSFER_MODE = RECORD	file
		/REPLACE	file
		/LOG	cmd
		/NOWARNINGS	cmd
Rename: changes the name of an existing file.	RENAME	/NEW_VERSION	file
		/LOG	cmd
		/NOWARNINGS	cmd
Set File: sets the owner and protection status for a file. (Set file is an alternative to the set protection operation.)	SET FILE	/LOG	cmd
		/NOWARNINGS	cmd
Set protection: sets or changes a file's protection status.	SET PROTECTION	/OWNER = [*uic*]	file
		/LOG	cmd
		/NOWARNINGS	cmd
Submit: submits a command file for execution. (SUBMIT/REMOTE submits a file located on a remote node for execution.)	SUBMIT/ REMOTE	/LOG	cmd
		/NOWARNINGS	cmd
Type: types the contents of a file to your terminal screen.	TYPE	/MACY11	file
		/LOG	cmd
		/NOWARNINGS	cmd

3.2.3 Definition of DCL Qualifiers

NOTE

The data type and transfer mode qualifiers have abbreviated synonyms that you can use to simplify command lines, as follows:

- /DATA_TYPE = ASCII: /ASCII
- /DATA_TYPE = IMAGE: /IMAGE
- /TRANSFER_MODE = AUTO: /AUTO
- /TRANSFER_MODE = BLOCK: /BLOCK
- /TRANSFER_MODE = RECORD: /RECORD

/DATA_TYPE = ASCII
(/ASCII)

The ASCII qualifier transfers files in ASCII record mode. For remote output files, NFT translates the records in the file into an appropriate format for the remote system. For local output files, NFT creates the file as a variable length format, implied CR/LF attribute file.

Use /DATA_TYPE = ASCII only for translation to ASCII. File transfers to remote nodes whose file systems are similar to RSX's (such as VMS) are more efficient in image block mode, which is the default.

/DATA_TYPE = IMAGE
(/IMAGE)

The image qualifier transfers files while retaining their current format and attributes. Some remote systems may not support the transfer, causing an error message to be displayed. If the file is ASCII, you can recover by using the /DATA_TYPE = ASCII qualifier. If the transfer was aborted due to unsupported file organization, you can retry the transfer using the /TRANSFER_MODE = BLOCK qualifier.

/TRANSFER_MODE = AUTO
(/AUTO)

The automatic transfer selection qualifier lets NFT select the file transfer mode. NFT will try to select either block mode or record mode to transfer a file, based on the remote FAL capabilities. Since automatic transfer selection is the default mode, you use this qualifier only when the command line contains qualifiers that specify different transfer modes.

/TRANSFER_MODE = BLOCK
(/BLOCK)

The block mode qualifier transfers files using 512.-byte blocks, which is more efficient than the usual record mode transfer. You can use block mode with any file organization type when transferring files to other Files–11 or RMS–11 systems that support block mode transfers, such as RSX, VMS, or the RSTS/E RMS FAL. Block mode transfers to and from foreign file systems may not always guarantee usable files, since all foreign systems do not store records compatibly.

/TRANSFER_MODE = RECORD
(/RECORD)

The record mode transfer qualifier transfers a file one record at a time, rather than by block. You would use this qualifier when copying to unit record equipment, for example.

/CONTIGUOUS

The contiguous output qualifier creates an output file with contiguous allocation.

/MACY11

The MACY11 qualifier stores a file in MACY11 format on a TOPS–10 or a TOPS–20 system.

/NEW_VERSION

The new version qualifier gives an output file the same or latest version number as the existing file, whichever is higher. However, this depends on the remote system's support of version numbers. For example, since RT–11 and RSTS/E systems do not support version numbers, they ignore this qualifier.

The /NEW_VERSION qualifier has no negative version. To specify that an output file retains the current version number, use the /REPLACE qualifier, as long as no other file with the same version number already exists.

/NO_SPAN

The no span blocks qualifier prevents an output file from spanning block boundaries.

/PROTECTION = *code*

The protection qualifier lets you set or change a file's protection status. File protection has four levels (SYSTEM:RWED, OWNER:RWED, GROUP:RWED, WORLD:RWED):

- SYSTEM: specifies access to the system UICs (UICs with group numbers of 10 octal or less).
- OWNER: specifies access to yourself.
- GROUP: specifies access to other members of your group.
- WORLD: specifies access to all other UICs.

At each level, the R, W, E, and D settings specify whether members of that level can read, write, extend, or delete a file. See Section 3.2.12 for a description of setting file protection status.

/OWNER = [*uic*]

The owner qualifier lets you specify an output file's owner.

/SUBMIT

The submit qualifier sends a file to a batch or indirect command file processor for execution.

/PRINT

The print qualifier queues a file for printing at a line printer.

/REPLACE

The replace qualifier transfers one or more files to a file whose file name, type, and version number already exist in a user file directory (UFD). The existing file is deleted, and a new file is created with the data from the input file(s). The new file keeps the existing file's name, type, and version number, but has a new file ID.

If an error occurs during file transfer, both the new output file and the file it is replacing will be lost.

/CONCATENATE

The concatenate qualifier merges two or more input files into one output file.

/LOG

The log qualifier displays the name of each file as NFT performs the operation on it.

/NOWARNINGS

The no warnings qualifier suppresses output messages that would otherwise appear on your terminal during NFT operations.

3.2.4 APPEND

The APPEND command appends the contents of one or more input files to the end of an existing output file. The output file retains its original attributes. However, if the input and output files have different attributes, the append operation may appear to succeed but produce an incorrect output file.

Format:

APPEND[/*qualifiers*] *infiles*[/*qualifiers*] *outfile*[/*qualifiers*]

where

outfile is the output file descriptor. You cannot use wildcards in the output file descriptor. The file and record attributes are taken from the existing file. You must specify the output file's name and file type.

infiles are one or more input file descriptors.

/qualifiers are one or more of the following qualifiers (defined in Section 3.2.3):

File Qualifiers	Command Qualifiers
/DATA_TYPE = ASCII	/LOG
/DATA_TYPE = IMAGE	/NOWARNINGS
/TRANSFER_MODE = AUTO	
/TRANSFER_MODE = BLOCK	
/TRANSFER_MODE = RECORD	
/MACY11	
/PROTECTION = *level*	
/OWNER = [*uic*]	
/PRINT	

Example 1:

```
$
->
```

In this example, the command verb has the /ASCII qualifier. Therefore, both input files in the command line (FIL2.DAT;1 located on node ESTER, and FIL3.DAT;1 located on node UTICA) are appended in ASCII record mode to FIL1.DAT;1 stored on DU1: on node LOWELL. The command qualifier /LOG displays the names of the two appended files as each append operation occurs.

Example 2:

```
$
->
```

In this example, the output file has the /ASCII qualifier. Therefore, both input files in the line (FIL2.DAT;1 located on node ESTER, and FIL3.DAT;1 located on node UTICA) are appended in ASCII record mode to FIL1.DAT;1 stored on DU1: on node LOWELL. The command qualifier /LOG displays the names of the two appended files as each append operation takes place.

Example 3:

```
$
->
```

In this example, the FIL2.DAT;1 input file has the /ASCII qualifier. Therefore, although both input files in the command line (FIL2.DAT;1 on node ESTER, and FIL3.DAT;1 on node UTICA) are appended to FIL1.DAT;1 stored on DU1: on node LOWELL, only FIL2.DAT;1 is transferred in ASCII record mode. The command qualifier /LOG displays the names of the two appended files as each append operation takes place.

3.2.5 COPY

To transfer files, you use the DCL COPY command. COPY transfers files in various ways:

- Single file transfer transfers a single file from one node to another.
- File transfer with concatenation transfers more than one input file to create a single output file. The input files can be from the same or different nodes.

  The attributes of the first input file in the command line become the attributes of the output file. If the input files have different attributes, the concatenation may appear to succeed, but the output file may not read correctly.
- Multiple file transfer transfers a number of input files to the same number of output files; each input file has a corresponding output file.

If an error occurs during transfer, NFT or the remote FAL attempts to delete the output file.

Format:

COPY[*/qualifiers*] *infiles*[*/qualifiers*] *outfile*[*/qualifiers*]

where

infiles are one or more input file descriptors.

/qualifiers are one or more of the following qualifiers (defined in Section 3.2.3):

File Qualifiers

/DATA_TYPE = ASCII
DATA_TYPE = IMAGE
/TRANSFER_MODE = AUTO
/TRANSFER_MODE = BLOCK
/TRANSFER_MODE = RECORD
/CONTIGUOUS
/MACY11
/NEW_VERSION
/NO_SPAN
/PROTECTION = *level*
/OWNER = [*uic*]
/PRINT
/REPLACE
/SUBMIT

Command Qualifiers

/CONCATENATE
/LOG
/NOWARNINGS

Example 1: Single file transfer

```
$
```

This example transfers THURS.TXT from remote node UTICA to the local node and renames it to THUR.TXT.

Example 2: Single file transfer between remote nodes

```
$
```

This example transfers THURS.TXT from node UTICA to node ALBANY and stores the file on DR5: under UFD [MIDLER]. On ALBANY, THURS.TXT will have the same or latest version number, whichever is higher.

Example 3: Single file transfer using a foreign file

```
$
```

This example transfers THURS.TXT in MACY11 format from remote node UTICA to the local TOPS–20 node and gives it the new name THUR.TXT.

Example 4: Transfer with file concatenation

```
$
```

This example transfers THURS.TXT and FRI.TXT from remote node UTICA to the local node and concatenates them to a single output file, WEEK.TXT.

Example 5: Multiple file transfer

```
$
_>
```

This example transfers THURS.TXT from node UTICA and FRI.TXT from node ALBANY to node ALEN. It stores them on DRA1: in the UFD [DATES]. Both files will have the same or latest version number, whichever is higher.

The version of THURS.TXT that this example transfers will replace any existing version of THURS.TXT on ALEN in DRA1:[DATES]. The existing file is deleted and a new one created with the data from the transferred file.

Example 6: Multiple file transfer using wildcards

```
$
_>
```

This example transfers the latest version of all .OBJ files and all versions of all files named STAFF. It transfers the files from DR5:[MIDLER] on node ALBANY to DB3:[HENDERSON] on node HAWAII. The resulting output files will have the same names as the input files. They will also have the same or latest version number, whichever is higher.

3.2.6 CREATE

The CREATE command creates a file that previously did not exist. You enter data at your terminal to create a file in ASCII record mode. The data that you enter is entered into the file until you type a CTRL/Z end-of-file indicator.

Format:

CREATE *outfile*[/*qualifiers*]

where

outfile is the name of the output file to be created.

/qualifiers are one or more of the following qualifiers (defined in Section 3.2.3):

File Qualifiers	Command Qualifier
/CONTIGUOUS	/LOG
/MACY11	
/NEW_VERSION	
/NO_SPAN	
/PROTECTION = *level*	
/OWNER = [*uic*]	
/PRINT	
/REPLACE	
/SUBMIT	

Example:

```
$
```

This example creates the output file, MYFILE.TXT, on node ALBANY.

3.2.7 DELETE

The DELETE command deletes one or more files that you specify. The files can be located on the same or different nodes.

Format:

DELETE[*/qualifiers*] *file-descriptors*[*/qualifiers*]

where

file-descriptors are one or more valid file descriptors, as defined in Section 3.1. To delete files on RSX nodes, you must specify the version number.

/qualifiers are one or both of the following qualifiers (defined in Section 3.2.3):

File Qualifiers	**Command Qualifiers**
(Not Applicable)	/LOG /NOWARNINGS

On RSX systems, if you use a wildcard (* or %) in the file name or type, you must specify the version number or use a wildcard; you cannot use 0 or −1.

Example:

```
$
```

This command line deletes the files TAX.LST;5 and BUDGET.LST;1.

3.2.8 DIRECTORY

Qualifiers

/FULL
/ATTRIBUTES
/BRIEF
/WIDTH = *n*
/OUTPUT = *outfile*

The DIRECTORY command displays a list of files and, optionally, information about the files. The displayed information varies according to how you enter the command:

- The DIRECTORY command, without qualifiers, lists the name, size, creation date, and optional contiguous and locked indicators for the files specified in the input file descriptor. The input file specifier consists of a remote node specification and associated file specification.

- The /FULL qualifier lists the name, size, creation date, optional contiguous and locked indicators, owner, protection, data last modified, and revision number for files on a remote node.

- The /ATTRIBUTES qualifier lists the attributes for each file in a directory listing. This display includes the full (/FU) directory information, plus the file organization, record format, and record attributes.

- The /BRIEF qualifier lists only the names of files located on a remote node.

Format:

```
DIRECTORY  [/ATTRIBUTES      ] infiles
           [/BRIEF           ]
           [/FULL            ]
           [/OUTPUT = outfile]
           [/WIDTH = n       ]
```

where

/OUTPUT = *outfile* is an optional file descriptor to which the directory listing will be written. The default output destination is TI:. The file descriptor consists of a node specification, two colons (::), and the associated file specification.

/WIDTH = *n* — specifies the width of the listing. The *n* value is a decimal number; the default value is 72. At least one field is always displayed per line, even if that field exceeds the specified maximum width. The /WIDTH = *n* qualifier cannot be used in conjunction with the /ATTRIBUTES or /BRIEF qualifiers.

infiles — are input file descriptors for each remote file for which you want a directory listing. The file descriptor consists of a remote node specification, two colons (::), and associated file specification. If you omit a file name or type in the descriptor, the directory listing will display information on all files in the specified directory.

Example 1: A DIRECTORY listing

The following example displays on TI: a directory of DB0:[WHITMAN] on node WASH:

```
$

Directory DBO:[WHITMAN]
7-AUG-86 22:06

REMINDER.CMD;1               3.  07-AUG-86 22:02
MASTERINI.CMD;1             11.  07-AUG-86 22:02
TELI.CMD;1                   2.  07-AUG-86 22:02
TESTFILE.COM;1               0.  07-AUG-86 22:02
LOGIN.CMD;1                  1.  07-AUG-86 22:02
PEOPLE.LST;1                 1.  07-AUG-86 22:02
NULLTEST.LST;1               2.  07-AUG-86 22:02
COPY.LST;1                   5.  07-AUG-86 22:02
NFTTEST.LOG;1               13.  07-AUG-86 22:04
TSETCMP.LOG;1               10.  07-AUG-86 22:04
SETCMP.LOG;1                 9.  07-AUG-86 22:04
```

Example 2: A DIRECTORY/FULL listing

The following example displays on TI: a directory of all .LST files located on DB0:[200,200] on node WASH:

```
$

Directory DRO:[200,200]
7-AUG-86 22:07

PEOPLE.LST;1    (40345,1)    1./1.        07-AUG-86 22:02 [7,132]  [RWED,RWED,RWED,R]
NULLTEST.LST;1  (40356,1)    2./2.        07-AUG-86 22:02 [7,132]  [RWED,RWED,RWED,R]
COPY.LST;1      (40375,1)    5./5.        07-AUG-86 22:02 [7,132]  [RWED,RWED,RWED,R]
```

Example 3: A DIRECTORY/ATTRIBUTES listing

This example writes the attributes of the files LOGIN.CMD and TESTFILE.CMD to the file DIR.LST. LOGIN.CMD and TESTFILE.CMD are located on DB0:[WHITMAN] on node WASH.

```
$
->
```

The information looks like the following:

```
DB0:[WHITMAN]LOGIN.CMD;1
                        Size:    1./1.          Created: 7-AUG-1984 22:02
                        Owner:   [WHITMAN]      Revised: 7-AUG-1984 22:02
                        File ID: (40320,1,0)    Expires: <none-specified>
File protection:        System:RWED, Owner:RWED, Group:RWED, World:R
File organization:      Sequential
File attributes:        Allocation=1, Extended=0
Record format:          Variable length, no maximum defined
Record attributes:      Carriage return

DB0:[200,200]TESTFILE.CMD;1
                        Size:    0./0.          Created: 7-AUG-1984 22:02
                        Owner:   [007,132]      Revised: 7-AUG-1984 22:02
                        File ID: (40302,1,0)    Expires: <none-specified>
File protection:        System:RWED, Owner:RWED, Group:RWED, World:R
File organization:      Sequential
File attributes:        Allocation=0, Extended=0
Record format:          Variable length, no maximum defined
Record attributes:      Carriage return
```

Example 4: A DIRECTORY/BRIEF listing

The next example displays on TI: a directory of all .LOG files located on DR5:[305,344] on node BOSTON:

```
$

Directory DR5:[305,344]

NFTTEST.LOG;1
TSETCMP.LOG;1
SETCMP.LOG;1
```

3.2.9 PRINT/REMOTE

The PRINT/REMOTE command prints one or more remote files on the remote node's line printer.

To copy a file to a remote node for printing, you must use COPY/PRINT.

Format:

PRINT/REMOTE [*/qualifiers*] *infiles*[*/qualifiers*]

where

/REMOTE specifies that the printing takes place on a remote node.

infiles are file descriptors of one or more files to print.

/qualifiers are one or more of the following qualifiers (defined in Section 3.2.3):

File Qualifiers	Command Qualifier
/DATA_TYPE = ASCII /DATA_TYPE = IMAGE	/LOG

Example:

```
$
```

In this example, one copy each of file COLD.FIL;1 and file SNOW.FIL;1 is spooled to the line printer on node HOME. The output file names default to the input file names.

3.2.10 RENAME

The RENAME command changes the names of files. You specify the existing names as the *infiles* and the new names as the *outfiles*. DCL deletes each input file and enters each output file into the appropriate directory. The rename operation does not transfer data across a network; both directories must be on the same node and device. Access control information is therefore necessary for the *infiles* only. By renaming files, you can also move them to a different directory; in that case, you must have access to the new directory.

Some systems do not support the Data Access Protocol (DAP) rename operation. Refer to the individual system documentation for more information.

Format:

RENAME[/*qualifiers*] *infiles*[/*qualifiers*] *outfile*[/*qualifiers*]

where

outfile is the file specification of the new file. See Section 3.1.2 for a description of file specification format. Each file name, type, and version number can be explicit, wildcard (*), or defaulted (null), with two restrictions:

- If there is a wildcard (*) in the file name or type, the version number must be explicit or wildcard (that is, it cannot be 0 or −1).
- You cannot use partial field or single character wildcard specifiers for *outfile*.

  When a field is wildcard or defaulted, NFT uses the corresponding field of the input file. You can therefore change one or more fields while preserving the others. See Section 3.1.2.1 for more information on wildcard specifications.

infiles are one or more file descriptors of files to rename.

/qualifiers are one or more of the following qualifiers (defined in Section 3.2.3):

File Qualifier	**Command Qualifiers**
/NEW_VERSION	/LOG /NOWARNINGS

Example:

```
$
```

This example renames all .MAR and .MAL files located on node WASH::DB0:[200,200], to .MAC files. Note that a space must separate the input file(s) from the output file(s).

3.2.11 SET FILE

The SET FILE command lets you set the owner and protection for a file. For more information on file protection, refer to the SET PROTECTION command, which follows.

Format:

SET FILE[*/qualifiers*] *infiles*[*/qualifiers*]- RET

__> [/PROTECTION = *level*] [/OWNER = [*uic*]

where

infiles are one or more input file specifications for which you specify protection and/or owner values.

/PROTECTION = *level* is the qualifier that gives the new protection level.

level File protection has four levels: SYSTEM:, OWNER:, GROUP:, and WORLD:. At each level, you specify R, W, E, and D to let members of the group read, write, extend, and delete the file(s). See Section 3.2.12 for more information.

/OWNER = [*uic*] is the UIC of the file's owner, in the form [*g*,*m*].

/qualifiers are one or both of the following qualifiers (defined in Section 3.2.3):

File Qualifiers	Command Qualifiers
(Not Applicable)	/LOG /NOWARNINGS

Example:

```
$
->
```

This command sets GROUP protection on MY.FIL to READ and WRITE. The other protection levels, SYSTEM, OWNER, and WORLD, remain unchanged. This command also sets the owner of HIS.FIL to [MIDLER].

3.2.12 SET PROTECTION

The SET PROTECTION command sets or changes a file's protection status.

File protection has four levels:

- SYSTEM: specifies access to the system UICs (UICs with group numbers of 10 octal or less).

- OWNER: specifies access to yourself.

- GROUP: specifies access to other members of your group.

- WORLD: specifies access to all other UICs.

At each level, you use R, W, E, and D to specify the following protection settings:

- READ: members of the group or their tasks can read, copy, print, type the file, or run it, if it is a task image.

- WRITE: members of the group or their tasks can add new data to the file by writing to it.

- EXTEND: members of the group or their tasks can increase the amount of disk space allocated to the file.

- DELETE: members of the group or their tasks can delete the file.

These protection settings apply to all input files in a command line, unless you override them for specific files.

To change a file's protection status, use SET PROTECTION with any or all of the four protection level qualifiers (SYSTEM:RWED, OWNER:RWED, GROUP:RWED, WORLD:RWED). The system default protection code is SYSTEM:RWED,OWNER:RWED,GROUP:RWED,WORLD:R.

You can change a file's protection status if you are the file owner or have a system level UIC, even without read or write access. You cannot change a file's protection level if you are in the group or world category for the file owner's UIC. You can display the protection attributes if you have read access to the file, however.

Format:

SET PROTECTION[[=](*code*)] *file-descriptors*
[/*qualifiers*][/OWNER = [*uic*]]

where

code specifies which user categories can access a file and what each user category can do with the file. For any or all of the SYSTEM, OWNER, GROUP, and WORLD levels, you specify any combination (or none) of the R, W, E, and D privileges. Parentheses are required around the code. The format for the code is (*level:setting*...).

file-descriptors specify all files to which the protection status applies.

/OWNER = [*uic*] sets the owner of a file to a specified UIC, in the form: [*g*,*m*].

/*qualifiers* are one or both of the following qualifiers (defined in Section 3.2.3):

File Qualifiers	**Command Qualifiers**
(Not Applicable)	/LOG /NOWARNINGS

Example:

```
$
->
```

The SET PROTECTION command in this example gives members of the system, world, and group levels read, write, extend, and delete access to file ALIAS.CMD, while it gives the owner just read, write, and delete access.

3.2.13 SUBMIT/REMOTE

The SUBMIT command executes the contents of a command file located on a remote node. SUBMIT requires the /REMOTE qualifier when the file specification includes a remote node name.

For RSX systems, the command file can be submitted to either a command file processor or a batch file processor, depending on the NETGEN generation of the destination node's command file submission task (MCM).

To execute a command file that is not already located on the remote node, you use the COPY/SUBMIT command. COPY/SUBMIT copies the command file to the remote node, and then submits it for execution. See Section 3.2.5 for a description of the copy operation and /SUBMIT qualifier.

Successful completion of the submit operation indicates that the execution request was successfully given to the remote MCM, not that the batch or command file executed successfully.

Format:

SUBMIT/REMOTE [*/qualifiers*] *command-files*

where

/REMOTE — is the qualifier that specifies remote execution.

command-files — are the output file descriptors for each remote command file to execute. Command files are not deleted after execution.

/qualifiers — are one or both of the following qualifiers (defined in Section 3.2.3):

File Qualifiers	**Command Qualifiers**
(Not Applicable)	/LOG /NOWARNINGS

Example:

```
$
```

In this example, SUBMIT/REMOTE submits the command file NAMES.CMD, located on node WASH, for execution. The /NOWARNINGS qualifier suppresses the display of any output messages to your terminal.

3.2.14 TYPE

The TYPE command displays the contents of a file on your screen.

Format:

TYPE[/*qualifiers*] *infiles*[/*qualifiers*]

where

infiles are one or more input file descriptors for files to display.

/qualifiers are one or more of the following qualifiers (defined in Section 3.2.3):

File Qualifier	Command Qualifiers
/MACY11	/LOG /NOWARNINGS

Example:

```
$
```

This command displays the contents of FIL2.DAT, located on node WASH, at your terminal.

3.3 NFT's Exit Status

When NFT exits, it returns a status code that may be useful

- in an indirect command file
- in a batch stream job (for RSX-11M-PLUS systems only)
- when NFT was spawned by or is connected to a parent task

NFT can issue two exit status codes:

Status	Code
NFT exited without error.	1 (EX$SUC)
NFT exited with an error.	2 (EX$ERR)

Status code 2 (EX$ERR) is returned if you make a syntax error or an operational error. Also, if any command in an indirect command file fails, an exit with status code 2 (EX$ERR) is issued when NFT exits.

You can test the status code returned by NFT and, on the basis of the results, specify alternative procedures as shown in the following examples.

Example 1: Indirect command file

```
.IF <EXSTAT> = 2 .GOTO JUNO
```

Example 2: Batch job

```
$IF ERROR = 2 THEN STOP
```

Consult the operator's procedures manual for your system for additional information about testing status in batch stream jobs and indirect command files. For information on testing return status in a parent task, refer to your system's executive reference manual.

For information on NFT error messages, refer to Appendix A.

4 The Network File Transfer (NFT) Utility

The Network File Transfer (NFT) utility lets you perform file operations on accessible network nodes from the MCR command line interpreter. If you have DECnet–11M–PLUS or DECnet–Micro/RSX, you can perform file operations from either MCR or DCL. For information on performing file operations from DCL, refer to Chapter 3.

This chapter is organized as follows:

- Section 4.1 describes how you specify files for NFT, use wildcards in file specifications, and specify files on non-RSX nodes. It also describes the NFT file descriptor defaults.

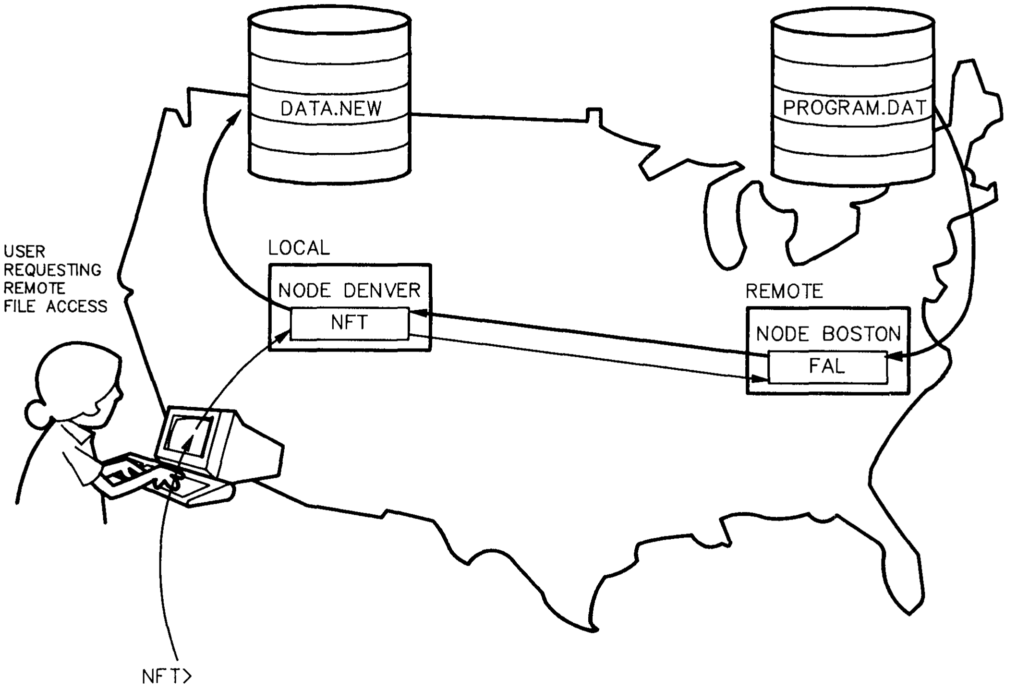
- Section 4.2 describes NFT's commands for file operations. Table 4–2 summarizes the NFT operations.

In DECnet–RSX systems, the NFT server task is the File Access Listener (FAL). FAL receives and handles incoming requests for file operations from NFT (see Figure 4–1).

For information on cross-system file operations, refer to Section 4.1.2.2 and to the *DECnet–RSX Release Notes*.

Figure 4–1 shows a user requesting a copy operation. The operation copies the remote file PROGRAM.DAT from node BOSTON to local file DATA.NEW. The user includes, as access control information for the remote system, the user name Silver and password PIANO.

Figure 4–1: The NFT Utility: File Transfer



LKG–0137

4.1 NFT File Descriptor Syntax

NFT commands operate on files. You give NFT file descriptors that identify files on a local or remote network node. A file descriptor consists of a node and file specification. The format for a file descriptor is:

[*node-specification*::][*file-specification*]

4.1.1 Node Specification

A node specification consists of a node name, followed by two colons. Optionally, you can include access control information for the node. The format is

node-name[*access-control*]::

where

node-name is a 1- to 6-alphanumeric character name that includes at least 1 alphabetic character. If a node name is an alias that includes access control information, you can omit the access control fields. For information on aliases, see Section 2.7.

access-control is a user identification, password, and account specification. You append the access control information to a node name to gain access to its file system. You can use either of the following formats to specify access control information:

/*user-id*/*passwd*/*accnt*::

or,

"*user-id passwd accnt*"::

where

user-id is a 1- to 39-character string that identifies the user at the remote system.

passwd is a 1- to 39-character password for gaining access to the remote file system.

accnt is a 1- to 39-character string that specifies an account at the remote system. RSX systems do not use this field.

To pass access control information to case sensitive systems such as ULTRIX–32, you must use the first format, since the second format passes access control information in uppercase. The embedded spaces in the second format are mandatory.

Different remote systems interpret the access control fields and the access control mechanism differently. If the remote system is a DECnet–RSX node that supports access control verification, the *user-id* field specifies a valid account by last name or UIC. The *passwd* string must be the password for that account, and the *accnt* field is ignored. For more information on access control, see the *DECnet–RSX Network Management Concepts and Procedures* manual.

If the remote node does not support access control verification, its File Access Listener (FAL) permits privileged and nonprivileged access control. For privileged access, use the privileged password specified for FAL during generation of the remote node. The default password is PRIV. Use any desired UIC for the user ID. For nonprivileged access, use the guest UIC and nonprivileged password specified for FAL during generation of the remote node. The default nonprivileged user ID is [200,200] and the default password is GUEST. The *accnt* field can be omitted for both privileged and nonprivileged access.

If you have a proxy account on a remote node, you can omit the access control information. If you have multiple proxy accounts, you can include just the user ID in the access control information to identify the account that contains the file.

Examples of Node Specifications:

Examples of valid node specifications follow. You can omit missing fields from the right. For example, you can enter the node specification QUEBEC/[310,2]// as QUEBEC/[310,2]. To denote a missing field on the left, use a slash or blank, depending on the format you use, as in the fourth example.

NOTE

> Brackets ([ ]) are optional around a user ID that has the form of a UIC, as in the last two examples.

```
NODE4/[7,7]/SECRET/ACCNT::
UTICA"5,10 LEFT"::
BOSS/EVERY/ONE::
BOSTON///ACCNTNE::
NODE1"RMES"::
BUTTE/1,1/PRIV::
PEORIA/200,200/GUEST::
```

4.1.2 File Specification

A file specification identifies the local or remote file on which to perform an operation. You can use many foreign file specifications (see Section 4.1.2.2.). RSX file specifications have the following format:

dev:[*ufd*] *file-name.type;ver*

For more information on RSX files, refer to the *RSX–11 Utilities Manual*. For a description of wildcard characters within file specifiers, see Section 4.1.2.1. For information on specifying file names that do not conform to RSX conventions, see Section 4.1.2.2.

A file in an NFT command can be an input (source) or output (destination) file. Input and output file specifications take different positions in the command string.The node name tells NFT whether the file is local or remote. NFT files can be:

- Local output files
- Remote output files
- Local input files
- Remote input files

Local or remote input files provide the data for an NFT operation; local or remote output files receive data from an NFT operation.

Table 4–1 summarizes defaults for DECnet–RSX NFT file descriptors. You can change these default values by using the set default operation (/DF).

Table 4–1: DECnet–RSX NFT File Descriptor Defaults

Field	Conditions	Default Value
node-name	The file is the first or only file in the input list or the output list.	Local node name.
	The file is a subsequent file in a given list.	Preceding node name specified in the list (including access control information).
user-id	The *user-id* has been assigned using an alias node name. (See Chapter 2 for information on the NCP SET ALIAS command.)	Value of *user-id* specified with the alias node name.
	A node name with access control information has been specified to NFT using the /DF (default) switch.	Value of *user-id* given for the node name.

(continued on next page)

Table 4-1 (Cont.): DECnet-RSX NFT File Descriptor Defaults

Field	Conditions	Default Value
passwd	The *passwd* value has been preassigned using an alias node name. (See Chapter 2 for information on the NCP SET ALIAS command.)	Value of *passwd* specified with the alias node name.
	A node name with access control information has been specified to NFT using the /DF (default) switch.	Value of *passwd* given for the node name.
accnt	The accounting number has been preassigned using an alias node name. (See Chapter 2 for information on the NCP SET ALIAS command.)	Value of *accnt* specified with the alias node name.
	A node name with access control information has been specified to NFT using the /DF (default) switch.	Value of *accnt* given for the node name.
dev	The file is the first or only file in an input list or an output list associated with a particular node.	SY: device associated with the access control given with the specified node name. If no node name is given, it defaults to the user's current SY: device.
	The file is a subsequent file in a given list, as previously defined.	Preceding device specified in the list.
	A device has been specified to NFT using the /DF (default) switch.	Device specified in the /DF command.
ufd	The file is the first or only file in an input list or an output list associated with a particular node.	Directory associated with the access control given with the specified node name. If no node name is given, it defaults to the user's current directory. (RSX-11M-PLUS and Micro/RSX systems now support named directories.)
file-name	The file is the first or only file in the input list.	None.
	The file is a subsequent file in the input list.	Preceding file name specified in the list.
	The file is an output file.	Name of the corresponding input file.

Table 4-1 (Cont.): DECnet-RSX NFT File Descriptor Defaults

Field	Conditions	Default Value
type	The file is the first or only file in the input list.	None.
	The file is a subsequent file in the input list.	Previous type specified in the list.
	The file is an output file.	Type of corresponding input file.
ver	The version number is omitted for any input file.	Highest version of file.
	The version number is omitted for an output file.	If a version number was specified for the input file (either wild or explicit), the output file will have the corresponding version number. If a version number was not specified for the input file, then no version is used on the output file. On RSX systems this results in the highest version being used.

To use a format that the table does not describe, enclose the foreign file directory and file specification with double quotation marks, as in ''*file-specification*''. NFT will transmit the file specification to the foreign node without checking its syntax or applying defaults for missing fields. File specifications within double quotation marks are not used in determining default values for output files.

Always use quotation marks to enclose file specifications when you transfer files to and from case sensitive systems such as ULTRIX–32.

The conventions of foreign operating systems restrict your use of wildcards in file specifications.The next section describes how you use wildcards in RSX–11 file specifications.

4.1.2.1 Wildcards in NFT File Specifications

NFT lets you use wildcards (asterisks and/or percent signs) in file specifications for existing local or remote files on DECnet–RSX systems. To use a wildcard specifier that NFT does not accept in a remote file specification, enclose it in quotation marks.

An asterisk (*) in a field means that NFT will take any value within that field. These are the wildcard field specifiers:

[*,*m*] *file-name.type;ver*	takes all group numbers.
[*g*, *] *file-name.type;ver*	takes all member numbers.
[*] *file-name.type;ver*	takes all named directories.
[*g,m*] *.*type;ver*	takes all file names.
[*g,m*] *file-name.* *;*ver*	takes all file types.
[*g,m*] *file-name.type;* *	takes all version numbers.

NFT also accepts wildcard characters for part of a field in the file name and file type, except when you are renaming files. The wildcard rules are as follows:

- An asterisk in a portion of a field means that that portion can be replaced with any characters of any length (including none). For example,

  SOME*.TYP will accept: SOME.TYP
  SOMETHG.TYP
  SOMEBODY.TYP
  SOMEBZ.TYP

- A percent sign (%) in a character position accepts any one character (except nulls) in that position. RSTS/E uses the question mark (?) rather than the percent sign as a single position wildcard. For example,

  MAIL%.MAI will accept: MAIL1.MAI
  MAIL2.MAI
  MAIL3.MAI
  MAIL4.MAI

  IMV%I%%.TXT will accept: IMV1I22.TXT
  IMV2I30.TXT
  IMV3I45.TXT

You can combine multiple wildcards in a single file specifier, as in these examples:

.;* means all versions of all files under the same UIC.

TEST.*;* means all versions and all types of files named TEST.

TEST*.DAT;* means all versions of files having names beginning with TEST and being of type DAT.

TEST%.* means the most recent version of all file types for files with 5-character names beginning with TEST.

Sample command line:

```
NFT>
```

In this example, NFT transfers to node QUEENS all files

- Located on device DM1: on the local node, and
- Listed under any group number for UFD member 10, and
- With a 5-character file name beginning with the characters FIL, and
- With the file type MAC

4.1.2.2 Foreign File Formats

A foreign file resides on a non-RSX node. A file operation that includes a foreign file must use a file specification that conforms to the foreign system. NFT can parse the following foreign file specification types:

Directory formats:

[*ufd*]	IAS
[*directory*]	VMS
[*ppn*] or (*ppn*)	RSTS/E
<*directory*>	TOPS–20, VMS
(not applicable)	RT–11
/*pathname*	ULTRIX–32

File specification formats:

file-name.type;ver	IAS, VMS
file-name.type.ver	TOPS–20, VMS
file-name.type	RT–11, RSTS/E
/pathname/file-name	ULTRIX–32

NOTE

> An ULTRIX–32 directory is specified by a path name, such as /usr/users/vaughan, where "/" denotes the root.
>
> Enclose an ULTRIX–32 file specification in quotation marks to pass it to a remote system in a case sensitive manner.

These are examples of foreign file specifications:

VMS:

```
SYS$SYSDISK:[COTTON.DAP]DAPV70.MEM;69
```

TOPS–20:

```
PS:[COTTON]MAIL.TXT.1439
```

RSTS/E:

```
SY:(1,4)SYSTEM.HLP
```

RT–11:

```
DU:TECO.SAV
```

ULTRIX–32:

```
"/usr/users/vaughan/main.c"
```

4.1.3 Logical Names in File Descriptors

RSX-11M-PLUS and Micro/RSX systems have extended logical name support. You can use logical names for the node and file specification in network file specifiers. For incoming remote files, you can use only global logicals; for outgoing local files, you can use both local and global logicals.

Use the MCR commands DFL (DEFINE LOGICALS) and ASN (ASSIGN) to define, delete, or display logical names. Using DFL, the following example equates a logical name, ADDR, to a node, disk, and directory name. Then it uses the logical name ADDR to simplify the file specification in displaying a file.

```
>
>
```

Note that the logical name is always at the leftmost part of the file specification.

Certain network commands require that you supply access control information. A logical name definition can contain access control information, but consider the following suggestions for protecting security:

- Do not put a logical name definition that contains access control information in a command file.

- Use aliases in combination with logical names. Once you define an alias, the access control information is protected. The NCP command SHOW KNOWN ALIASES displays information about the alias, but does not echo the access control information on a terminal. In contrast, the MCR commands, ASN and DFL, display the entire equivalence string.

  To create a secure logical name, first put the node name and access control information in an alias. Then use the alias when defining the logical name. Any DFL or ASN command will then show the alias instead of the node name and access control information.

Suppose you want to do NFT operations that require access to user Arnold's account on node UTICA. The following example uses NCP to set the alias REM equal to node UTICA, user ID ARNOLD, and password SECRET. Then it substitutes the alias for those elements in an MCR DFL command. DFL equates logical name STAFF to disk DB1 and directory [275,10] on UTICA. The next command then uses logical name STAFF for all elements of the file specification other than the file name.

```
>
>
>
```

Note that the ASN and DFL commands have different syntax. For more information on using logical names, refer to the *RSX–11M/M–PLUS MCR Operations Manual*.

4.2 NFT Operations for MCR Users

To instruct NFT to perform a file operation, you use a switch. Each NFT primary switch specifies a file operation. For example, the primary switch /AP specifies the append operation, which appends the contents of one file to the contents of another. Some file operations also have qualifying switches that further define the action of the primary switch. For example, if you use the /AS qualifying switch with /AP, the /AS qualifying switch causes NFT to append the file in ASCII record mode. For a summary of the NFT operations from MCR, refer to Table 4–2. For detailed information on each operation, refer to Sections 4.2.4 through 4.2.15.

4.2.1 Specifying MCR Switches in a Command Line

Primary switches can be anywhere in a command line.

Qualifying switches can apply to the operation or to specified files. Depending on what the switch qualifies, you place it on the command line as follows:

1. Operation qualifying switches apply to an NFT operation. They can be anywhere on the command line.

2. File qualifying switches apply to one or more files specified on the command line. On an input file specification, a file qualifying switch applies only to that file. On an output file, however, the switch becomes global and applies to all input files in the command, unless you override the switch for specific files.

4.2.2 NFT Command Line Continuer

To use multiple input lines to specify an NFT operation, enter a hyphen (-) and press the RETURN key anywhere in the command line. NFT will then prompt you for the remainder of the command. You can use any number of lines to enter a command.

Example:

```
NFT>
NFT>
```

Table 4-2: NFT Operations for MCR Users

NFT Operation	NFT Primary Switch	NFT Qualifying Switches	Scope of Qualifier
Append: appends files to the end of an existing file.	/AP	/AS	file
		/AX	file
		/BK	file
		/IM	file
		/PR/SY/OW/GR/WO	file
		/PR/FO:[*uic*]	file
		/RAT:MACY11	file
		/RC	file
		/SP	file
		/LO	cmd
		/NM	cmd
Copy: transfers one or more existing files to one or more files on the local or remote node.	no switch	/AS file	file
		/AX	file
		/BK	file
		/CO	file
		/IM	file
		/NV	file
		/PR/SY/OW/GR/WO	file
		/PR/FO:[*uic*]	file
		/RAT:MACY11	file
		/RAT:NOSPAN	file
		/RC	file
		/SB	file
		/SP	file
		/SU	file
		/LO	cmd
		/NM	cmd
		/ME	cmd
Delete: deletes one or more files.	/DE	/LO	cmd
		/NM	cmd
Directory: lists a directory of one or more files.	/LI /BR /FU /AT		
Execute: executes a command file stored on a remote or local node.	/EX	/LO	cmd
		/NM	cmd
Help: displays a descriptive message about switches and commands.	/HE HELP [*subject*]		
Identify: displays the version of NFT being used.	/ID		

(continued on next page)

Table 4-2 (Cont.): NFT Operations for MCR Users

NFT Operation	NFT Primary Switch	NFT Qualifying Switches	Scope of Qualifier
Rename: changes the name of an existing file.	/RE	/NV /LO /NM	file cmd cmd
Set defaults: sets certain default vales for input and output files.	/DF		
Set protection: sets or changes a file's protection status depending on the qualifying switch (/SY:*p*, /OW:*p*, /GR:*p*, /WO:*p*) and its setting (*p* = R, W, E, D). The /FO qualifying switch specifies the file owner.	/PR[:*n*]	/SY /OW /GR /WO /FO:[*uic*] /LO /NM	file file file file file cmd cmd
Spool: sends files to a line printer for printing.	/SP	/AS /AX /BK /IM /NV /RC /SU /LO /NM	file file file file file file file cmd cmd
Submit: copies a command file to a local or remote node, executes the file, and then deletes it.	/SB	/AS /AX /BK /IM /NV /RC /LO /NM	file file file file file file cmd cmd

4.2.3 The NFT Qualifying Switches

Switch	Description
/AS	The ASCII switch transfers files in ASCII record mode. For remote output files, NFT translates the records in the file into an appropriate format for the remote system. For local output files, NFT creates the file as a variable length format, implied CR/LF attribute file. Use the /AS switch only if you want translation. File transfers are more efficient in the default image block mode if the remote node has a file system similar to RSX (VMS, for example, does).
/AX	The automatic transfer selection switch causes NFT to use its default algorithms in selecting the file transfer mode. NFT will try to select either block or record mode, based on the remote FAL capabilities. Since automatic transfer selection is the default mode, you need /AX only when other switches on the same command line specify a different transfer mode.
/BK	The block mode switch transfers files using 512.-byte blocks. This is a more efficient method than the usual record mode transfer. You can use block mode with any file organization type on transfers to other Files–11 or RMS–11 systems that support block mode transfers, such as RSX, VMS, or the RSTS/E RMS FAL. When using block mode to transfer files to or from a foreign file system, there is no guarantee that the files will be usable, since all systems do not store records compatibly.
/CO	The contiguous output switch creates an output file with contiguous allocation.
/IM	The image switch transfers files while retaining their current format and attributes. Some remote systems may not support the transfer, causing an error message to be displayed. If the file is ASCII text, you can recover by using the /AS switch. If the transfer was aborted due to unsupported file organization, you can retry the transfer using the /BK switch.
/NV	The new version switch specifies that an output file will have the same or latest version number, whichever is higher. However, the effect of the /NV switch depends on the remote system's support of version numbers. For example, RT–11 and RSTS/E do not support version numbers, and therefore would ignore this switch. The negative version of the /NV switch (/–NV) specifies that an output file retains the current version number. If an existing file has the same version number, an error can occur.

(continued on next page)

Switch	Description
/PR /SY/OW/GR/WO	The protection switch (/PR) lets you set or change a file's protection status. File protection has four levels (/SY:*p*, /OW:*p*, /GR:*p*, /WO:*p*): /SY: (System) gives access to the system UICs (UICs with group numbers of 10 octal or less). /OW: (Owner) gives access to yourself. /GR: (Group) gives access to other members of your group. /WO: (World) gives access to all UICs. The *p* variable represents the protection settings R, W, E, and D. The R, W, E, and D settings specify whether members of each level can read, write, extend, or delete a file.
/FO:[*uic*]	The file owner switch lets you specify the owner of an output file.
/RAT:MACY11	The MACY11 switch stores a file in MACY11 format on a TOPS-10 or TOPS-20 system.
/RAT:NOSPAN	The no span blocks switch prevents the output file from spanning block boundaries.
/RC	The record mode transfer switch transfers a file one record at a time, instead of in block mode. You use the /RC switch to copy to unit record equipment or override NFT's default transfer mode selection.
/SB	The submit switch submits a file to a batch or indirect command file processor on a local or remote node, executes the file, and then deletes it.
/SP	The spool switch sends a file to a line printer on a local or remote node for printing. When a file is spooled to a remote node, its format must be compatible with the remote system.
/SU	The supersede switch transfers one or more files to a file whose file name, type, and version number already exist in a user file directory (UFD). The existing file is deleted, and a new file is created with the data from the input file(s). The new file retains the current file name, type, and version number but has a new file ID. An error during the file transfer causes both the new output file and the file it is replacing to be lost.
/LO	The log switch displays the names of affected files as NFT performs an operation, such as copy, delete, and so forth.

Switch	Description
/NM	The no messages switch suppresses output messages that your terminal would otherwise display during NFT operations.
/ME	The merge switch merges two or more input files into one output file. If you use wildcards in one or more fields of the output file specification, the resulting output file specification will contain information that corresponds to those fields in the first input file specification in the command line.

4.2.4 Append (/AP)

The append operation appends the contents of one or more input files to the end of an existing output file. The output file retains its original attributes. The input and output file attributes should be the same; if they are different, the operation may appear to succeed, but produce an incorrect output file.

Format:

outfile[/*sw*] = *infiles*[/*sw*]/AP

where

outfile is the output file descriptor. The output file descriptor cannot have wildcards. The file and record attributes are taken from the existing file. You must specify the the output file's file name and file type.

infiles are one or more input file descriptors.

/sw may include one or more of the following qualifying switches (defined in Section 4.2.3):

File Qualifying Switches	Command Qualifying Switches
/AS	/LO
/AX	/NM
/BK	
/IM	
/PR/SY/OW/GR/WO	
/PR/FO:[*uic*]	
/RAT:MACY11	
/RC	
/SP	

Example:

```
NFT>
NFT>
```

In this example, FIL1.DAT;1, which is stored on DU1: at node LOWELL, is opened. FIL2.DAT;1, located at node ESTER, and FIL3.DAT;1, located on node UTICA, are then appended to FIL1.DAT;1 in ASCII record mode.

4.2.5 Copy

The NFT copy operation, which does not require a primary switch, transfers files in these ways:

- Single file copy transfers a single file from one node to another.
- Copy with concatenation transfers more than one input file to create a single output file. The files can be from the same or different nodes.

  The attributes of the first input file in the command line become the attributes of the output file. If the attributes of the input files differ, the concatenation may appear to succeed but may produce an output file that is not correctly readable.
- Multiple file copy transfers a number of input files to the same number of output files so that each input file has a corresponding output file.

If an error occurs during transfer, NFT or the remote FAL (File Access Listener) attempts to delete the output file.

Format:

[*outfile*][/*sw*] = *infiles*[/*sw*]

where

outfile is an optional output file descriptor to which the input files are copied.

infiles are one or more input file descriptors.

/sw may include one or more of the following qualifying switches (defined in Section 4.2.3):

File Qualifying Switches	Command Qualifying Switches
/AS	/LO
/AX	/NM
/BK	/ME
/CO	
/IM	
/NV	
/PR/SY/OW/GR/WO	
/PR/FO:[*uic*]	

File Qualifying Switches	Command Qualifying Switches
/RAT:MACY11	
/RAT:NOSPAN	
/RC	
/SB	
/SP	
/SU	

Example 1: Single file copy

```
NFT>
```

In this example, the file THURS.TXT, located on remote RSX node UTICA, is copied to the local node. Because no output file name is specified, the output file name will be THURS.TXT. Access control information is not specified for either the local node or node UTICA. Therefore, missing fields are assumed to be defined either in the respective default node names or in alias node names.

Example 2: Single file copy between remote nodes

```
NFT>
NFT>
```

In this example, the file MYSTIC.MAC;2, located on NODEY, an RSX–11 node, on the system device under UFD [30,60] is copied to NODEX, also an RSX–11 node. The file is renamed MAGIC.MAC;3, and is stored on DU0: in the UFD [VAUGHAN]. Access control information is passed in the input file specifier to allow access to MYSTIC.MAC. Access control information is required for the creation of output file MAGIC.MAC. (Note the use of the command line continuer for the additional input line.)

Example 3: Single file copy using a foreign file

```
NFT>
```

In this example, the foreign file BOOTS.EYE, located on NODEB, a VMS node, is sent to NODEA, an RSX–11 node. On NODEA, the file is named MARY.MAC (latest version) and is stored on NODEA's system device (SY0:) under the current UIC. Because access control information is not specified for either NODEA or NODEB, the default values for these missing fields are assumed to be defined for each node in an alias node name or to be previously defined by the set default operation (/DF).

In addition to the access control information, the *dev*, *ufd,* and *ver* arguments are also omitted for NODEB. Their default values are determined by the syntax conventions of NODEB's operating system. Keep in mind that NODEB is a foreign node, and may not require some of the file descriptor information. If you omit any required argument in the command line and do not adhere to default regulations, however, you will get an error message. Refer to Section 4.1.2.2 for information on the required formats for foreign file specification.

Example 4: Copy with file concatenation

```
NFT>
NFT>
```

In this example, SAM.LIT and ONE.WAY located on RSX NODEY, are copied to RSX NODEA. The two input files are concatenated into a single output file. The output file is SUM.TIM, which is stored on DECtape under UFD [50,10]. Notice how ONE.WAY inherits default values from SAM.LIT. (ONE.WAY and SAM.LIT are located on NODEY on DU0: under UFD [50,10].) Access control information for both NODEA and NODEY is assumed to be defined in either the respective default node names or in alias node names.

Example 5: Multiple file transfer

```
NFT>
NFT>
```

In this example, BEE.FIL, CEE.FIL, and DEE.FIL are copied to NODEA in ASCII record mode. They are stored under their original names, since the output file names default to the input file names. Specifying an output file name would cause concatenation rather than multiple file transfer (see Example 4).

When you transfer multiple files, all input files must be sent to the same node. However, the input files can be accessed from different nodes.

Example 6: Multiple file transfer using wildcards

```
NFT>
```

In this example, all files on RSX node BOSTON, in the account associated with UFD [310,2] and password MGR, are copied to the local node. All files retain their original names.

NOTE

> Brackets ([ ]) around the user ID are optional if the user ID is in the form of a UIC, as in the preceding example.

4.2.6 Delete /DE

The delete operation deletes one or more specified files. The files can be located on the same or different nodes.

Format:

file-descriptors[/*sw*]/DE

where

file-descriptors are one or more valid file descriptors, as defined in Section 4.1. The version number must always be specified for files deleted on RSX nodes.

/sw may include one or both of the following qualifying switches (defined in Section 4.2.3):

File Qualifying Switches	Command Qualifying Switches
(Not Applicable)	/LO
	/NM

On RSX systems, if there is a wildcard (* or %) in the file name or type, the version number must be explicit or wildcard (that is, it cannot be 0 or −1).

Example:

```
NFT>
```

This command line deletes the files TAX.LST;5 and BUDGET.LST;1.

4.2.7 Directory (/LI,/FU,/AT,/BR)

Switches:

/LI[:*width*]

/FU[:*width*]

/AT

/BR

MCR has four types of file directory listings. The /LI, /FU, /AT, and /BR switches display different types of directory listings:

- /LI (list switch) lists a directory consisting of the name, size, creation date, and optional contiguous and locked indicators for files on a remote node.

- /BR (brief listing switch) lists a directory containing only the names of files located on a remote node.

- /FU (full directory listing switch) lists a full directory for files on a remote node. This list consists of name, size, creation date, optional contiguous and locked indicators, owner, protection, data last modified, and revision number.

- /AT (attributes listing switch) displays the attributes for each file in a directory listing. This display includes the file organization, record format, and record attributes, as well as the information included in a full (/FU) directory listing.

Format:

$$[\textit{outfile} =]\textit{infiles} \left\{\begin{array}{l} \text{/LI}[:\textit{width}] \\ \text{/FU}[:\textit{width}] \\ \text{/BR} \\ \text{/AT} \end{array}\right\}$$

where

outfile is the optional file descriptor to which the directory will be written. If omitted, the output defaults to TI:.

infiles are one or more file descriptors for which you want directory listings. If you omit the file name or type from the descriptor, you will receive a directory listing of all files.

width is a decimal number that specifies the width of the listing. The default value is 72. At least one field is always displayed per line, even if that field exceeds the maximum width specified.

Example 1: The List switch (/LI)

```
NFT>
```

The DB0:[SMITH] directory on node X appears on TI: as follows:

```
Directory X::DB0:[SMITH]
3-FEB-86 13:26:02

X.MAC;3         2./2.        29-JAN-86 17:46:23
X.OBJ;4         1./1.        29-JAN-86 17:46:34
X.TSK;2         4./4.    C   29-JAN-86 17:46:51
Y.TSK;3         0./0.    CL  29-JAN-86 17:47:08
```

The fields, from left to right, are file name, blocks used/allocated, [contiguous indicator], [locked indicator], and creation date and time. All fields may not be displayed for all systems.

Example 2: The Brief listing switch (/BR)

```
NFT>
```

The DB0:[7,7] directory on node X is displayed on TI: as follows:

```
Directory X::DB0:[7,7]
3-FEB-86 13:21:22

X.LST;3
X.MAC;3
X.OBJ;4
X.TSK;2
```

Foreign file name formats may differ. RT–11 file names, for example, do not have version numbers.

Example 3: The Full listing switch (/FU)

```
NFT>
```

The directory of DB0:[7,7] on node X is displayed on TI: as follows:

```
Directory X::DB0:[7,7]
3-FEB-86 13:27:39

X.OBJ;4        1./1.                 29-JAN-86 17:46:34
[7,7]          [RWED,RWED,RWED,R]    01-FEB-86 09:58:32(2.)
```

The fields, from left to right by line, are:

First Line

file name
blocks used/allocated
creation date and time

Second Line

file owner (UIC)
file protection
last modified date and time
revision number

An uppercase character C to the left of the creation date indicates that the file is stored on contiguous blocks, as in the following example:

```
X.TSK;2         4./4.              C   29-JAN-86 17:46:51
[7,7]           [RWED,RWED,RWED,R]     01-FEB-86 09:58:36(2.)
```

An uppercase L to the left of the creation date indicates that the file is locked, as in Example 1. The fields vary according to the file system; RT–11 directories, for example, omit file owner and modification date and time.)

Example 4: The Attributes listing switch (/AT)

```
NFT>
```

The attributes of the file LOGIN.CMD;22 on node QUEENS are displayed:

```
Directory DB0:[EVANS]
15-JUN-86 12:00:06

LOGIN.CMD;22
                   Size:         2./2.    Created: 15-JUN-1983 10:40
                   Owner:     [EVANS]     Revised: 15-JUN-1983 10:41 (2.)
                                          Expires: 31-DEC-1983
File protection:   System:RWED, Owner:RWED, Group:R, World:
File organization: Sequential
File attributes:   Allocation=2, Extend=0
Record Format:     Variable length, no maximum defined
Record attributes: Carriage return
```

4.2.8 Execute (/EX)

The execute operation executes the contents of a command file located on a local or remote node. For RSX systems, the command file can be submitted to either a command file processor or a batch file processor, depending on how the destination node's command file submission task (MCM) was generated during NETGEN. To execute a command file that does not already exist on a remote node, you must use the submit operation.

The success of the execute operation indicates that the execution request was given successfully to the MCM, not that the batch or command file executed successfully.

Format:

command-files[/*sw*]/EX

where

command-files are output file descriptors for each remote node command file to execute. Command files are not deleted after execution.

/sw can be one or both of the following qualifying switches (defined in Section 4.2.3):

File Qualifying Switches	Command Qualifying Switches
(Not Applicable)	/LO /NM

Example:

```
NFT>
```

In this example, the command file NAMES.CMD is to be executed on node WASH.

4.2.9 Help (/HE or HELP)

The NFT help operation displays information on using NFT. For a list of available help topics, enter /HE or HELP.

Format:

/HE

HELP

For information on a topic that the initial help screen lists, type

NFT > HELP [*subject*]

Example:

```
NFT>
```

This command line requests information on transferring files.

Help files must be located on device LB:[1,2]. If they are not there, you get the following message:

```
NFT - No HELP Available
```

If this occurs, consult your system manager.

4.2.10 Identify (/ID)

The identify operation displays the current NFT and NFARs version numbers, your node's operating system and file support, and the DAP buffer size. If you specify a remote node in the command, /ID displays information for the remote FAL.

Format:

[*node-specification*::]/ID

where

node-specification is a remote node name and access control information (see Section 4.1 for information on specifying node names and access control information).

Example 1: Local node

```
NFT>

NFT - Version 4.2
Local NFARs V4  DAP V7.0  Buffer size=528.  OS=RSX-11M-PLUS  FS=FCS-11
```

Example 2: Remote RSX node

```
NFT>

NFT - Version 4.2
Local NFARs V4  DAP V7.0  Buffer size=528.  OS=RSX-11M-PLUS  FS=FCS-11
Remote FAL  V2  DAP V7.0  Buffer size=550.  OS=RSX-11M-PLUS  FS=RMS-11
```

Example 3: Remote VMS node

```
NFT>

NFT - Version 4.2
Local NFARs V4  DAP V7.0  Buffer size=528.  OS=RSX-11M-PLUS  FS=FCS-11
Remote FAL  V5  DAP V6.0  Buffer size=4160. OS=VAX/VMS       FS=RMS-32
```

4.2.11 Rename (/RE)

The rename operation changes the name of one or more files. You specify the existing names as the *infiles* and the new names as the *outfiles*. The rename operation deletes existing file names from their directory and enters new file names into their directory. The operation does not transfer data; both directories must be on the same node and device. Access control information is therefore necessary for the *infiles* only. If the renaming includes moving the files to a different directory, that directory must also allow access.

Some systems do not support the Data Access Protocol (DAP) rename operation. Refer to the individual system documentation for more information.

Format:

outfile[*/sw*] = *infiles[/sw*]/RE

where

outfile is the file specification for the new file. See Section 4.1.2 for a description of file specification format. Each individual file name, type, and version number can be explicit, wildcard (*), or defaulted (null), with two restrictions:

- If there is a wildcard (*) in the file name or type, the version number must be explicit or wildcard (that is, it cannot be 0 or −1).
- Partial field or single character wildcard specifiers are not allowed for *outfile*.

When a field is wildcard or defaulted, the corresponding field of the input file is used. You can change one or more fields while preserving the others. See Section 4.1.2.1 for more information on wildcards.

infiles are one or more file descriptors of files to rename.

/sw may include one or more of the following qualifying switches (defined in Section 4.2.3):

File Qualifying Switch	Command Qualifying Switches
/NV	/LO
	/NM

Example 1:

```
NFT>
NFT>
```

The example changes the name only, from OLDNAME.FIL to NEWNAME.FIL.

Example 2:

```
NFT>
NFT>
```

This example changes both the name and the directory; no file data is moved.

4.2.12 Set Default (/DF)

The set default operation defines default values for various arguments in a file descriptor. Using the /DF switch, you can set default values for node names (complete with access control information), devices, and UICs. However, you cannot set defaults for the file name, type, or version number.

You set default values for input files by specifying IN, and for output files by specifying OUT. You can give the same default values to both input and output files by using one command and omitting the IN/OUT arguments. Defaults apply to subsequent files until you change defaults again or exit from NFT.

Format:

[*defaults*]/DF $\begin{bmatrix}\text{:IN}\\ \text{:OUT}\end{bmatrix}$

where

defaults is any string of default argument values that you define in the command line. If you omit the default values from the command line, NFT displays the current defaults.

:IN is the input file specifier. When appended to the /DF switch, :IN applies the default values defined by the command to input files only.

:OUT is the output file specifier. When appended to the /DF switch, :OUT applies the default values defined by the command to output files only.

Any file descriptor arguments that the /DF switch does not set have the standard NFT defaults listed in Table 4–1. To display current defaults, enter the following:

NFT > /DF (RET)

NFT displays the current default values at your terminal.

Example:

```
NFT>
NFT>
NFT>
```

This example defines input arguments (node name, access control, and UIC) and an output argument (device) for subsequent files. The command NFT> = TIN.MAN assumes the following input and output file descriptors:

```
NFT>
```

To display the defaults for this example, you enter NFT> /DF RET and get the following display:

```
Input defaults = MASTER/GUMBO/...:[4,5]

Output defaults =TI:
```

Notice that the default password is in ellipses (...) to maintain security.

4.2.13 Set Protection (/PR)

The set protection operation sets or changes a file's protection status. File protection has four levels:

- /SY:*p* (System) specifies access to the system UICs. System UICs have group numbers of 10 octal or less.

- /OW:*p* (Owner) specifies access to yourself.

- /GR:*p* (Group) specifies access to other members of your group.

- /WO:*p* (World) specifies access to all other UICs.

The *p* variable represents the R, W, E, and/or D protection settings. These settings specify whether members of each level can read, write, extend, or delete a file. The R, W, E, and/or D settings apply to all input files in a command line, unless you override the setting for a specific file.

To change a file's protection level, you use either the /PR qualifying switches (/SY:*p*, /OW:*p*, /GR:*p*, /WO:*p*) or an octal representation (/PR:*n*). You can change a file's protection level if you are the file owner or have a system level UIC, even without read or write access. However, a file's protection level is a protected attribute, and you cannot change it if you are in the group or world category for the file owner's UIC. You can read protected attributes if you have read access to the file.

Format:

```
                  ⎧:n[/FO:uic]                                     ⎫
file-descriptors/PR⎨ [/SY:p][/OW:p][/GR:p][/WO:p][/FO:[uic]]          ⎬
                  ⎩/FO:[uic]                                       ⎭
```

where

file-descriptors are one or more file descriptors for each file in the command line.

/PR is the protect switch.

:*n* is an optional octal value that can specify the protection status. (For a list of the possible octal codes and their meanings, refer to the *RSX–11M* or *RSX–11M–PLUS Pocket Guide*.)

/SY: is the system level protection qualifying switch (defined in this section and in Section 4.2.3).

/OW: is the owner level protection qualifying switch (defined in this section and in Section 4.2.3).

/GR: is the group level protection qualifying switch (defined in this section and in Section 4.2.3).

/WO: is the world level protection qualifying switch (defined in this section and in Section 4.2.3).

p is R, W, E, and/or D, representing the access to be allowed. R, W, E, and D represent the following access types:

R – read allowed	E – extend allowed
W – write allowed	D – delete allowed

Only the access that you specify (R, W, E, and/or D) will be allowed. An access level (/SY:*p*, /OW:*p*, /GR:*p*, /WO:*p*) without a *p* value gives no access at that level.

/FO:[*uic*] is the file owner qualifying switch. This switch allows you to set a file's owner to a specified UIC, in the form: [*g,m*].

/sw can include one or both of the following qualifying switches (defined in Section 4.2.3):

File Qualifying Switches	**Command Qualifying Switches**
(Not Applicable)	/LO /NM

Example 1: Changing protection status on a transfer operation

```
NFT>
```

File ALIAS.CMD allows read, write, and delete operations to the owner only.

Example 2: Changing protection status on an existing file

```
NFT>
```

File ALIAS.CMD allows the owner to read only.

4.2.14 Spool (/SP)

The spool operation queues a file for printing. You can use /SP to queue a local or remote file on its own system or to transfer the file for queuing on a different system.

To spool a file on a remote node, you must use a file specification compatible with the remote system. For format conversion, use the /AS and /RC switches.

Format:

[*outfile*[/*sw*] =]*infiles*[/*sw*]/SP

where

outfile is an optional file descriptor. It specifies the output file to copy to and the system to spool to. If you omit *outfile*, the *infiles* are spooled on the input node.

infiles are one or more file descriptors for each file to spool for printing.

/sw may include one or more of the following qualifying switches (defined in Section 4.2.3):

File Qualifying Switches	Command Qualifying Switches
/AS	/LO
/AX	/NM
/BK	
/IM	
/NV	
/RC	
/SU	

Example 1:

```
NFT>
```

This example spools one copy of the file COLD.FIL;1 to the line printer on node MACON and one copy of the file SNOW.FIL;1 to the line printer on node HOME. No transfer occurs.

Example 2:

```
NFT>
```

This example spools the local file STARTUP.CMD on the remote node QUEENS.

4.2.15 Submit (/SB)

The submit operation transfers a command file to a remote node and submits it to either a remote command file processor or a batch file processor. The remote node executes the contents of the file and then deletes it. The remote node must support command file submission and execution (RT–11 nodes, for example, do not). Also, the syntax of the command lines in the command file must conform to the remote system's syntax standards.

For RSX systems, NFT submits the command file to either a command file processor or a batch file processor, depending on how the destination node's command file submission task (MCM) was generated during NETGEN.

Successful completion of the submit operation means that the file was transferred and the execution request passed to the MCM, not that the file was executed successfully.

To execute a command file that is already on the remote node, use the execute (/EX) operation described in Section 4.2.8.

Format:

outfile[*/sw*] = *command-file*[*/sw*]/SB

where

outfile — is a file descriptor for the temporary file created on the destination node. This file receives the command file's contents and is deleted after execution.

command-file — is the file descriptor of the input command file to execute at the destination node. The submit operation takes only one command file at a time.

/sw — can include one or more of the following qualifying switches (defined in Section 4.2.3):

File Qualifying Switches	Command Qualifying Switches
/AS	/LO
/AX	/NM
/BK	
/IM	
/NV	
/RC	

Example:

```
NFT>
```

In this example, file INST.CMD is transferred to DENVER as SVEFIL.CMD. SVEFIL.CMD is then executed and is deleted after execution is complete.

5

The File Transfer Spooler (FTS) Utility

The File Transfer Spooler (FTS) utility can transfer and manipulate files like NFT (see Chapter 4), but can also queue file operations for later processing. You can use FTS only from MCR. FTS can

- Append files to an existing file
- Transfer files between two nodes
- Delete files
- Execute command files located on a remote node
- Spool files to a line printer
- Submit local command files to a remote command file processor for execution and subsequent deletion

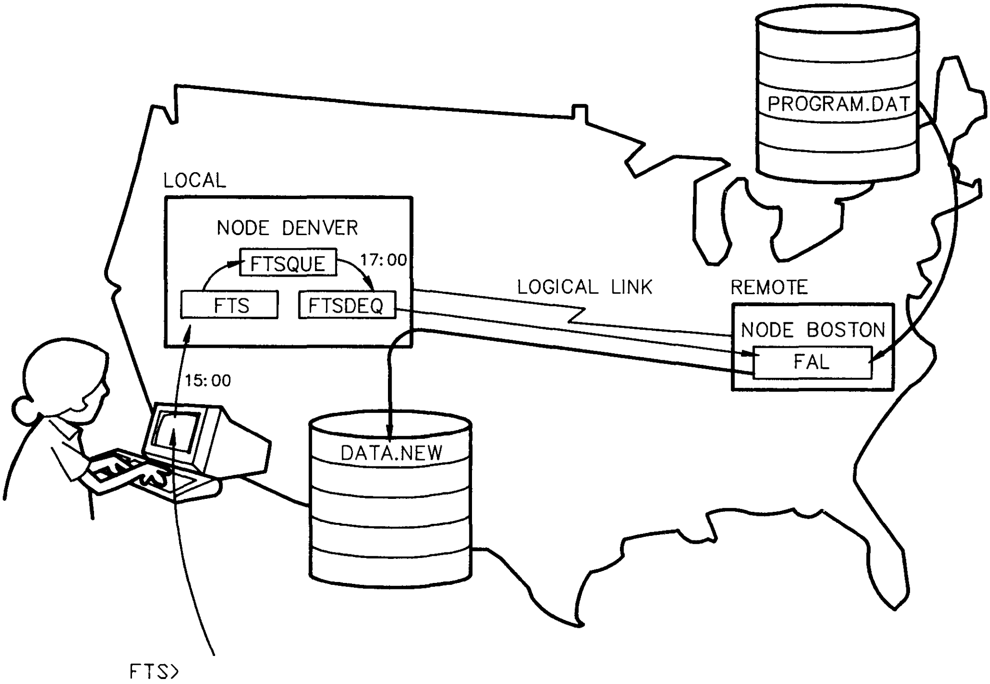
In addition, FTS can

- Queue and process user requests according to specified dates and times
- List pending requests
- Kill pending requests
- Log the completion of your request

When you send commands to FTS, FTS sends your request to the system queue manager (QMG...) task on your node. The queue manager puts your request into the FTS queue (FTSQUE). When your request is ready to be processed, the FTS dequeuer (FTSDEQ) task removes it from the queue. FTSDEQ communicates with the DECnet server task File Access Listener (FAL) on the remote node, to answer your request.

Figure 5–1 shows a user requesting an FTS copy operation. The operation copies the file BOSTON::PROGRAM.DAT to DENVER::DATA.NEW after 5:00 P.M. The user includes, as access control information for the remote system, the user name Billie and password MOONLIGHT.

Figure 5–1: The FTS Utility: Queuing File Transfer Requests



LKG–0596

5.1 FTS File Descriptor Syntax

An FTS command requires file descriptors that identify the files on which the command operates. The file descriptors can be for local and remote files. FTS uses file descriptor syntax that is similar to NFT's, with the following exceptions:

- FTS does not accept logical names in file descriptors.
- FTS allows wildcards only in local input file specifications.

FTS, unlike NFT, looks up wildcards immediately and records the applicable files in the user request block (URB) file for subsequent execution. An FTS wildcard operation therefore includes only the applicable files at the time that you submit the command. The operation will not include any subsequent files, even if the wildcard applies.

For information on file descriptors, refer to the NFT file descriptor information in Section 4.1. For information on wildcard field specifiers, refer to the NFT wildcard information in Section 4.1.2.1.

5.2 FTS Operations

To specify an FTS operation, you use a switch. Each FTS primary switch represents a file operation. For example, to append the contents of one file to the contents of another, you use the /AP switch. Some operations also have qualifying switches that further define the action of the primary switch. For example, if you use the /AS qualifier with /AP, the /AS switch causes FTS to append the file in ASCII record mode. Table 5–1 summarizes the FTS operations. Later sections describe each FTS operation.

5.2.1 Specifying FTS Switches in a Command Line

Primary switches can be anywhere in a command line.

A qualifying switch on an input file specification affects only that file. A qualifying switch on an output file specification, however, becomes global and affects all input files in the command, unless you override the switch for specific files.

A switch on a file specification cannot appear before the file name, type, version, or UIC.

5.2.2 FTS Command Line Continuer

To use multiple input lines to specify an FTS operation, enter a hyphen (-) and press the RETURN key anywhere in the command line. FTS will then prompt you for the remainder of the command. You can use any number of continuing lines within a command. The maximum length of a command string is 256 characters.

Example:

```
FTS>
FTS>
```

Table 5-1: Summary of FTS Operations

Operation	Primary Switch	Qualifying Switches
Append: appends files to the end of an existing file.	/AP	/AF /AS /IM /LO /PR /SE
Copy: transfers one or more existing files to one or more files on the local or remote node.	default	/AF /AS /IM /LO /PR /SE
Delete: deletes one or more files.	/DE	/AF /LO /PR /SE
Execute: executes a command file stored on a local or remote node.	/EX	/AF /LO /PR /SE
Help: displays information about switches and commands.	/HE HELP [*subject*]	
Identify: identifies your version of FTS.	/ID	
Kill: removes a request from the FTS queue.	*/KI	

* Switches that NFT does not support or that have different functions with NFT.

Table 5–1 (Cont.): Summary of FTS Operations

Operation	Primary Switch	Qualifying Switches
List: lists your pending user requests.	*/LI	
Set default: sets certain default values for input and output files.	/DF	
Spool: spools files to a line printer for printing.	/SP	/AF /LO /PR /SE
Submit: copies a command file to a local or remote node, executes the file, and deletes it.	/SB	/AF /LO /PR /SE

* Switches that NFT does not support or that have different functions with NFT.

5.2.3 FTS Qualifying Switches

/AF The after switch queues a user request for execution after a specified date and time.

Format:

command-line/AF[:*dd-mmm-yy*]:*hr:mn*

where

command-line is a user request that FTS queues for execution at the specified time.

dd-mmm-yy is the day, month, and year on which to execute the request. If you omit this argument, the request uses the current date.

hr:mn is the time interval after which to execute the request. Use the 24-hour format to specify hours (*hr*) and minutes (*mn*).

Example 1:

```
FTS>
FTS>
```

FTS will transfer PAYROLL.TSK;4 from node LONDON to node DALLAS on May 21, 1986, after 3:30 P.M.

Example 2:

```
FTS>
```

FTS will transfer file SALES.LST;2 from the local system to LONDON after 5:15 P.M. on the current date.

/AS The ASCII switch transfers files in ASCII record mode. The switch applies only to the current command line.

For remote output files, FTS translates the records in the file into an appropriate format for the remote system. For local output files, FTS creates the file as a variable length format, implied CR/LF attribute file.

Use the /AS switch only if you want translation. File transfers are more efficient in the default image block mode if the remote node has a file system similar to RSX (VMS, for example, does).

/IM The image switch transfers files while retaining their current format and attributes. Some remote systems may not support the transfer, causing an error message to be displayed. If the file is ASCII text, you can recover by using the /AS switch.

/LO The log switch enables FTS to write a completion message when it completes one of your requests. Normally, the system manager enables system logging. Only privileged users can access that system log file, however. To put FTS messages in a file of your own, you use the /LO switch. The /LO switch enables FTS logging to a file in your directory. The default directory is your user directory, and the default file type is .LOG. You can, however, create a log file with the file specification you want, by appending the file specification to the /LO switch. If the log file does not yet exist, FTS creates it. If the file does exist, FTS appends messages to it.

To use /LO with a sequence of user requests, you put the /LO switch on the first command line of the sequence. To disable logging for a specific command, you use /NOLO or /–LO on the command line.

Format:

/LO:*file-spec* This format for /LO enables logging and gives a file specification for your user log file, overriding the default log file destination for the current command line. The default file type is .LOG.

/LO — This format for /LO sends logging to the default file specification SY0:FTS.LOG when logging is enabled by the system manager.

/–LO
or
/NOLO — These negative formats for /LO disable logging for the current request.

Examples:

The following command logs a message by default to the common system user log file:

```
FTS>
```

The following command logs a message by default to the user's own log file, FTS.LOG:

```
FTS>
```

The third command defines a user log file with the specification PAYROLL.LOG:

```
FTS>
```

The next command line uses the /NOLO switch to suspend logging for only this request:

```
FTS>
```

The fifth and sixth commands show how you use logical device names in a user log file specification. FTS directs the completion message to the console pseudodevice (CO:) in the fifth command and to the user's terminal (TI:) in the sixth command.

```
FTS>

FTS>
```

Refer to Appendix B for information about the completion messages that the log file contains.

/PR The priority switch specifies the priority that FTS assigns in queuing a request.

Format:

/PR:*priority*

where

priority is a decimal number from 1 to 250.

If you do not specify the priority switch, FTS queues the user request at the priority of the running task, which is FTS at default priority 50.

Example:

```
FTS>
```

FTS places the user request to transfer FISCAL.DOC;1 to MAIN in the queue, giving it a priority of 65.

/SE The sequence switch processes requests on the current command line and on subsequent command lines in the order in which you input them. You can use the /SE switch to ensure that FTS processes requests in the proper order.

Format:

command-line1
command-line2/SE
command-line3/SE
command-line4/SE
command-line5
command-line6

FTS processes command lines 2 through 5 in the order in which you enter them. Command line 5 is subsequent to command line 4 and is therefore included, even though it has no /SE switch. An error at any point in the sequence will cause FTS to stop and not queue the requests for processing. In the sequence, FTS processes command lines 1 and 6 independently.

Example:

```
FTS>
FTS>
FTS>
FTS>
FTS>
FTS>
```

The commands for transfers to node BOSTON will be processed in the order in which they are input.

5.2.4 Append (/AP)

The append operation appends the contents of one or more input files to the end of an existing output file. The output file retains its original attributes. The attributes of the input and output files should be the same, or the append operation may appear to succeed but produce an incorrect output file.

Format:

outfile = *infiles*[/*sw*]/AP

where

outfile is the output file descriptor. You cannot use wildcard specifiers in the output file descriptor. The file and record attributes are taken from the existing file, but you must explicitly specify the file name and type.

infiles are one or more input file descriptors.

/*sw* can be one or more of the following qualifying switches (defined in Section 5.2.3):

/AF
/AS
/IM
/LO
/PR
/SE

Example:

```
FTS>
FTS>
```

In this example, FTS opens FIL1.DAT;1, which is stored on DU1: at node LOWELL. FTS then appends FIL2DAT;1, located at node ESTER, and FIL3.DAT;1, located on node UTICA, to FIL1.DAT;1 in ASCII record mode.

5.2.5 Copy

The FTS copy operation includes three types of file transfers:

- Single file copy transfers a single file from one node to another.
- File copy with concatenation transfers more than one input file to create a single output file. The input files can be from the same or different nodes.

  You need not use a switch to specify the copy operation; you use an equals sign between the input and output file specifications, as the format shows.

  The attributes of the first input file in the command line become the attributes of the output file. If the input files have different attributes, the concatenation may appear to succeed, but may create an incorrect output file.
- Multiple file copy transfers more than one input file to the same number of output files; each input file has a corresponding output file.

If an error occurs during transfer, FTS or the remote File Access Listener (FAL) attempts to delete the output file.

If you include a name and type in the output file specification, FTS will concatenate multiple input files to form the output file. If an error occurs while FTS is processing a concatenation copy request, it stops processing the request without transferring any of the files.

A multiple file copy has more than one output file and more than one input file. To specify a multiple file copy , omit the output file name and type. You can specify a node name, device, and UIC.

Format:

outfile = *infiles*[*/sw*]

where

infiles are one or more input file descriptors.

/sw can be one or more of the following qualifying switches (defined in Section 5.2.3):

/AF
/AS
/IM
/LO
/PR
/SE

Example 1: Single file copy from a remote node to the local node

```
FTS>
```

This example copies the file INVENTORY.LST;4 from node DENVER to the local node and stores the file under the same file name in the UFD [BURRELL].

Example 2: File copy with concatenation from the local node to a remote node

```
FTS>
FTS>
```

This example copies FICA.LST, FEDERAL.LST, and STATE.LST to DALLAS and concatenates them as TAXES.LST.

Example 3: Multiple file copy from the local node to a remote node

```
FTS>
```

This example copies PENS.INV, PAPER.INV, and PENCILS.INV to node DENVER, where they retain their file names.

Example 4: Single file copy between two remote nodes

```
FTS>
FTS>
```

In this example, FTS copies COMPUTERS.LST;4 from node DENVER to node BOSTON, where it has the same file name and type. If a file already has that file name and type on BOSTON, a new version of the file is created.

5.2.6 Delete (/DE)

The delete operation deletes one or more files (see Format A), or transfers a file and then deletes its local input version (see Format B). You can transfer and delete files only through FTS; NFT has no equivalent operation.

Format A:

The following format deletes one or more files. No transfer occurs:

file-descriptors[/*sw*]/DE

where

file-descriptors — are one or more valid file descriptors. You must include the version number to delete files from RSX nodes.

/sw — can be one or more of the qualifying switches listed under Format B.

If you use a wildcard (* or %) in the file name or type, you must give an explicit or wildcard version number. The version cannot be 0 or −1. See Section 3.1.2.1 for more information on wildcards.

Other systems may have different requirements for deleting files. Check the operating system documentation for information on its requirements.

Format B:

The following format copies a single input file and then deletes it from the source node. If the copy operation fails, the input file is not deleted.

outfile = *infile*[/*sw*]/DE

where

outfile — is a file descriptor for the file to receive the data.

infile — is a file descriptor for the file to transfer and delete.

/sw can be one or more of the following qualifying switches, defined in Section 5.2.3:

/AF
/LO
/PR
/SE

Example 1: Request to delete a list of files

```
FTS>
FTS>
```

In this example, FICA.LST;3 and TAXES.LST;4 on node DALLAS are deleted.

Example 2: Request to delete a file after copy

```
FTS>
```

In this example, USELESS.LST;4 is deleted locally after FTS successfully copies it to node LYNN.

5.2.7 Edit (/ED)

Edit mode lets you check your command lines before queuing them. In edit mode, FTS accepts and checks your command lines but does not submit them for execution.

In edit mode, FTS detects syntax errors, nonexistent local input files, and illegal local input file descriptors. It does not, however, check the remote file descriptors. FTS will find any errors in remote file descriptors at execution time.

To return to queuing mode, enter CTRL/Z.

Example:

```
FTS>

FTS - Entering Edit mode

FTS>

FTS - Leaving Edit mode
```

5.2.8 Execute (/EX)

The execute operation executes the contents of a local or remote command file. For RSX systems, you submit the command file to a command file processor or a batch file processor, depending on how the destination node's command file submission task (MCM) was generated during NETGEN. To execute a command file that is not already on the remote node, you must use the submit operation. The submit operation copies the command file to the remote node and then submits it for execution.

The FTS execute operation succeeds when the MCM receives the execution request. A successful FTS execute operation does not guarantee successful execution of the batch or command file.

Format:

command-files[/*sw*]/EX

where

command-files are one or more output file descriptors for each remote node command file to execute. Command files are not deleted after execution.

/sw can be one or more of the following qualifying switches (defined in Section 5.2.3):

/AF
/LO
/PR
/SE

Example:

```
FTS>
```

In this example, the remote node BOSTON will execute the command file RETRIEVE.CMD.

5.2.9 Help (/HE or HELP)

The help operation displays information on using FTS. To display the list of available help topics, enter the switch or its abbreviation.

Format:

/HE

HELP

For information about a topic that the initial help screen lists, enter:

HELP [*subject*]

Example:

```
FTS>
```

This help request lists the valid FTS switch options.

Help files must be located on device LB:[1,2]. If they are missing, you get the following message:

```
FTS - No help available
```

If you get this message, consult your system manager.

5.2.10 Identify (/ID)

The identify operation displays the current FTS version number.

Example:

```
FTS>

FTS - VERSION 4.2
```

5.2.11 Kill (/KI)

The kill operation deletes a pending user request or aborts a request in progress. The kill switch has the following forms:

job#/KI deletes the specified user request.

node-name::/KI deletes all your requests involving the specified node.

/KI deletes all user requests you have queued.

where

job# is a 6-digit number that FTS assigns to a user request when you issue the request. To display the FTS job number, use the /LI switch.

node-name is the name of the node to which you want to stop requests.

The kill switch deletes or aborts only your own requests.

When you enter the /KI request, FTS attempts to delete the user request file and to send a request to the queue manager to kill the queue entry.

Example 1: Delete a single user request

```
FTS>
```

Removes user request with job number 162880 from the queue.

Example 2: Delete all user requests involving a specific node

```
FTS>
```

Removes all your user requests involving node BOSTON from the queue.

5.2.12 List (/LI)

The list operation displays pending FTS requests from your UIC.

Format:

[*node-name*::]/LI

where

node-name limits the display to the node that you specify. If you omit *node-name*, FTS lists all user requests from your UIC.

The display of pending requests contains two header lines. The first contains the date and time. The second contains the job number, type/priority, and options headings. If no requests are queued, FTS displays a message saying that no requests were found. If requests are in the queue, but not for the node name that you specified, FTS displays the headings and a message that no requests were found.

On the first line of each request description, FTS displays the job number. Next is a code that specifies the operation to perform (as defined in Table 5–2), a slash, and the priority of the request. Next are the options that you specified. /IM is a default that appears unless you used /AS to specify ASCII transfer mode. /LO is a default that appears unless the system manager has disabled logging. /AF, with a date and time, appears as an option if you have specified it or if the request has been requeued.

The second line of the request description tells when the request was queued and how many times it was tried. (The value of times tried is greater than 0 when an attempted request has failed and been automatically requeued with the /AF switch for execution at a later time.) Next are the input and output file descriptors. The last line shows the user log file specification if you used /LO to specify one.

Example 1: Print user requests involving specified node name

```
FTS>

FTS USER REQUEST LISTING                  28-Jun-86     11:30
Job #         Type/Priority        Options

141954         TLR/50               /IM/LO
 Queued 28-Jun-86 10:45                   Active
 Source File - DB0:[224,1]FICA.LST;1
 Destination File - DALLAS::DU1:[200,200]TAXES.LST

382077         TLR/50              /AF:28-Jun-86:11:35/IM/LO
 Queued 28-Jun-86 10:50                   Tried 1 Time
 Source File - DB0:[224,1]STATE.LST;1
 Destination File  DALLAS::DU1:[200,200]TAXES.LST
 Log File  DB0:DALLAS.LOG
```

Example 2: Print all user requests

```
FTS>

FTS USER REQUEST LISTING                  28-Jun-86     11:45
Job #          Type/Priority       Options

234892         DR/50               /IM/DE/LO
 Queued 28-Jun-86 9:00                    Active
 Source File - BOSTON:DB0:[224,1]USELESS.LST;4

415121         TLR/50              /AF:28-Jun-86:11:50/IM/LO
 Queued 28-Jun-86 9:15                    Tried 2 Times
 Source File - DB0:[224,1]PAYROLL.CBL
 Destination File - DENVER::SY0:[200,200]PAYROLL.CBL
 Log File - PAYROLL.LOG

415220          TRR/50             /AF:28-Jun-86:11:55/IM/LO
 Queued 28-Jun-86 9:20                    Tried 1 Time
 Source File - BOSTON::DB0:[200,1]TAXES.CBL
 Destination File - LONDON::DU4:[221,221]TAXES.CBL

410455         SLR/50              /AF:28-Jun-86:12:00/IM/LO/SB
 Queued 28-Jun-86 9:25                    Tried 1 Time
 Source File - DB0:[372,1]SALES.LST
 Destination File - GENEVA::SALES.LST

463340         ER                  /AF:28-JUN-86:15:30/IM/EX/LO
 Queued 28-Jun-86 9:30                    Tried 0 Times
 Source File - BURMA::COMMANDS.EXE
```

Table 5-2: Summary of FTS Function Codes

Function Code	Description
CLL	Concatenate local files to a local file
CLR	Concatenate local files to a remote file
CRL	Concatenate remote files to a local file
CRR	Concatenate remote files to a remote file
DL	Delete files on local node
DR	Delete files on remote node
EL	Execute command files issued for the local node
ER	Execute command files issued for a remote node
PL	Print a file on the local node
SLL	Submit a command file from a local node to a local node
SLR	Submit a command file from a local node to a remote node
SRL	Submit a command file from a remote node to a local node
SRR	Submit a command file from a remote node to a remote node
TLL	Transfer local files to local files
TLR	Transfer local files to remote files
TRL	Transfer remote files to local files
TRR	Transfer remote files to remote files

5.2.13 Set Default (/DF)

The set default operation lets you define your own default values for arguments in a file descriptor. The /DF switch sets default values for node names (complete with access control information), devices, and UICs. You cannot set default values for the file name, type, and version number.

:IN sets default values for input files only, and :OUT sets default values for output files only. A command that does not specify :IN or :OUT sets the same default values for input and output files. Defaults apply until you change them or exit from FTS.

Format:

[*defaults*]/DF [:IN
               :OUT]

where

defaults is a string of default argument values. Without *defaults*, /DF displays the current default values.

:IN applies the operation to input files only.

:OUT applies the operation to output files only.

Any file descriptor arguments not set with the /DF switch have the standard defaults defined in Table 3–1. To display the current default values, enter:

FTS > /DF RET

FTS displays the current /DF default values.

Example:

```
FTS>
FTS>
FTS>
```

In this example, the first command line gives input file defaults for node name, access control information (*/userid* and */password*), and UIC. The second command line gives the default device for output files. The transfer command FTS > = INVENTORY.LST assumes the following input and output file descriptor fields:

```
DBO:=INVENT/STATUS/GEARS::[224,1]INVENTORY.LST
```

To display the default values for this example, you enter the /DF switch at the FTS > prompt and get the following response:

```
Source defaults = INVENT/STATUS/...::[224,1]

Destination defaults = DB0:
```

Note that the default password is displayed in ellipses (...) to maintain security.

5.2.14 Spool (/SP)

The spool operation queues a file on a local or remote node for printing. When you spool a file to a remote node, the file specification must be appropriate to that system.

Format:

[*outfile* =]*infiles*[/*sw*]/SP

where

outfile is an optional file descriptor specifying the output file to which FTS copies the file and the system on which FTS spools it. If you omit *outfile*, FTS spools the *infile* on the source node.

infiles are one or more file descriptors for each file to spool for printing.

/sw can be one or more of the following qualifying switches, defined in Section 5.2.3:

/AF
/LO
/PR
/SE

Example 1:

```
FTS>
```

In this example, the spool operation queues one copy of COLD.FIL;1 to the line printer on node MACON, and one copy of SNOW.FIL;1 to the line printer on node HOME.

Example 2:

```
FTS>
```

In this example, the spool operation queues one copy each of COLD.FIL;1 and SNOW.FIL;1 to the line printer on node HOME. The input file names default to the output file name.

5.2.15 Submit (/SB)

The submit operation transfers a command file to a remote node, submits it to a remote command file processor or batch file processor for execution, and, after execution, deletes it on the remote node. The remote node must support command file submission and execution (RT–11 nodes, for example, do not). Remember that the syntax of the command lines in the command file must conform to the conventions of the remote system.

For RSX systems, the command file can be submitted to either a command file processor or a batch file processor, depending on how the destination node's command file submission task (MCM) was generated during NETGEN.

Success of the submit operation means that FTS has successfully transferred the file and given the execution request to the MCM. It does not mean that the file has executed successfully.

To execute a command file that resides on a remote node, use the execute (/EX) operation (see Section 5.2.8).

Format:

outfile = *command-file*[/*sw*]/SB

where

outfile is the file descriptor for a temporary file on the remote node. The submit operation copies the command file's contents to this file and then deletes it after execution.

command-file is the file descriptor of command files to execute on the remote node. You can specify only one command file on a command line.

/sw can be one or more of the following qualifying switches, defined in Section 5.2.3:

/AF
/LO
/PR
/SE

Example:

```
FTS>
```

This example shows a submit operation that transfers file COMMANDS.CMD;5 to GENEVA as NEWFILE.CMD, executes the file, and then deletes NEWFILE.CMD.

6

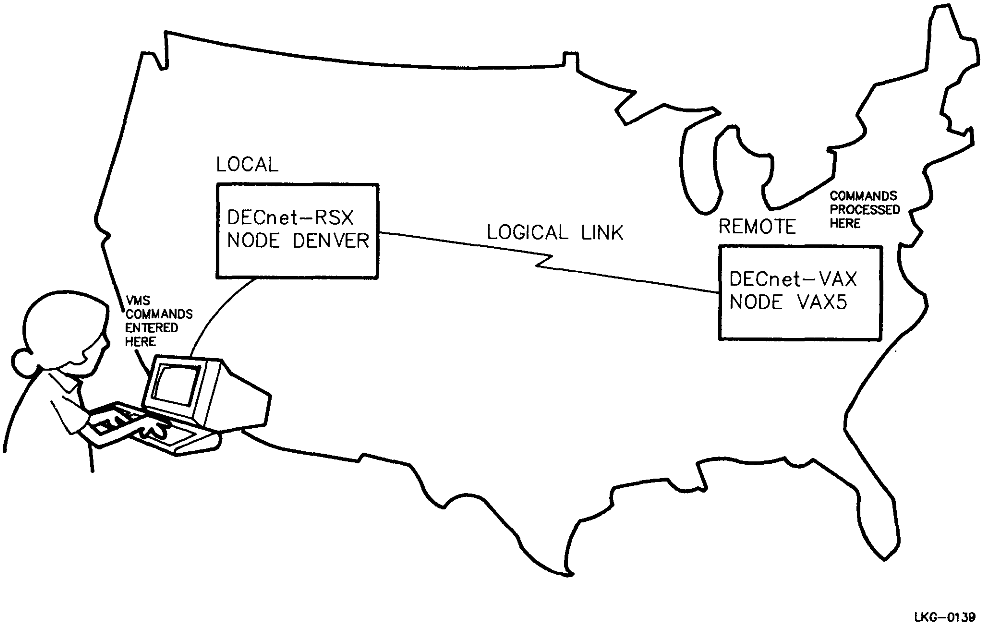
The Network Command Terminal (NCT) Utility

The Network Command Terminal utility (NCT) lets you use the resources of remote host nodes in the network. NCT creates a logical connection that lets you log on to and use the remote operating system as if your terminal had a physical connection to the host (see Figure 6–1).

On your DECnet–RSX host, the Network Command Terminal (NCT) process services outgoing requests, while the Remote Terminal Host (RTH) process services incoming requests. NCT and RTH both implement the command terminal protocol, CTERM. To support your NCT connection, the remote host requires a process that implements CTERM. CTERM is available on most operating systems that support DECnet.

NCT is available on RSX–11M–PLUS, RSX–11M/RSX–11S (V4.2 or later), and Micro/RSX systems. Note that NCT is not supported on DECnet–11M–PLUS versions prior to V3.0, DECnet–11M/11S versions prior to V4.1, or VMS versions prior to V4.0. Also note that when using NCT to connect to a VMS system, you cannot do line editing.

Figure 6-1: The NCT Utility: Access to Remote Host Operating Systems



6.1 Connecting to a Host System

Before you can log on to a remote host operating system, you use a SET command that connects your terminal to another system. The following sections describe the command's MCR and DCL formats.

6.1.1 Using MCR Format

You use the following format on DECnet–11M and DECnet–11M–PLUS systems:

SET /HOST = *host-node* [::]

You use the following MCR format on DECnet–11S systems:

RUN SETHST
Host: *host-node* [::]

where *host-node* is the 1- to 6-alphanumeric character node name of the VMS or RSX host node. The node name must contain at least 1 alphabetic character.

Example:

This example shows a connection to an RSX host.

```
>

Connected to "TOLEDO", System type = RSX-11M-PLUS
System ID: RSX-11M-PLUS V4.0

>
Password:

GOOD AFTERNOON

>
```

Note that the screen does not echo the password.

6.1.2 Using DCL Format

You can use the following format with DCL:

SET HOST *host-node*

where *host-node* is the 1- to 6-alphanumeric character node name of the VMS or RSX host node. The node name must contain at least 1 alphabetic character.

Example:

This example shows a connection to a VMS host.

```
$

Username:
Password:
                 Welcome to VAX/VMS V4.4 on node VAX5

$
```

Note that the screen does not echo the password.

6.2 Disconnecting from a Host System

There are two ways to terminate your session and disconnect from a host system:

- Log off the system by entering the appropriate command.
- Clear your connection by entering a control character sequence that returns you to the local node. At the local node, you enter the CLEAR HOST command.

Logging off may be simpler in most cases. However, the second method lets you return to your local node to terminate an RMT session even when you are not logged on to or do not know the commands of the host.

6.2.1 Logging Off a Host System

To log off RSX host systems, you use the BYE command. To log off VMS host systems, you use the LOGOUT command.

6.2.1.1 Logging Off an RSX Host

BYE logs you off RSX hosts, as in the following example:

```
>
>
Connect time: 1 minute
CPU time:     1 second
Task total:   7
Have a good afternoon
7-AUG-86 22:02 RT2: logged off TOLEDO
>

NCT -- Control returned to node "MIAMI"

>
```

On an RSX host node, you can also use BYE/HOLD. BYE/HOLD logs you out while maintaining your connection to the host; you return to the host's initial CLI prompt. To use the host again, you can log back in.

6.2.1.2 Logging Off a VMS Host

LOGOUT logs you off a VMS host, as in the following example:

```
$
HANDEL logged out at 7-AUG-86 22:02:58.31

NCT -- Control returned to node "MIAMI"

>
```

6.2.2 Clearing Your Connection

To clear your connection with the remote host and return to your local node, press CTRL/\ RET (press the CONTROL and back slash (\) keys and then press the RETURN key). Your local node prompt appears:

```
$
TOLEDO>
```

You can now choose to terminate or resume your connection with the remote host. To terminate, enter the CLEAR HOST command. NCT confirms your return to the local node, as in the following example:

```
TOLEDO>
NCT -- Control returned to node "TOLEDO"
>
```

To resume your session on the host node, enter the CONTINUE command at the local node prompt.

6.3 Sample NCT Session

In the following NCT session, you connect from a local DECnet–RSX node, TOLEDO, to a remote VMS host node, VAX5. Once logged on to the VMS host, you use the DIRECTORY command to display a list of files and the COPY command to copy PEOPLE.LST;1 to your account on node TOLEDO. You then enter the LOGOUT command to terminate your session.

```
>

Username:
Password:

                   Welcome to VAX/VMS V4.4 on node VAX5

$


Directory USER$DISK:[HANDEL]

REMINDER.CMD;1    MASTERINI.CMD;1    TELI.CMD;1      TESTFILE.COM;1
LOGIN.CMD;1       PEOPLE.LST;1       NULLTEST.LST;1  COPY.LST;1
NFTTEST.LOG;1     TSETCMP.LOG;1      SETCMP.LOG;1

Total of 11 files.

$
_From:
_To:
$
HANDEL logged out at 7-AUG-86 22:02:58.31

NCT -- Control returned to node "TOLEDO"

>
```

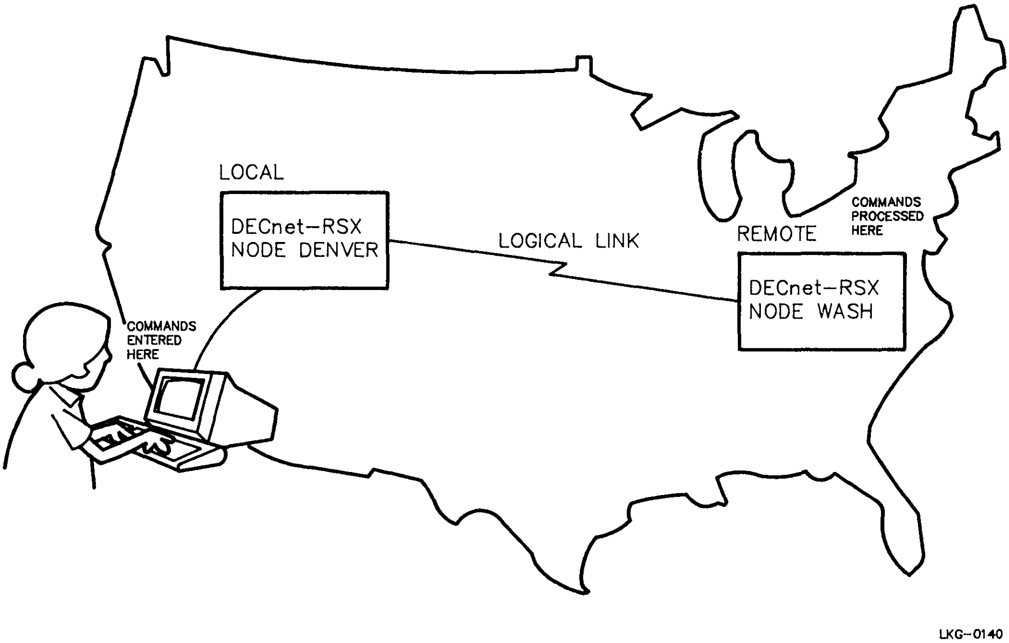
7 The Remote Terminal (RMT) Utility

The Remote Terminal (RMT) utility lets you log on to another DECnet–RSX host system. Once RMT makes the logical connection and you log on, you use the host system's resources as if your terminal were physically connected to it. When you use RMT, the local node is the one to which your terminal is physically connected. The host node is the one to which RMT logically connects your terminal. Figure 7–1 illustrates the RMT utility.

The RMT utility lets you connect to any host in the network that supports the RMT host facility, RMHACP. When you are connected to the host system by RMT, your terminal device is designated HT:. RMT lets you execute any program that is mutually supported by HT: and the local terminal driver. This excludes the TLK utility, which is supported only by the TT: device. See Appendix C for information on the QIO functions that the HT: driver supports.

You can use RMT on RSX–11M, RSX–11M–PLUS, and RSX–11S systems.

Figure 7-1: The RMT Utility: Access to Remote DECnet-RSX Node Resources



7.1 Connecting to a Host Node

To connect to a host node, you invoke RMT and name the node to which you want to connect. The format for invoking RMT depends on your system. The next sections describe how you invoke RMT from a DECnet-11M, DECnet-11M-PLUS, or DECnet-11S system.

Do not invoke the RMT utility from an indirect command file or batch job; the results are unpredictable.

7.1.1 Connecting from DECnet-11M and DECnet-11M-PLUS Systems

There are two methods for invoking RMT from DECnet-11M and DECnet-11M-PLUS systems.

The first method is to type RMT after the MCR (or DCL) prompt. RMT prompts you for a host name. RMT then establishes the connection and identifies the operating system to which you are connected, as in the following example:

```
>
Host:

Connected to "MIAMI", System type = RSX-11M-PLUS
System ID: Quality Assurance Division
```

The second method is to invoke RMT and enter the host node name on the same line, as this example shows:

```
>

Connected to "MIAMI", System type = RSX-11M-PLUS
System ID: Quality Assurance Division
```

7.1.2 Connecting from DECnet-11S Systems

To invoke RMT from a DECnet-11S system, issue the RUN command for RMT after an MCR prompt (>), as in the next example:

```
>
Host:

Connected to "MIAMI", System type = RSX-11M-PLUS
System ID: Quality Assurance Division
```

The host node name is defined by $HOST and can be changed by the system manager using the Network Control Program (NCP) or virtual network processor (VNP). If you omit the host node name, RMT substitutes the default host node name. The default host node name is initialized as the local node name for RSX-11M, RSX-11M-PLUS, and RSX-11S systems. If an RSX-11S system has been down-line loaded, $HOST is set to the name of the system from which RSX-11S was loaded. Refer to the *DECnet-RSX Network Management Concepts and Procedures* manual for more information on setting the default host node name.

The following example illustrates an RMT log-in to remote host node SALEM.

Example of RMT log-in:

```
>
Host:

Connected to "SALEM", System type = RSX-11M
System ID: MAPPED RSX-11M V4.3

>
PASSWORD:

RSX-11M V4.3 MULTI-USER

GOOD MORNING
3-MAY-86 10:12 LOGGED ON TERMINAL HT3:

>
```

Note that the password does not echo.

7.2 Disconnecting from a Host Node

When you finish remote terminal operations, type BYE to log off the host system. On multiuser host systems, BYE automatically disconnects you from the host node and returns control to the local operating system.

To maintain your connection with the host node, you can use BYE/HOLD. BYE/HOLD returns you to the MCR prompt. Although you are still connected to the remote host, you must log back in to use the host system resources.

Example:

```
>
>
HAVE A GOOD AFTERNOON
21-MAY-86 14:20 HT2: LOGGED OFF

RMT -- Control returned to node TULSA::

>
```

You can also disconnect from the host node by typing EXIT RMT at an MCR prompt.

If a local system error causes your terminal to log off without first exiting from the RMT task, RMT automatically terminates your connection with the host node. If an error causes RMT's connection to the host to terminate before you can log off, the HT: on the host will automatically be logged off.

7.3 Aborting RMT

On DECnet-11M-PLUS systems, the Remote Terminal task (RMTACP) can be aborted through an Executive feature. Since this feature does not exist on RSX-11M or RSX-11S, it is possible for a privileged user to abort the task while it is in use. If RMTACP or any other privileged task aborts, system data structures can become corrupted. If this occurs, users connected through RMT can enter a ''garbage'' line. This line invokes RMTACP again and resets user terminals to their original state.

If RMTACP aborts, do not remove it from the system until all terminals reconnect to the local system. Premature removal of RMTACP can cause a system crash.

7.4 RMT Control Character Handling

RMT handles most control characters (CTRL/O, CTRL/S, CTRL/Q, and CTRL/R) according to standard RSX-11 conventions. CTRL/C, however, is different. CTRL/C displays the RMT> prompt. RMT then passes your input to the command language interpreter (CLI) at the host system, generally MCR or DCL. In normal RSX-11 usage, CTRL/C causes the local CLI prompt to appear.

In the following example, the user enters CTRL/C to interrupt a session and use the RMT command ACT. This command lists the active tasks at the terminal.

Example:

```
>

RMT>
MCR...
ACTH3
MACH3
```

7.5 Sample RMT Session

The following example shows a typical RMT session. The example shows how a user connects to host node FLINT from local node RACINE, issues a PIP command for a directory list, and then exits from RMT. Note that passwords never echo on the screen.

```
>

Connected to "FLINT", System type = RSX-11M
System ID: REMOTE TERMINAL TEST SITE

>
PASSWORD:

RSX-11M V4.3 MULTI-USER SYSTEM

GOOD AFTERNOON
15-MAR-86 14:23 LOGGED ON TERMINAL HT3:

>

DIRECTORY DB0:[354,20]
15-MAR-86 14:27:03

RMHACP.OLB;10                110.          04-MAR-86 11:52
HTDRV.OLB;10                 19.        C  04-MAR-86 11:53
RMHACP.CMD;7                 14.           06-MAR-86 12:32

RMT>
14:28:57              TASK "PIPH3" TERMINATED
                      ABORTED VIA DIRECTIVE OR MCR
                      AND WITH PENDING IO REQUESTS
>
>

RMT -- Control returned to node RACINE::

>
```

8

The Phone Communications (PHO) Utility

The Phone Communications (PHO) utility lets you have an interactive dialog with one or more other users. The other users can be on your node or a remote DECnet–RSX, DECnet–VAX, or PRO/DECnet node. PHO simulates the features of an actual telephone, including the hold feature, the telephone directory, and conference calling. PHO also has many advanced features that are not available with the TLK utility (see Chapter 9) such as the ability to ''phone'' a person by name, leave a message with someone who was not in when you ''phoned'', and include the contents of a file in your conversation.

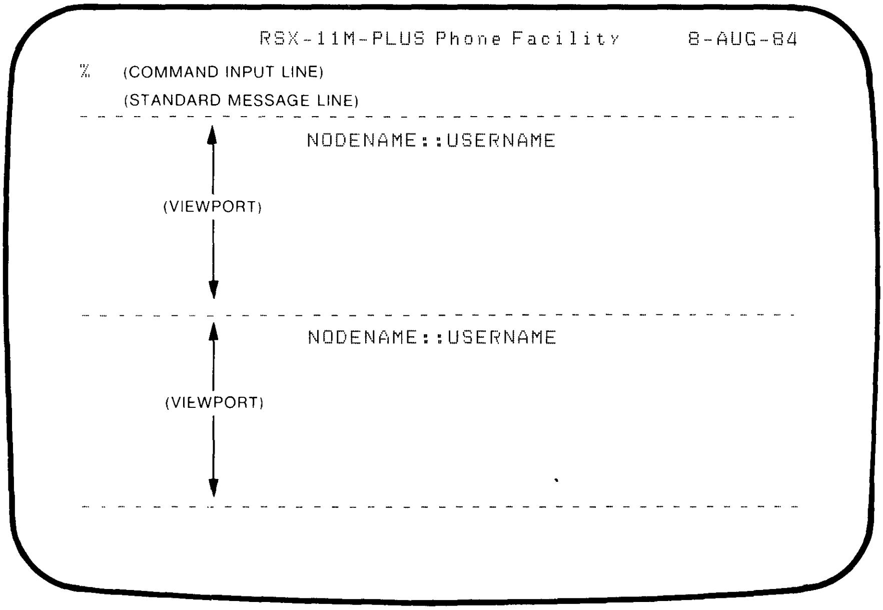
PHO is available only on RSX–11M–PLUS and Micro/RSX systems.

The rest of this chapter explains how to use PHO and describes the commands, qualifiers, and control characters for your PHO session.

8.1 PHO Screen Format

PHO works only on video terminals with direct cursor positioning, such as the VT200 and VT100 series and VT52 terminals. When you invoke PHO, your screen will look like the sample screen in Figure 8–1.

Figure 8–1: The PHO Utility: Sample Screen Format

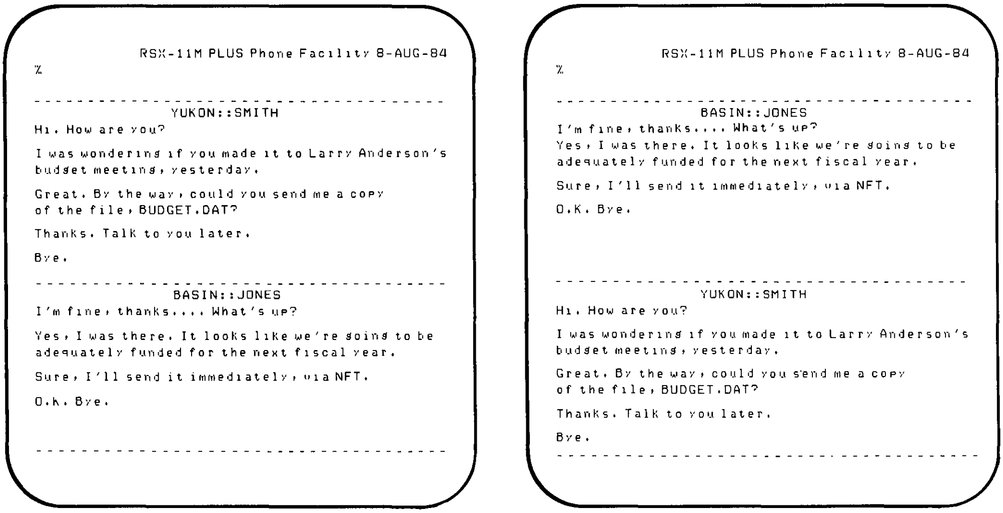


```
               RSX-11M-PLUS Phone Facility      8-AUG-84
%  (COMMAND INPUT LINE)
   (STANDARD MESSAGE LINE)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                    NODENAME::USERNAME
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                    NODENAME::USERNAME
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

NOTE: The switch hook character shown in this sample screen is the default percent sign, (%). The switch hook character is always displayed in column 1 of the command input line.

TW141

Figure 8-2: The PHO Utility: Sample PHO Conversation



USER SMITH'S SCREEN ON NODE YUKON

```
           RSX-11M PLUS Phone Facility 8-AUG-84
%

--------------------------------------------
               YUKON::SMITH
Hi. How are you?
I was wondering if you made it to Larry Anderson's
budget meeting, yesterday.
Great. By the way, could you send me a copy
of the file, BUDGET.DAT?
Thanks. Talk to you later.
Bye.
--------------------------------------------
               BASIN::JONES
I'm fine, thanks.... What's up?
Yes, I was there. It looks like we're going to be
adequately funded for the next fiscal year.
Sure, I'll send it immediately, via NFT.
O.K. Bye.

--------------------------------------------
```

USER JONES' SCREEN ON NODE BASIN

```
           RSX-11M PLUS Phone Facility 8-AUG-84
%

--------------------------------------------
               BASIN::JONES
I'm fine, thanks.... What's up?
Yes, I was there. It looks like we're going to be
adequately funded for the next fiscal year.
Sure, I'll send it immediately, via NFT.
O.K. Bye.


--------------------------------------------
               YUKON::SMITH
Hi. How are you?
I was wondering if you made it to Larry Anderson's
budget meeting, yesterday.
Great. By the way, could you send me a copy
of the file, BUDGET.DAT?
Thanks. Talk to you later.
Bye.
--------------------------------------------
```

TW142

The percent sign (%) at the top of the screen is the default switch hook character. The switch hook is the prompt for entering PHO commands.

Each participant in the phone conversation has a separate area on the screen, called a viewport. Each viewport is labeled with a participant's DECnet node name and user name, and is separated from the other viewports by a broken line (see Figures 8–1, 8–2). The text that each participant enters in the conversation appears in his/her own viewport, along with various status messages, such as HOLD messages.

PHO can display up to six viewports at a time. To make room for more participants, you can put one or more on hold, temporarily eliminating those viewports from your screen.

8.2 Conference Calls

Conference calls allow three or more users to have a concurrent PHO session. To set up a conference call, one participant acts as the operator. The operator calls the other participants in turn and waits for a response.

Users who are participating in a conference call can use the DIAL command to bring new participants into the conference. If the person you "dial" is already participating in a conference call, your conversation will be with only that person, not the members of the other conference call.

8.3 Invoking and Exiting from PHO

To invoke PHO, use the following command format:

PHONE [*/qualifier(s)*] [*command*]

where

/qualifiers are one or more qualifiers that modify the characteristics of a PHO session. Table 8–1 lists and describes the PHO qualifiers.

command is a PHO command that PHO will execute before prompting you for additional commands. Table 8–2 lists and summarizes each PHO command.

You can truncate the PHONE command or any PHO command keywords to the first three letters.

To exit from PHO and return to the CLI:

- Enter the EXIT command. EXIT hangs up your phone, terminates your PHO session, and returns you to the CLI.

- Enter CTRL/Z to hang up the phone, and then another CTRL/Z or an EXIT command to exit from the PHO utility.

8.4 Summary of PHO Qualifiers, Commands, and Control Characters

This section includes tables that list the PHO qualifiers, commands, and control characters. Table 8–1 lists the PHO qualifiers. You enter these qualifiers on the command line that invokes PHO to specify conditions for your session. Table 8–2 lists the PHO commands. You can use these commands during a PHO session; you can, additionally, use some of them on the command line that invokes PHO. Table 8–3 lists control characters that you can use during your PHO session. PHO ignores any other control characters.

Table 8–1: PHO Qualifiers

Qualifier	What It Does
/SCROLL	Specifies how PHO scrolls text in your viewport when it becomes full. With /SCROLL, your conversation text moves up one line and the next line of text appears at the bottom of your viewport. With /NOSCROLL, new text is wrapped to the top line of your viewport. /NOSCROLL gives faster response time on slow-speed terminals. The default is /SCROLL.
/VIEWPORT_SIZE = *n*	Specifies the maximum number of lines in your viewport. A viewport extends from the heading line down to the broken bottom line. The size range is 3 to 10 lines; 10 is the default. Smaller viewports give faster response time on slow-speed terminals.
/SWITCH_HOOK = *c*	Specifies the switch hook character for a PHO session. The switch hook character signals the PHO utility that the text that follows is a PHO command. Text not preceded by this character is part of the PHO conversation. The default switch hook character is the percent sign (%). For example, suppose that while having a PHO conversation, you want to use the DIR command to display the names of users on node AKRON. You type the switch hook character and enter DIR AKRON RET. Your screen displays node AKRON's phone directory. The next character you type returns your phone conversation. The switch hook character appears in column 1 of the command input line (see Figure 8–1).

Table 8–2: PHO Commands

Command	What It Does
ANSWER	Answers the phone when you get a call.
DIAL	Phones another user.
DIRECTORY	Displays a list of users that you can phone on a specified node.
EXIT	Terminates your PHO session. Typing CTRL/Z from the command input line is equivalent to EXIT.
FACSIMILE	Includes the contents of a specified file in your conversation.
HANGUP	Hangs up your phone and disconnects all current links. Typing CTRL/Z is equivalent to HANGUP.
HELP	Displays information on how to use PHO.
HOLD	Places conversation participants on hold.
MAIL	Leaves a short MAIL message for another user.
UNHOLD	Reverses a previous HOLD command.

Table 8–3: PHO Control Characters

Character	Function on a Command Line	Function in a Conversation
DEL	Deletes previous character.	Same
LF	Ignored	Deletes previous word.
RET	Executes command.	Advances cursor to beginning of next line in viewport.
TAB	Ignored	Advances cursor to next tab stop.
CTRL/G	Ignored	Sounds bell at your terminal and the terminal of all participants in the current phone conversation.
CTRL/L	Ignored	Clears all text in your viewport.
CTRL/S	Freezes the screen.	Same
CTRL/Q	Negates a CTRL/S	Same
CTRL/U	Clears current command line.	Clears current viewport line.
CTRL/W	Refreshes the entire screen.	Same
CTRL/Z	Exits you from PHO and returns control to the CLI. Equivalent to EXIT command.	Hangs up your own phone. Equivalent to HANGUP command.

8.5 PHO Commands

This section describes each PHO command. You can truncate all PHO command keywords to the first three letters. Some PHO commands require additional information after the command keyword.

If you are using PHO but are not currently having a conversation, you can enter PHO commands at the switch hook character prompt (see Figure 8–1), instead of typing the switch hook character before entering the command. During a phone conversation, however, you must type the switch hook character before entering a PHO command. The switch hook character tells PHO to treat the following text as a command. After the command executes, press RET to return to your phone conversation.

Table 8–3 describes the control characters for formatting your PHO session.

8.5.1 ANSWER

The ANSWER command lets you receive a call from another user. If you get a call when you are not using the PHO utility, PHO broadcasts a message to your terminal at approximate 10-second intervals. The message includes the name of the caller and the time of the call. The message repeats until you type PHONE ANSWER, or until the caller hangs up. The message is similar to the following example:

```
MIAMI::SMITH is phoning you on node UTICA::      (14:00:05)

MIAMI::SMITH is phoning you on node UTICA::      (14:00:11)
```

If someone calls while you are using PHO, the message appears in PHO's standard message line, under the command input line (see Figure 8–1).

When you receive a message that someone is phoning you, you have three choices:

- Answer the call, using the command PHONE ANSWER
- Reject the call, using the command PHONE REJECT
- Ignore the call

To prevent calls from reaching your terminal, you can set your terminal to /NOBROADCAST status.

8.5.2 DIAL

The DIAL command places a call to another user. To place the call, you enter the DIAL command and specify the person to call. If the person is on your node, just enter the user name. If the person is on a remote node, enter the DECnet node name and user name. The format for the DIAL command is:

PHONE DIAL [*node-name*::]*user-name*

You can substitute an alias for the *node-name*:: and/or *user-name* parameters.

If you enter a user name with no other PHO command, DIAL is the default command. You can enter a user name from the switch hook character to call another user.

Example:

```
>
```

When you enter this command, PHO broadcasts a message to Smith's terminal. The message repeats at Smith's terminal at approximate 10-second intervals until Smith answers or rejects your call, or you hang up the phone.

Note that you can also enter the PHO utility and call another user by entering PHONE *user-name* from the CLI.

8.5.3 DIRECTORY

The DIRECTORY command displays a list of users that you can phone on a given node. If you omit the node name, the command displays a directory for your node. If you specify a remote node name, the command displays a directory for that node. The format for the DIRECTORY command is:

PHONE DIRECTORY [*node-name* [::]]

You can substitute an alias for the *node-name* [::] parameter.

The directory appears on your screen line by line, until the entire user list is displayed or until you type any keyboard key. If you are having a PHO conversation, typing any keyboard key cancels the directory listing and returns your conversation. If you are not having a PHO conversation, typing a key returns you to the PHO command input line.

The directory display contains the following information:

- The process name and user name of each person currently logged on
- The terminal identifier, TT*n*, for each user
- The availability status of each user's simulated phone. /NOBROADCAST status prevents a person from receiving incoming calls.

Sample Directory Listing:

```
>

               RSX-11M-PLUS Phone Facility     15-AUG-86

%
Press any key to cancel directory listing and continue.
-------------------------------------------------------------

Process Name          User Name      Terminal     Phone Status

BROWNSTONE            BROWNSTONE     unusable     ---
Allen                 Ginsburg       TT2          available
Joe Smith             SMITH          TT4          /nobroadcast
Etta James            James          TT13         available

4 persons listed
```

8.5.4 EXIT

The EXIT command executes the HANGUP command, terminates your session with PHO, and returns control to the Command Language Interpreter (CLI).

Typing CTRL/Z when you are not having a conversation is equivalent to the EXIT command.

8.5.5 FACSIMILE

The FACSIMILE command displays the contents of a file during your phone session. The file appears in the viewport of every participant in the session. PHO displays the contents of the file until it reaches the end of the file or until you type any keyboard key. The format for the FACSIMILE command is:

FACSIMILE *file-spec*

8.5.6 HANGUP

The HANGUP command hangs up your own phone. This command disconnects all current links — the current conversation, everyone you have on hold, and anyone who has you on hold. When someone issues a HANGUP command, the other participants in the conversation get the message:

user-name just hung up the phone

where *user-name* is the name of the person who has hung up.

Typing CTRL/Z while having a conversation is equivalent to the HANGUP command.

8.5.7 HELP

The HELP command displays information about the PHO utility. For general information about PHO, type:

PHONE HELP

For information on an individual command or topic, add the command or topic name to the HELP command:

PHONE HELP *topic*

The help information appears on your screen. To stop the help display, type any keyboard key. If you issued the HELP command from a PHO conversation, the keyboard key returns you to your conversation; otherwise, the key returns you to the command input line.

8.5.8 HOLD

The HOLD command places everyone currently on the phone with you on hold, including anyone who has you on hold. PHO informs each person that he/she has been placed on hold. You can call other users while you have someone on hold.

8.5.9 MAIL

The MAIL command sends a short message to another user. The message can be no longer than one line. You can use this command to leave a message for someone who is not in when you call.

When sending mail to someone on your own node, enter the MAIL command, the user name, and the message. Enclose the message in quotation marks. When sending mail to someone on a remote mode, include the person's DECnet node name with the user name. The format for the MAIL command is:

MAIL [*node-name*::]*user-name "single line message"*

You can use an alias for the *node-name*:: and/or *user-name* parameters.

8.5.10 REJECT

The REJECT command lets you refuse to receive a phone call from another user. PHO will display a message on the caller's terminal that you have rejected the call.

You can use the REJECT command with the optional EXIT parameter. With EXIT, REJECT refuses the call and then terminates the session. The format for the REJECT command is:

```
PHONE REJECT [EXIT]
```

You can also define REJECT in your log-in command file as the PHONE REJECT EXIT command. You can then simply enter REJECT to refuse a call.

8.5.11 TRANSCRIBE

The TRANSCRIBE command writes the preceding conversation text to a file that you specify. The other conversation participants receive a message saying that the conversation has been transcribed. The format for the TRANSCRIBE command is:

TRANSCRIBE *file-spec*

8.5.12 UNHOLD

The UNHOLD command negates the last HOLD command. UNHOLD returns the users you previously placed on hold to their prior conversation. Note that UNHOLD reverses the effects of only the most recent HOLD command.

9 The Terminal Communications (TLK) Utility

The Terminal Communications (TLK) utility lets you send single line messages to or have a dialog with other terminal users in the network. Using TLK, you can have "conversations" with users on your local node or an accessible remote node (see Figure 9–1). The remote node must support the Listen (LSN) utility in order to receive and execute your TLK requests. TLK is supported on DECnet–RSX, DECnet–IAS, DECnet–RT, and DECnet/E nodes. Indirect command files can be used with TLK, except on RSX–11S nodes.

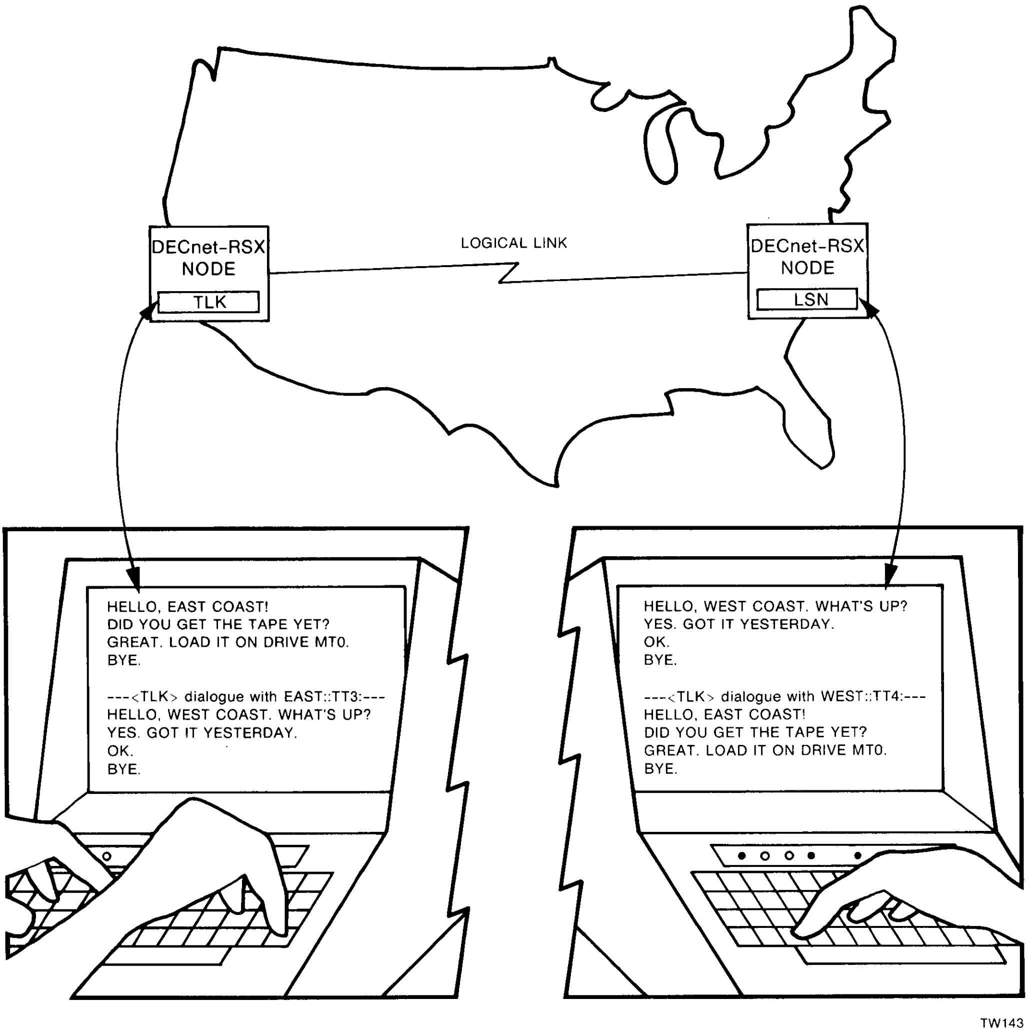
9.1 TLK Communication Modes

TLK has two communication modes for transmitting messages:

- Single message mode lets you send a single line message to another terminal (see Section 9.2).

- Dialog mode lets you conduct an interactive exchange of messages—a dialog—with another terminal (see Section 9.3).

Figure 9–1: The TLK Utility: Communication Between Terminals



```
HELLO, EAST COAST!
DID YOU GET THE TAPE YET?
GREAT. LOAD IT ON DRIVE MT0.
BYE.

---<TLK> dialogue with EAST::TT3:---
HELLO, WEST COAST. WHAT'S UP?
YES. GOT IT YESTERDAY.
OK.
BYE.
```

```
HELLO, WEST COAST. WHAT'S UP?
YES. GOT IT YESTERDAY.
OK.
BYE.

---<TLK> dialogue with WEST::TT4:---
HELLO, EAST COAST!
DID YOU GET THE TAPE YET?
GREAT. LOAD IT ON DRIVE MT0.
BYE.
```

9.2 Single Message Mode

Using single message mode, you can send a short message or notice to any other terminal in the network that supports TLK. The message must be no longer than a single line.

9.2.1 Format for Sending a Message in Single Message Mode

The format for sending a single-line TLK message is:

TLK [*target-node*::][TT*n*:]'*message*

where

target-node:: is the 1- to 6-character name of the node to which you are sending the message. Follow the node name with a double colon. Your local node is the default target node; to send a message to a terminal on your local node, omit this argument.

TT*n*: is the the target terminal identification, where *n* is an octal number from 0 to 177. Omit this parameter to reach the default operator's console (CO:). See the description in Section 9.3.1 on how to find the TT*n*:.

'*message* is any ASCII string to send as a message.Do not extend the string beyond the end of the command line. Precede the string with an apostrophe; the apostrophe is optional if the command line includes the terminal device (TT*n*:).

Example 1: Sending a single-line message to a remote user

A local user sends a message from his terminal, TT12: on node UTICA, to remote user TT5: on node AKRON. The example includes the target terminal argument and therefore omits the apostrophe.

```
>
```

When the message is received, the target terminal displays it along with header information identifying the source node and TT*n*:, and the date and time of receipt:

```
<TLK> UTICA::TT12: 16-DEC-86 10:11:30
I AM READY FOR A FILE TRANSFER.
```

As soon as it sends the message, TLK prompts you for another message if you entered the message from the TLK> prompt, or returns the CLI prompt if you entered the message from the CLI. If TLK prompts you but you want to end the TLK session, type CTRL/Z.

Example 2: Sending a single-line message to a local user

```
>
```

Because this example omits the target node, TLK sends the message to TT2: on the local node by default.

9.3 Dialog Mode

Using dialog mode, you can have a conversational exchange of messages with another network terminal user. Dialog mode is interactive. The user at another terminal can respond to you without establishing a new connection. Both terminals in the TLK session can send and receive messages, regardless of which terminal initiated the dialog. If the remote user sends a message while you are typing a line, the message is displayed, interrupting the line you are typing. TLK then redisplays your interrupted line so you can resume typing where you left off. See Section 9.3.4 for a sample dialog mode exchange. See Appendix A for information on the messages TLK displays when an attempt to start dialog fails.

If your terminal driver allows IO.RPR and IO.WBT terminal functions (see Appendix C), you have hard-copy dialog support.With IO.ATA, you get video dialog support for TLK and LSN.

9.3.1 Format for Sending Messages in Dialog Mode

The format for a TLK dialog is:

```
>TLK [target-node::][TTn:] RET
<TLK> - START OF DIALOG
TLK>message
      .
      .
      .
TLK> CTRL/Z
```

where

target-node:: is the 1- to 6-character name of the node to which you are sending the message. Follow this node name with a double colon. Your local node is the default target node; to send a message to a terminal on your local node, omit this argument.

TT*n*: is the target terminal identification, where *n* is an octal number from 0 to 177. Omit this parameter if you want to reach the default operator's console (CO:).

message is any ASCII string that you send as a message.

To send TLK messages, you must enter the target node name and terminal identifier (TT*n*:) for the user with whom you want to communicate. One way of learning a user's TT*n*: is to use the PHO utility (see Chapter 8) to get a directory for the target node. The directory listing includes the TT*n*: for each user currently logged onto the target node.

Another way of learning a user's TT*n*: is to use the RMT utility (see Chapter 7) to log on to the target node and then enter the MCR DEVICE /LOG or the DCL SHOW USERS command. On an RSX–11M–PLUS system, the command displays the TT*n*:, among other information about each user currently logged on. On an RSX–11M system, the command displays only the TT*n*: and UIC.

Example:

```
>

TT11:   [232,201] [350,363] 14-APR-86 09:58  1 B. SMITH
TT12:   [316,314] [350,362] 14-APR-86 10:14  0 J. JONES
TT13:   [350,346] [350,346] 14-APR-86 11:57  1 S. ANDERSON
TT22:   [300,210] [350,377] 14-APR-86 14:55  2 D. PAGE
```

9.3.2 Sending and Receiving Dialog Messages

In the following examples, you are a user on local node UTICA::TT12: who wants to start a dialog with a remote user on node AKRON::TT5:. You enter the following command:

```
>
```

The remote user then receives the following question and must answer YES or NO:

```
<TLK> -- Do you want to talk to UTICA::TT12:? [Y/N]:
```

If the remote user who receives this question does not answer within approximately 10 seconds, a message like the following will appear:

```
<TLK> -- Timed out on response (10:15:02)
```

If the remote user answers YES to have a TLK session, a message like the following appears and prompts for input:

```
<TLK> IN DIALOG WITH UTICA::TT12: 16-DEC-86 10:14:55
TLK>
```

You, as initiator of the dialog, get a message that tells you that dialog has started and prompts you for messages:

```
>
<TLK> - START OF DIALOG
TLK>
```

You can enter a new message line whenever you get the TLK> prompt:

```
>
<TLK> - START OF DIALOG
TLK>
TLK>
TLK>
```

9.3.3 Terminating Dialog Mode

When you finish your dialog, either you or the remote user can press CTRL/Z to terminate. CTRL/Z displays the following message on both terminals:

```
<TLK> - END OF DIALOG
```

9.3.4 Sample Dialog Mode Exchange

In the following example, NODEZ initiates a TLK dialog to request files from NODEM. The messages that NODEM sends in response temporarily interrupt NODEZ's input.

As this example shows, dialog mode terminates when either user presses CTRL/Z (displayed as ^Z). CTRL/Z causes the end of dialog message to appear on both terminals.

```
NODEZ                                    NODEM

>

                                         <TLK> - DO YOU WANT TO TALK TO
                                         NODEZ::TT2:? [Y/N]:
```

```
<TLK> - START OF DIALOG                 <TLK> IN DIALOG WITH NODEZ::TT2:
                                        12-JAN-86 12:30:02
TLK>                                    TLK>
                                        <TLK> HELLO NODEM. THIS IS NODEZ.
TLK>                                    TLK>
                                        <TLK> DO YOU READ ME?
TLK>                                    TLK>
<TLK> HELLO NODEZ. YES, I READ YOU.
                                        TLK>
                                        <TLK> I NEED THE FLANAGAN FILE.
TLK>                                    TLK>
<TLK> I DO NOT HAVE THAT FILE.
                                        TLK>
                                        <TLK> CAN YOU SEND THE O'NEILL
TLK>                                    TLK>
                                        <TLK> FILE INSTEAD THEN?
TLK>                                    TLK>
<TLK> I'LL SEND IT AT 3:00 VIA NFT.
TLK>                                    TLK>
                                        <TLK>THANKS.
TLK>
<TLK> - END OF DIALOG                   <TLK> - END OF DIALOG
```

9.3.5 Video Screen Dialog

If you use a VT52 or VT100 series display terminal in dialog mode, TLK can use a split video screen display; that is, the screen is divided into two halves. The top half displays the messages that you type and the bottom half displays the messages that the other user types. Each half operates independently and scrolls messages when filled. TLK wraps lines at terminals with transmission rates below 600 baud. Figures 9–1 and 9–2 illustrate the TLK split video screen display.

Video dialog mode operates in the same way as normal dialog mode. Video dialog mode is interactive, and TLK messages can be sent by either terminal, regardless of which terminal initiated the dialog.

For support of the TLK video screen, the target system must support the IO.WBT, IO.RPR, and IO.ATA terminal functions (see Table C–1). The target node need not support the video mode option, in which case the messages appear on the remote terminal in normal dialog mode format (see Section 9.3.1). However, both source and target nodes must support TLK's dialog mode. Also, your display terminal must be properly defined to your system using the RSX SET command:

For MCR:

```
     {/VT52 }
>SET {      } = TT: RET
     {/VT100}
```

For DCL:

```
              {VT52 }
>SET TERMINAL {     } RET
              {VT100}
```

Sending and Receiving Video Dialog Messages

To use video dialog mode, invoke TLK as you would for normal dialog mode (see Section 9.3). The procedures for initiating and accepting or rejecting a dialog (see Section 9.3.2) are the same for video dialog mode as for normal dialog mode. When the remote user accepts the dialog, both terminals automatically enter video dialog mode if they both support the video mode option.

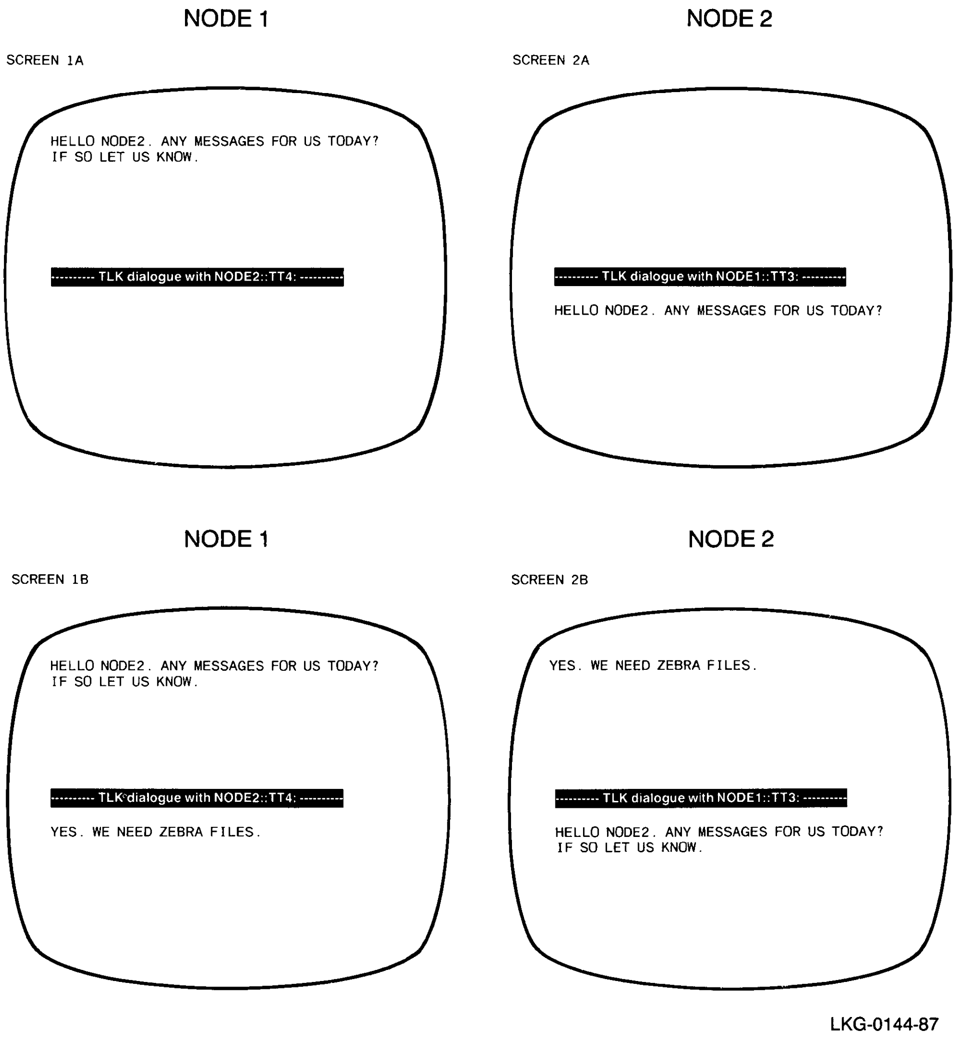
Example:

To initiate a dialog with terminal TT3: on NODE2 where both terminals are VT100s, enter

```
>
```

As in a session with normal dialog mode, TLK asks the remote terminal user to "talk" to you. If the remote user answers YES, the TLK split screen display appears on your screen, and you can begin entering messages.

Figure 9–2: The TLK Utility: Sample Split Screen Video Display



```
HELLO NODE2. ANY MESSAGES FOR US TODAY?
IF SO LET US KNOW.

---------- TLK dialogue with NODE2::TT4: ----------
```

```
---------- TLK dialogue with NODE1::TT3: ----------
HELLO NODE2. ANY MESSAGES FOR US TODAY?
```

```
HELLO NODE2. ANY MESSAGES FOR US TODAY?
IF SO LET US KNOW.

---------- TLK dialogue with NODE2::TT4: ----------
YES. WE NEED ZEBRA FILES.
```

```
YES. WE NEED ZEBRA FILES.

---------- TLK dialogue with NODE1::TT3: ----------
HELLO NODE2. ANY MESSAGES FOR US TODAY?
IF SO LET US KNOW.
```

As you enter a message, it appears on the upper half of your screen, as shown in Screen 1A of Figure 9–2. TLK sends your message line when you press RET or reach the end of the current line (for example, the 80th character on a VT100). Additional characters that you type appear on the next line. Screen 2A of Figure 9–2 illustrates how your message appears on the receiving terminal, NODE2. Notice that the receiving terminal does not receive your second line until you press RET.

Messages from the remote user appear on the bottom half of your screen. If the remote user sends a message while you are typing a line, TLK displays the incoming message on the lower half of your screen without interrupting your typing (see Screens 1B and 2B of Figure 9–2).

Terminating Video Dialog Mode

As with normal dialog mode, video dialog mode terminates when either terminal user presses CTRL/C or CTRL/Z. Either CTRL/C or CTRL/Z causes the following message to print on both terminals:

```
<TLK> - END OF DIALOG
```

Special Character Usage with Video Mode

Some terminal keys, used singly or in combination, cause special effects when used in video dialog mode. Table 9–1 describes these special effects.

Table 9–1: Special Terminal Keys for TLK Video Dialog Mode

Key	What It Does
RET or ESC	Sends the current line and repositions the cursor to the next line. TLK will not send an empty line. To send a blank line, enter a space, followed by RET or ESC.
CTRL/R or CTRL/W	Clears your screen and updates the entire display. These are useful to "refresh" the screen after a system message or broadcast overwrites part of your screen.

TLK displays all other control characters, including TAB, as a space.

9.4 Executing TLK Command Files

The TLK utility lets you create and execute command files that contain TLK single message or dialog mode messages. The maximum number of levels of indirect command files is six. DECnet–11S does not support the use of indirect command files.

9.4.1 Executing Single Message Command Files

Single message mode command/message lines that you create in a TLK command file must have the following format:

[*target-node*::][TT*n*:]'*message*

where the variable arguments have the same values as those listed in Section 9.2.1.

The following example shows the type of command/message lines that you can enter in a single message command file:

```
NODEA::TT2:'MOUNT A DECTAPE PLEASE.
TT4:'THIS IS A LOCAL MESSAGE.
NODEB::'TO NODEB'S CONSOLE TERMINAL.
'ALL DEFAULTS - LOCAL AND CONSOLE.
```

Each message in the file will be sent to the node and terminal destination specified for that message. To execute the file, you type:

TLK *@file-name*[/TR]

where

file-name is the name of the command file containing the TLK messages. The default type for a command file is CMD.

/TR is an optional trace switch. Using /TR displays each line in the file at your terminal as it executes.

Example:

```
>
```

TLK executes the command/message lines in MONGO.CMD and displays them at your terminal.

9.4.2 Executing Dialog Command Files

TLK lets you execute command files for dialog mode messages. A command file can contain an actual list of dialog messages to send to a remote terminal. Messages in this kind of file must have the following format:

target-node::TT*n*:
message
message
message
...

To execute a dialog mode message command file, use the same command syntax that you use for single message mode (see Section 9.2.1). However, if your command file does not include the target node name and terminal specification, you must specify them in the command line:

>TLK *target-node*::TT*n*: RET
<TLK> - START OF DIALOG
TLK> *@file-name* RET

A command file can also execute other command files. You can create a command file that contains a target node name and terminal specification, plus a command that executes a secondary command file. The secondary command file would contain the TLK dialog to send to that target destination. For example, the file FEIN.CMD contains the following lines:

```
NODER::TT3:   @MAYO.CMD
```

The file MAYO.CMD contains the following messages:

```
ALL MORNING FLIGHTS TO MULGRAVE,
NOVA SCOTIA HAVE BEEN CANCELLED.
CALL OUR CANADIAN OFFICE AFTER
6 P.M. TO OBTAIN INFORMATION ON
TOMORROW'S FLIGHTS.
```

When you execute FEIN.CMD, MAYO.CMD is also executed, and its dialog messages are printed on TT3: at NODER.

To execute files of this type, use the syntax that Section 9.2.1 gives for single message mode.

NOTE

> Command files invoke only nonvideo dialog mode, regardless of your terminal type.

9.5 Exit with Status

TLK exits with a status code that can be useful in an indirect command file or when TLK was spawned by or connected to a parent task.

TLK has two exit status codes:

Status	Code
TLK exited without error.	1 (EX$SUC)
TLK exited with an error.	2 (EX$ERR)

If any command in an indirect command file fails, TLK issues an exit status code 2 (EX$ERR).

Refer to RSX system documentation for information about testing status in indirect command files and testing return status in a parent task.

A
Status and Error Messages

This appendix lists the status and error messages displayed by the DECnet–RSX user utilities. The messages are listed in the following order:

- Network Control Program (NCP) messages (Section A.1)
- Network File Transfer (NFT) utility messages (Section A.2)
- File Transfer Spooler (FTS) utility messages (Section A.3)
- Network Command Terminal (NCT) utility messages (Section A.4)
- Remote Terminal (RMT) utility messages (Section A.5)
- Phone Communications (PHO) utility messages (Section A.6)
- Terminal Communications (TLK) utility messages (Section A.7)

A.1 NCP Messages

The first section of NCP messages lists error messages specific to DECnet–RSX. The second section of NCP messages lists standard error messages for all DECnet implementations. For a complete list of NCP error messages, refer to the *DECnet–RSX Guide to Network Management Utilities*. For information on using NCP and descriptions of NCP commands, refer to Chapter 2 of this manual.

A.1.1 RSX System Specific Error Messages

NCP – Error reading command

NCP could not read a command from the terminal or a command file because of a hardware error or some other unrecognized command read error condition.

NCP – File privilege violation

The file that you specified with the TO option in a SHOW command is write protected against NCP users.

NCP – HELP file error *-nn*

An error occurred while reading one of the NCP HELP files from LB:[1,2]. The variable *nn* is an FCS error code.

NCP – Invalid device name syntax

The device name in a command file or a TO file specification did not have valid syntax.

NCP – Invalid directory syntax

The directory identification (UIC) in a command file or TO file specification did not have valid syntax.

NCP – Invalid file name syntax

The file name in a command file or a TO file specification did not have valid syntax.

NCP – No HELP available

NCP could not find the help file. Normally, the help files for NCP are copied to LB:[1,2] as a result of network generation (see the *DECnet–RSX Network Generation and Installation Guide*).

NCP – Unrecognized device or unit

The RSX operating system could not find the device name or unit number in a command file or TO file specification.

NCP – Unrecognized file, device, or directory

The RSX operating system could not find the file name, device name, or directory (UIC) in a command file or a TO file specification.

A.1.2 Standard DECnet Error Messages

NCP – Invalid identification format

You used improper syntax in identifying a component of the operation. For example, you may have included a non-alphanumeric character in a node name.

The error message identifies which component caused the error, such as the line, node, and so forth.

NCP – Invalid parameter grouping

You specified too many parameters for a single command.

NCP – Listener connect failed

The logical link from NCP to the network management listener could not be established. This error message generally has one of the following secondary messages:

Access control rejected

The remote node or network management listener could not understand or would not accept the access control information.

Invalid node name format

The executor rejected the format of the remote node name. The name may contain illegal or too many characters.

Local node shut down

The executor node is shutting down and will not accept new logical link connections.

Network resources

Either the local or remote node had insufficient network resources to connect the logical link.

No response from object

The network management listener did not respond. This can occur due to slow response or abnormal termination.

Node unreachable

No path exists to the remote node, or the remote node is not currently on the network.

Object too busy

The remote network management listener had insufficient available resources to handle the connect request.

Remote node shut down

The remote node is shutting down and will not accept new logical link connections.

Unrecognized node name

The destination node name does not correspond to any known node address.

Unrecognized object

The remote node does not have a network management listener.

NCP – Listener link disconnected

The logical link from NCP to the network management listener was unexpectedly disconnected. This error message will have one of the following error details:

Node or object failed

The network aborted the logical link because the remote node or the network management listener terminated abnormally.

Node unreachable

The network aborted the logical link because it could no longer find a path to the remote node.

NCP – Operation failure

The requested operation failed. For some RSX system specific failures, one of the following extra text messages may be included.

Insufficient buffer space for additional aliases

You requested a display of aliases, but there were more aliases than the network management software could internally buffer.

System not loaded

NCP cannot perform the requested function because the DECnet system has not been installed and the network has not been loaded. See the system manager to make sure that the system is loaded.

NCP – Parameter missing

You omitted a parameter that the requested operation requires. The message tells what type of parameter was missing.

NCP – Parameter not applicable

You included a parameter inappropriate to the specified operation on the component. The message tells what type of parameter caused the error.

NCP – Privilege violation

You do not have sufficient privilege for the requested operation.

NCP – System-specific management function not supported

The requested operation is RSX system specific and not supported by the network management listener.

NCP – Unrecognized command

The command that you entered is not an NCP command.

NCP – Unrecognized component

The specified component does not exist. This message tells what type of component caused the error, such as line, node, and so forth.

NCP – Unrecognized function or option

The executor does not implement the requested operation.

NCP – Unrecognized keyword

NCP does not know one of the keywords in a command.

NCP - Unrecognized parameter type

The executor does not implement one of the specified parameters. The message tells what type of parameter caused the error.

NCP - Unrecognized value

NCP does not know a specified parameter value.

NCP - Value out of range

A parameter value that you specified is not in the range that NCP accepts.

A.2 NFT Messages

This section lists error messages that the Network File Transfer (NFT) utility displays when an error occurs during a file operation. These messages only apply to using NFT from MCR. NFT services to DCL are internal; when you use DCL for file operations, DCL reports any errors that occur. For information on using NFT and descriptions of NFT commands, refer to Chapter 4 of this manual.

NFT error messages fall into three categories:

- **Command error messages** pertain to the NFT interface to the network or operating system. These include I/O errors to the command device and errors caused by trying to run NFT on a system without enabled network support (see Chapter 4).

- **Primary error messages** pertain to I/O errors that occur when you transfer or submit a file to another node. When NFT encounters a file I/O error, it displays an error message. The message specifies the file name and the operation being performed when the error occurred. A secondary message generally gives the cause of the error.

  When a command line specifies multiple operations, NFT tries to execute subsequent operations, even after an error has occurred. During file transfer errors, NFT deletes the output file, except when appending files.

- **Secondary error messages** follow primary error messages and provide specific information on the error. There are two error types: network errors and file I/O errors. The message may be text or a DAP, FCS, or NSP error code.

The following messages are responses to a command line containing an incorrect node name in the input file specification:

```
NFT - Error in opening input file UTICA::FIL.CMD
      Unrecognized node name
```

The first line is the primary error message, telling you what NFT was doing when the error occurred. The second line is the secondary error message, isolating the cause of the error.

The next sections alphabetically list the NFT error messages, according to the three categories of errors.

A.2.1 Command Error Messages

NFT – Command file error

An error occurred when NFT retrieved a command line from an indirect command file. The command line might have exceeded the 80-character limit.

NFT – Get command line error

NFT encountered an error while retrieving a command line from a terminal or opening a command file specified from a terminal. The command line might have exceeded the 80-character limit.

NFT – Help file error: *n*

An error occurred when you tried to use the help operation. The variable *n* is a standard RSX I/O error code.

NFT – No help available

NFT could not find the help files. Normally, the help files for NFT are copied to LB:[1,2] as a result of network generation (see the *DECnet–RSX Network Generation and Installation Guide*).

NFT – No such command file

The command file you specified cannot be found.

NFT – Syntax error

Either NFT cannot determine what operation you requested, or the command line is ambiguous. Try the command line again.

NFT – Syntax error in file name *file-descriptor*

The file name you specified did not conform to RSX file specification syntax.

NFT – Unable to access network

NFT was invoked but did not find network support.

A.2.2 Primary Error Messages

Primary error messages indicate the source of an error and are followed by secondary messages, which further define the nature of the error.

NFT – Error in changing file protection *file-descriptor*

NFT – Error in closing input file *file-descriptor*

NFT – Error in closing output file *file-descriptor*

NFT – Error in deleting file *file-descriptor*

NFT – Error in getting record from file *file-descriptor*

NFT – Error in opening input file *file-descriptor*

NFT – Error in opening output file *file-descriptor*

NFT – Error in purging output file *file-descriptor*

When an error occurs during a file operation, NFT tries to delete the output file being created. If that fails, NFT displays the preceding message.

NFT – Error in putting record in file *file-descriptor*

NFT – Error in reading directory *file-descriptor*

NFT – Error in renaming file *file-descriptor*

This message reflects a problem with an input or output file name. When the rename operation is local, NFT displays the incorrect file descriptor. When the rename operation is remote, NFT displays only the input file descriptor, whether the error is in the input or output file name.

NFT – Error in spooling file *file-descriptor*

NFT – Error in submitting file *file-descriptor*

A.2.3 Secondary Error Messages

Secondary error messages give further information on the cause of network errors and file I/O errors.

Network errors pertain to network operation failures or rejections. For example, the network might reject connect requests due to insufficient network resources or an invalid node name format in the file descriptor.

File I/O errors pertain to file input or output errors on the local or remote node. Examples of file I/O errors are improper file descriptor specification or file read errors.

Network Error Messages

Aborted by network management

Either the network (NSP) rejected an attempted connect, or an operator or program used network management to abort the connected logical links.

Access control rejected

The network (NSP) rejected an attempted connect because the remote node or the object could not understand or would not accept the access control information. This occurs when the user ID, password, and account do not match those of a valid account on the remote node.

Channel already active

The specified channel is already in use.

Connection rejected by network

NFT's attempt to connect to the remote server task was rejected by the network.

Connection rejected by remote FAL

NFT's attempt to connect to the remote server task was rejected by the remote server task.

Data overrun

A protocol error. Either the remote FAL sent a packet too large for the NFT buffer, or NSP delivered a packet too large for the allocated buffer. The received data was truncated to fit in the available buffer space.

Insufficient network resources

The network (NSP) rejected an attempted connect due to insufficient network resources on either the local or remote node.

Internal consistency error

Network File Access Routines (NFARs) error occurred within NFT. Submit a software performance report (SPR) documenting the error. See the *DECnet–RSX Release Notes* for instructions on filing an SPR.

Internal inconsistency

An error occurred during error processing. Submit a software performance report (SPR) documenting the error. See the *DECnet–RSX Release Notes* for instructions on filing an SPR.

Invalid node name format

The network (NSP) rejected an attempted connect because of an invalid remote node name format. You may have used illegal characters or too many characters in the name.

Invalid object name format

The network (NSP) rejected an attempted connect because the node did not understand the object name format.

Network already open

This is an internal error; if it occurs, file a software performance report (SPR). See the *DECnet–RSX Release Notes* for instructions on filing an SPR.

Network error code: *n*

The specified error occurred during a network operation. The value for *n* is a signed octal number. If this error occurs, file a software performance report (SPR). See the *DECnet–RSX Release Notes* for instructions on filing an SPR.

Network not open

The issuing task is not part of the network; that is, it was never opened.

NFT buffer allocation failure

NFT's dynamic buffer space ($$FSR1) was exhausted during the operation. NFT must be relinked with a larger $$FSR1 section.

No more connections at remote FAL

The remote node currently has its maximum number of connections. Reissue the command after waiting for other connections to finish.

No response from remote FAL

Either the network (NSP) rejected an attempted connect because there was no response from the object, or a connected logical link was aborted because the remote node or object terminated abnormally.

Node unreachable

Either a logical link could not be connected, or a connected logical link was aborted because no path to the remote node existed.

Remote disconnect or line failure

A file transfer operation failed because the line to the remote node was disconnected or never established.

Remote FAL not found

The File Access Listener (FAL) is not installed on the remote node.

Remote node shut down

The logical link could not be connected because the remote node was shutting down and refused new logical link connections.

Request terminated

The logical link was disconnected with the request outstanding.

Resource allocation failure

Resources needed for the requested operation are not available.

Tasks out of synchronization

A problem occurred in communications between NFT and the remote server task (FAL). The problem caused FAL to send NFT an unexpected message.

Unrecognized node name

The network (NSP) rejected an attempted connect because the destination node name did not correspond to any known node address.

File I/O Error Messages

These file I/O messages can originate from either the local or the remote node.

Allocation failure on device

Disk space was not available to create the file according to the allocation method you specified. For example, disk space may be available, but not for a contiguous file. When disk space is unavailable for any files, you get a device full error.

Bad device name

The specified device name is not according to the target system's syntax.

Bad directory syntax

The specified directory is not according to the target system's syntax.

Bad file name

The specified file name is not according to the target system's syntax.

Bad record size

The specified record size is invalid or illegal for the specified operation.

Bad version number

The version number is out of range or has been illegally specified in a delete or rename operation.

DAP error code (macro:micro) = *xx:yyyy* [STV = *n*] *secondary error message*

This message appears when no other specific error message can be provided. The error code (*xx:yyyy*) is reported as unsigned octal numbers, as defined in Appendix C of the *DECnet–RSX Programmer's Reference Manual.*

The STV (secondary status value) may be returned by the remote RMS FAL, depending upon the system, to provide additional information about the error. The variable *n* is an octal number. See the RMS reference manual for the appropriate system for information on the meaning of the *n* values.

The secondary DAP error message further defines the type of error indicated by the macrocode *xx* and can be one of the following:

Data transfer error

An error occurred on a data transfer operation.

File close error

An error occurred on a file close operation.

File open error

An error occurred on a file open operation.

Function not supported by remote FAL

The remote FAL does not support the requested operation.

Invalid message field

The contents of the DAP message field contain an error. Refer to the microcode for more information.

Message format error

The DAP message format contains an error. Refer to the microcode for more information.

Message out of sync

The DAP message sequence contains an error. Refer to the microcode for more information.

Operation successful

The operation succeeded, but with a microcode of other than 225 (octal).

Data overrun

A protocol error. Either the remote FAL has sent a packet too large for the NFT buffer, or NSP delivered a packet too large for the allocated buffer.

Device full

The output file cannot be created because disk space is unavailable on the output device. You might get this message when you specify a contiguous file and contiguous space is unavailable; some systems report this with the allocation failure on device message.

Device off line

The specified file cannot be accessed because the device on which it resides is off line or not mounted.

Device write locked

The specified output file cannot be created because the output device is write locked.

Directory not found

The directory does not exist on the node/device that you specified.

End of file detected

An unexpected end of file (EOF) has been detected.

Fatal hardware error

The specified file transfer was terminated due to an unrecoverable hardware error.

FCS error code: -*n*.

The specified local file system error occurred during an operation to a local file. The number reported is a signed decimal value. See the *IAS/RSX–11 I/O Operations Reference Manual*.

File access failed

The specified file cannot be opened; it is probably locked.

File accessed for write

The specified file cannot be opened because it is currently being modified.

File already exists

The specified file name is already assigned to a file.

File already open

The specified file is already open.

File improperly closed

The specified input file is locked.

File locked by other user

Another user currently has the specified file locked or open for write.

File read error

An irrecoverable error has occurred while reading on the device.

File write error

An irrecoverable error has occurred while writing on the device.

Illegal record encountered

A record that is illegal for this operation has been encountered.

Invalid device or unit

The specified device or unit is not known to the target system.

Invalid or unsupported file organization

The file organization value (sequential, relative, or indexed) is invalid or unsupported by NFT or the remote FAL.

Invalid or unsupported record attributes

The file's record attributes (FORTRAN, implied CR/LF, embedded, or VMS print file) are invalid or unsupported by NFT or the remote FAL.

Invalid or unsupported record format

The file's record format (fixed, variable, VFC, or stream) is invalid or unsupported by NFT or the remote FAL.

Invalid wildcard operation

A wildcard is inappropriate for the specified operation. For example, you might have used a wildcard for a partial field on a /RE operation. See Section 4.1.2.1 for information on using wildcards.

No such file(s)

The specified file or files do not exist.

Privilege violation

You specified an operation for which you do not have privileges.

Rename using two different devices

You tried to rename a file on two different devices; this is an illegal operation.

Request terminated

An I/O operation has been prematurely terminated.

Resource allocation failure

Resources for the requested operation are unavailable. The primary error indicates whether the error is on the input or the output file.

Send/receive error

A network error occurred during a transfer operation.

Spool or submit command file failure

An error on a spool (/SP), submit (/SB), or execute (/EX) operation was caused by problems between the remote FAL and the print spooler or command file/batch submission task. This error may happen when the spooler or the command submission task is not installed.

System directive error

A system directive failed while trying to perform an I/O operation.

Unsupported access function

The requested DAP access function is not supported by the remote FAL. Valid DAP access functions are open, create, rename, delete, directory, submit, and execute.

Unsupported file access request

The value of the file access field (FAC) is not supported. The FAC field contains the block I/O access request bit, which may not be supported by the remote FAL.

Unsupported file option

The value of the file options field (FOP) is not supported by the remote FAL. Some of the FOP field values may not be supported by the remote FAL. These include those that represent the contiguous file, maximum version, spool on close, submit on close, and delete on close options.

Unsupported file organization

The file organization value (sequential, relative, or indexed) is valid but unsupported by either NFT or the remote FAL.

Unsupported record attributes

The file's record attributes (FORTRAN, implied CR/LF, embedded, or VMS print file) are valid but unsupported by either NFT or the remote FAL.

Unsupported record format

The file's record format (fixed, variable, VFC, or stream) is valid but unsupported by either NFT or the remote FAL.

Wildcard syntax error

There is an error in a wildcard specification; see Section 4.1.2.1.

A.3 FTS Messages

This section describes the messages that the File Transfer Spooler (FTS) utility displays when an error occurs during processing of a user request. For information on using FTS and FTS commands, refer to Chapter 5.

FTS displays one of the following messages if it encounters an error while processing a user request:

FTS -- Bad indirect file name

The indirect file name did not conform to the operating system's conventions. RSX–11M systems prefix the file specification with the at sign (@).

FTS -- Command line too long

An FTS command line cannot exceed 256 characters. You can continue the physical input line with the RSX–11M continuation marker (–), but the actual command cannot exceed the limit.

FTS -- Error in accessing work area

FTS was unable to open or write user request files to the work file area. Notify the system manager to set the work area directory to unprotected.

FTS -- Error in communicating with alias task

FTS attempted to translate an alias node name to a real node name by issuing a QIO to the network management driver, and the QIO failed.

FTS -- Error in date-time value

You specified one or more illegal values with the /AF switch. Check the syntax of the time and date argument in the switch specification.

FTS -- Error in destination file *file-descriptor*

An error occurred in accessing the specified local file. Check the output file descriptor to see if it is correct. The variable *file-descriptor* specifies the incorrect string.

FTS -- Error in getting command line

FTS found an error while retrieving a command line from the terminal. Retry the command and notify the system manager if repeated failures occur.

FTS -- Error in parse

A system error occurred while FTS was trying to check a command line for syntax. Retry the command and notify the system manager if repeated failures occur.

FTS -- Error in source file *file-descriptor*

An error occurred in accessing the specified local file. Check the input file descriptor. The *file-descriptor* specifies the incorrect string.

FTS -- File I/O error -- *FATAL*
secondary error message

FTS encountered an error in processing a local file. The secondary error message gives the cause of the error, as follows:

FCS error code: -*n*.

A local file system error occurred during an operation involving a local file. The number reported is a signed decimal value (see the *IAS/RSX-11 I/O Operations Reference Manual*).

No such file

The specified local input file does not exist.

Privilege violation

You do not have privileges for the operation you specified.

QMG error code: *n*

The error specified in the primary error message occurred during an operation involving the queue manager (QMG). Refer to Table A–1 for the values for *n*, and to the *RSX–11 Utilities Manual* for more detailed information.

FTS -- Help file error: *n*

An error occurred during an attempt to use the help function. The variable *n* is a standard RSX I/O error code.

FTS -- Illegal command

A command line without file descriptors must specify one of the following options: /ID, /HE, /LI, /ED, or /DF.

FTS -- Illegal equals sign

FTS format allows only one equals sign in a command line.

FTS -- Illegal use of switch

A command line contains a switch that is incompatible with the command type.

FTS -- Illegal wildcard

FTS allows wildcard specifications only for local input files.

FTS -- Incompatible switches

A command contains incompatible options.

FTS -- Initialization error -- *FATAL*

FTS failed to start up due to error responses to executive or network directives.

FTS -- Job does not exist

The FTS job number that you specified is not currently in the pending request queue.

FTS -- Maximum indirect file depth exceeded

FTS allows only three indirect files in an indirect file. Check your indirect file to see if you have exceeded the maximum.

FTS -- No help available

FTS could not find the help file. Normally, the help files for FTS are copied to LB:[1,2] as a result of network generation (see the *DECnet–RSX Network Generation and Installation Guide*).

FTS -- No input files

FTS transfer command format requires at least one input file.

FTS -- No queue file space available

The FTS queue is temporarily full.

FTS -- No requests found

No user requests are queued for the node name and/or UIC that an /LI switch specified.

FTS -- Open failure on command file

FTS could not open the specified command file.

FTS -- QMG communication error
QMG error code: *n*

FTS has encountered an unexpected error while attempting to communicate with the queue manager. Notify the system manager.

This error message is accompanied by the secondary error message QMG error code: *n*. This message indicates that the specified queue manager error (*n*) occurred during the queuing of an FTS user request. The reported number is an unsigned decimal value. Table A–1 lists the error codes.

FTS -- QMG marked for exit

The system manager has begun shutting down the queue manager, disabling the queuing of FTS requests.

FTS -- QMG not installed

FTS was unable to communicate with the queue manager. Notify the system manager.

FTS -- Queue does not exist

The FTS queue has not been properly initialized. Notify the system manager.

FTS -- Syntax error

The command line is not in a correct FTS format.

FTS -- Syntax error in option argument

A switch argument does not conform to FTS format.

FTS -- Too many files for command type

Certain FTS commands do not allow multiple input files.

FTS -- Too many files in command

FTS failed to find sufficient dynamic work space to process the command.

FTS -- Too many output files

FTS transfer command format allows only one output file. To specify multiple file transfers, omit the file name and type.

FTS -- Unable to access network -- *FATAL*

FTS was started up and did not find network support.

FTS -- Unable to communicate with alias task

FTS attempted to translate an alias node name to a real node name by issuing a QIO to the network management driver, and the QIO failed.

Queue Manager Error Codes

A queue manager error that occurs during queuing of an FTS user request displays the following message:

QMG error code: *n*

The value of *n* can be any of the decimal values listed in Table A–1.

Table A–1: Queue Manager Error Codes

Error Code	Meaning
1.	Illegal function code
43.	Privilege violaton
63.	Bad processor name
64.	Queue exists already
65.	Processor exists already
66.	Queue does not exist
67.	Processor does not exist
68.	Queue marked for delete
69.	Processor marked for delete
70.	Queue directory full
71.	Processor directory full
72.	Queue and processor are not the same type
73.	Job exists already
74.	Job does not exist

(continued on next page)

Table A-1 (Cont.): Queue Manager Error Codes

Error Code	Meaning
75.	No queue file space available
76.	I/O error on queue file
77.	Redundant operation
78.	Illegal argument value
79.	Executive directive failure
80.	Spooled device bad type
81.	Spooled device does not exist
82.	Spooled device driver off line
83.	Spooled device not loaded
84.	Spooled device redirected
85.	Spooled device not available
86.	Reserved
87.	No pool space
88.	Prototype processor task not installed
89.	Request failure on processor task
90.	QMG marked for exit
91.	Immediate device busy
92.	Bad queue name
93.–99.	Reserved
100.	Virtual terminals not supported
101.	Entry is not a job entry
102.	Operation inconsistent with job state
103.	Reserved
104.	Illegal or nonexistent intermediate device
105.	Intermediate device not mounted

A.4 NCT Messages

This section describes messages that the Network Command Terminal (NCT) utility displays when errors occur. For information on using NCT and descriptions of NCT commands, refer to Chapter 6.

NCT has two types of error messages:

- Connect errors are returned by either your local node or the host node, depending on the reason for the error.

- Disconnect errors occur when the network aborts a logical link to the host node.

A.4.1 Connect Errors

NCT -- No more LUNs

Your system has no available outgoing ports. You can try again. A port will be available when another user ends a remote session.

NCT -- Terminal is already attached

Your terminal is attached by another task. To free up your terminal, find out what task is running and abort the task.

NCT -- No pool available

No system pool is available for you to use NCT. You can wait awhile and try again.

NCT -- Connection rejected, Invalid node name format

The logical link could not be connected because the host node name format was incorrect. A node name is invalid if it contains illegal characters and/or is too long.

NCT -- Connection rejected, Rejected by object

The logical link could not be connected because the remote object rejected the connect.

NCT -- Connection rejected, Network modules not installed

The required CTERM modules are not installed on your system. Consult your system manager.

NCT -- Connection rejected, Node unreachable

The logical link could not be connected because no path to the host node existed.

NCT -- Connection rejected, No response from object

The logical link could not be connected because the remote object did not respond. The object either responded too slowly or terminated abnormally.

NCT -- Connection rejected, Access control rejected

The logical link connection was rejected because the network partner could not validate the access control information it received.

NCT -- Connection rejected, Local node shutting down

The logical link could not be connected because the network on the local node was shutting down.

NCT -- Connection rejected, Object too busy

The logical link connection was rejected by the network partner because the remote object was too busy handling other logical links.

NCT -- Connection rejected, Remote node shutting down

The logical link could not be connected because the network on the remote node was shutting down.

NCT -- Connection rejected, Unrecognized node name

The logical link could not be connected because the local node could not map the destination node name to any known node address.

NCT -- Connection rejected, Insufficient network resources

The logical link connection was rejected due to insufficient network resources on either the local or remote node.

NCT -- Connection rejected, Unknown reject reason = *reason*

This error is returned in rare cases of unexpected unknown errors. Try the connect again. If the error persists, you can submit a software performance report (SPR). See the *DECnet–RSX Release Notes* for instructions on submitting an SPR.

NCT -- CTERM not available on host

The CTERM host protocol is not available on the host node. NCT will attempt to use RMT as an alternative method of connecting to the host.

If your node supports RMT, you receive the following message:

```
NCT -- Using RMT
```

If your node does not support RMT, you receive the following message:

```
NCT -- RMT not available on local node
```

Note that RMT cannot connect to a VMS system. If NCT tries to use RMT as an alternative way to connect to a VMS host, an error will occur.

A.4.2 Disconnect Errors

NCT -- Connection aborted, Incompatible CTERM versions

The versions of CTERM on your local node and on the host node cannot communicate.

NCT -- Connection aborted, CTERM protocol error

A CTERM protocol error occurred on either the local or host node. You can submit a software performance report (SPR). See the *DECnet–RSX Release Notes* for instructions on submitting an SPR.

NCT -- Connection aborted, Remote node or object failed

The logical link was aborted because the remote node or object terminated abnormally.

NCT -- Connection aborted, Link aborted by object

The logical link was aborted because the remote object terminated abnormally.

NCT -- Connection aborted, Link aborted by network management

The logical link was aborted by a user or a program using network management.

NCT -- Connection aborted, Unknown disconnect reason = *reason*

The logical link was aborted due to some rare, unexpected, and unknown reason. In this case, you may want to submit a software performance report (SPR). See the *DECnet–RSX Release Notes* for instructions on submitting an SPR.

NCT -- Connection aborted, Node unreachable

The logical link was aborted because the existing path to the host node was broken.

A.5 RMT Messages

This section explains the messages that the Remote Terminal Utility (RMT) displays when an error occurs. When one of these errors occurs, RMT automatically logs you off the host system and disconnects you from the host node. For information on using RMT, refer to Chapter 7.

RMT -- Cannot access RMT service task (RMTACP)

The local node service task (RMTACP) was not installed.

RMT -- Cannot access terminal

A terminal input error has occurred.

RMT -- Command syntax error

You entered the node name in an incorrect format.

RMT -- Illegal RMT terminal device

The terminal from which you invoked RMT is not a TT: device.

RMT -- Initialization failure

RMT failed to go through its start-up initialization properly.

RMT -- Network access failure

RMT could not open the network.

RMT -- No connections available at remote node

The host node has reached its maximum number of connections and cannot make the connection you requested.

RMT -- No more remote connections available

The number of local users currently using RMT has reached the limit.

RMT -- No such node

RMT could not connect to the specified host node. Either the node is not in the network, or its name has not been defined by the system manager.

RMT -- Recursive use of RMT not allowed

The RMT task cannot be invoked on the host node.

RMT -- Remote connection aborted

The connection to the host node was terminated abnormally. This error usually indicates that the link was aborted.

RMT -- Remote connection already established

The terminal from which you invoked RMT already has an RMT connection to a host node.

RMT -- Remote host incompatibility error

The host service task (RMHACP) version is not compatible with RMT.

RMT -- Remote server not installed on remote node

RMT successfully connected to the host node, but the remote terminal host service task (RMHACP) was not installed.

RMT -- Resource allocation failure

System resources were unavailable for use.

RMT -- Unable to communicate with node

RMT could not successfully connect to the host node.

A.6 PHO Utility Messages

PHO displays informational messages that tell you about the status of the PHO utility and error messages that explain any problems that you may have while using the PHO utility. For information on using the PHO utility and descriptions of PHO commands, refer to Chapter 8.

Informational messages describe PHO's normal conditions, such as another user's attempts to call you, PHO's current status, and the commands entered by conversation participants.

Error messages tell you about problems that may occur when you are using PHO.

A.6.1 Informational Messages

user-name* has set up a conference call with *user-name.

The person with whom you are conversing has included another individual in the conversation.

***user-name* just hung up the phone.**

The person with whom you were conversing has terminated the conversation.

***node-name*::*user-name* is phoning you on *node-name*::**

A user identified by a DECnet *node-name*::*user-name* is attempting to establish a PHO conversation with you on your local node, specified by *node-name*::. You can answer, reject, or ignore the call.

Attempting to answer...

PHO is processing your command to answer an incoming call. This message starts to appear when you enter the ANSWER command and stops when PHO establishes a connection with the caller.

Establishing DECnet link...

PHO has received your command to call a remote user and is currently establishing a DECnet logical link to that user. This message starts to appear when you enter a PHONE or DIAL command and stops when PHO establishes a connection to the remote PHO utility.

Press any key to cancel the directory listing and continue.

You can stop the directory listing that you requested with the DIRECTORY command by typing any keyboard key.

Press any key to cancel the help information and continue.

You can terminate the display of help information that you requested with the HELP command by typing any keyboard key.

Ringing *user-name*...(Press any key to cancel call and continue.)

PHO has established a connection with the specified *user-name*'s phone and is currently ringing that person. This message starts appearing when PHO establishes the connection with a remote PHO utility and stops when the person you are calling answers or rejects your call.

That person has answered your call.

The person you are phoning has answered your call. At this point, either you or the remote user can begin the PHO conversation.

That person has rejected your call at this time.

The person that you are phoning does not want to answer your call and has entered a REJECT command.

That person just canceled the call.

The person with whom you were conversing has hung up the phone.

That person's phone is busy right now. Try again later.

The "phone" (or terminal), of the person you called is in a state that prevents it from receiving an incoming call; the terminal may be attached to another task, for example.

Sending mail...

PHO is currently sending the mail message that you specified with the MAIL command.

A.6.2 Error Messages

Extraneous text *text* ignored on command line.

In processing a command, PHO is ignoring some text that you included in the command. The message displays the extraneous text, shown here as *text*.

That person's phone just went dead. Try again.

The logical connection to the node of the person you called was lost. Try the call again.

An error has occurred in remote communication.

PHO received an error in issuing a network directive.

An error has occurred in transmitting to *node-name*

An error occurred when PHO issued a network transmit directive to the remote node of the person you phoned, shown here as *node-name*.

Help library is missing. Please inform your system manager.

HELP information is available only when your system has an installed help library. If no help library exists, inform your system manager.

Invalid command entered. Try again.

You entered a command that PHO cannot interpret. Check the spelling of the command and/or list of valid PHO commands in Table 8–2 of Chapter 8.

Invalid help keywords specified. Try HELP.

You have asked for help on a topic that PHO does not recognize. Enter the HELP command without an argument for a list of the available topics.

Invalid specification of node or person. Try again.

You have entered a node or user name that is incorrect. Check your spelling. Note that node names cannot be longer than 6 characters.

No one is calling you now.

You have attempted to answer a call when no one is phoning you. Someone who was previously calling you may have just hung up the phone.

No one is on hold now.

You have entered an UNHOLD command but no one is currently on HOLD.

No one with that name is available at this time.

The person you are phoning is not logged on. Check the spelling of the person's name and that you specified the correct node for that user. You can use the DIRECTORY command to display the names of users who are currently available for calls on a given node.

Only a node-name is allowed in a DIRECTORY command.

The correct syntax of the DIRECTORY command is as follows:

DIRECTORY [*node-name*[::]]

That person's phone is unplugged (/NOBROADCAST).

The terminal of the person you are calling is set to /NOBROADCAST. /NOBROADCAST status prevents the PHO utility from displaying messages about incoming calls.

That person's terminal cannot be used as a telephone.

The person you are calling has a type of terminal, such as a hardcopy terminal, that cannot be used for PHO conversations.

There is no room on the screen for another person.

PHO can display a maximum of six viewports at a time. You have reached that limit and cannot add another conversation to this conference call.

Try again, using the following format: "MAIL addressee MSGX"

You have issued an incorrect MAIL command. The error message gives the MAIL command's correct format.

You cannot place redundant calls.

You can have only one PHO conversation at a time with a particular person.

You did not specify a person to call. Try again.

You omitted from a DIAL command the name of the person to call. The format for DIAL is as follows:

PHONE DIAL [*node-name*::]*user-name*

Your phone is ringing; you cannot place an outgoing call.

You cannot phone someone while an incoming call is pending.

Unrecognized object at *node-name*

PHO is not installed, or the object is improperly set up at the remote node, shown here as *node-name*. Inform the remote node's system manager.

You cannot use this type of terminal as a telephone.

You cannot use PHO from certain types of terminals, such as hardcopy terminals.

The network has not been properly loaded or started.

For you to use PHO, the network must be loaded and active. Consult your system manager, or wait and try again.

A.7 TLK Messages

The Terminal Communications (TLK) utility displays messages when it cannot start a dialog or complete a message transmission. The first section of TLK messages describes dialog mode failure messages; the second section describes message transmission error messages. For information on using the TLK utility and descriptions of TLK commands, refer to Chapter 9.

A.7.1 Dialog Mode Failure Messages

The following messages give reasons for failure to start a dialog that you requested:

TLK -- Dialog mode rejected by remote user

The remote user answered "no" to TLK's question asking for a dialog.

TLK -- Terminal not accessible

The remote user failed to respond to the request for a TLK dialog within the 10-second time limit.

TLK -- Terminal busy

The remote user's terminal is attached by a task for exclusive use.

A.7.2 Message Transmission Error Messages

If TLK cannot complete a message transmission that you requested, it displays one of the following messages:

TLK -- Command line read error

TLK detected an error in reading the command line. Check the command format and try again.

TLK -- Connection aborted

The network aborted the connection to the target terminal. There may have been a line failure between the two nodes or a remote node crash. Also, the network software on either node may be turned off.

TLK -- Dialog mode not supported

You requested TLK dialog mode but your system supports only single message mode.

TLK -- Dialog mode not supported by remote TLK

The remote TLK server task does not support dialog mode. Use single message mode.

TLK -- Dialog mode rejected by remote user

The request for TLK dialog mode was rejected by a user of a system other than RSX or IAS.

TLK -- Error in connecting to remote TLK

TLK could not establish a connection to the remote TLK server task. The remote node may be unreachable, for example.

TLK -- Invalid TLK device

TLK cannot be invoked from a non-TT: device, such as HT:, if you are using RMT.

TLK -- No more connections at remote TLK

The target node has reached its maximum number of connections. Try again later.

TLK -- No such command file

The system could not find the command file that your command line specified.

TLK -- No such node

You specified a target node name that does not exist in the network and is not a known alias node name.

TLK -- Remote TLK not installed

The TLK server task is not installed or is not supported in the remote system.

TLK -- Syntax error

The command line has a syntax error. For example, you might have omitted the apostrophe that precedes a single message after a defaulted device specification, or the double colon that follows the target node name.

TLK -- Terminal busy

The target terminal is busy. It may be engaged in a dialog or attached by another task.

TLK -- Terminal not accessible

The target node's operating system cannot access the terminal that your command line specified. The target terminal might be dedicated to another job, or you might have used the wrong number.

TLK -- Terminal not logged on

The target terminal is not logged on to the remote system.

TLK -- Unable to access network

The network software on the node initiating TLK is not in a usable state (not installed or not running).

B

FTS Log File Messages

This appendix explains the completion messages that the File Transfer Spooler (FTS) sends to a log file when it completes a request. The appendix shows the format of the message, explains each component of the message, and gives examples of FTS commands and the resulting completion messages.

Completion Message Format and Components

A completion message has the following format:

```
node-name    type    Job = queue-job-number
Time:   dd-mmm-yy start-time end-time
User:   user-id    Blocks:  size
Status:   status    [error-code disposition]
File:  file
Input:   input
```

where

node-name identifies the remote node. If there is no remote node, *node-name* identifies the local node. If the request was a remote-to-remote file transfer, *node-name* identifies the destination node.

type is a character code that identifies the type of user request. The codes are summarized in Table 5–2 in Section 5.2.12.

queue-job-number is the job number of the request as listed by the /LI switch (see Section 5.2.12).

dd-mmm-yy

is the day, month, and year when the processing of the user request started.

start-time

is the time in hours, minutes, and seconds (*hr:mn:ss*) when the processing of the user request started.

end-time

is the time in hours, minutes, and seconds (*hr:mn:ss*) when the processing of the user request completed.

user-id

is a 1- to 16-character string that identifies the owner of the processed user request.

size

is the number of user blocks transferred over the logical link. The value for *size* is 0 for execute and delete operations or for instances when errors occurred to prevent file transfer. For a concatenation request, *size* is the sum of the input file sizes.

status

tells whether or not the user request completed successfully. If the request completed successfully, the success message appears in the status field, and the error code and disposition field are omitted. If the request did not complete successfully, the status field displays error = *nodename*. This message indicates where the error occurred. LOCAL indicates an error at the local node; *errnodnam* is the name of a remote node on which the error occurred.

You also get information in the error code and disposition fields when a request has been unsuccessful.

error-code

is a 6-character DAP, NSP, or FCS error code returned when an operation fails. The next paragraphs describe the types of error messages that you can get. In these messages, *nnn* is a positive decimal number defining a specific set of error codes that each program returns.

FCS*nnn*

describes an error in accessing the local file. In this case, *nnn* is the positive decimal value of the F.ERR byte in the local FDB. See the *IAS/RSX-11 I/O Operations Reference Manual* for a list of these codes.

NET*nnn*

describes an error in accessing the remote node. For NET, *nnn* refers to the low-order byte of the second word of the network I/O status block. Refer to the *DECnet–RSX Programmer's Reference Manual* for a list of these codes.

N*nnn*07

appears when the network rejected an attempted connect. For this category, *nnn* is the value of the NSP rejection code. 07 signifies that the connection was rejected. See the *DECnet–RSX Programmer's Reference Manual* for a list of disconnect or reject reason codes.

SND002

is the network send/receive error. For example, the system may be unable to request execution of an indirect command file because the task that submits the file to the system command handler is not installed or is not available.

FSB*nnn*

indicates an FTS buffer allocation error. For FSB, *nnn* refers to the number of the file descriptor that would have caused the file specification block buffer to overflow, had it been stored. The disposition code is F2.

The number of files per user request is set to 20(decimal). This size can be increased by rebuilding the task FTSDEQ and extending the $$FSB1 PSECT in NFT's data area.

xxyyyy

describes the DAP errors. The value *xx* is the octal value of the macro error code, maccode. Maccode is the global code specification for the error. The value *yyyy* is the miccode, which provides the specific code for the error. See the *DECnet–RSX Programmer's Reference Manual* for the values and meanings of *xxyyyy*.

REM*nnn*

describes all other errors that can occur. For REM, *nnn* is the value of the low-order byte of the first word in the network I/O status block. See the *DECnet–RSX Programmer's Reference Manual* for the values and meanings of *nnn*. (Note that *nnn* is never 003 or 004, since 003 is recorded as a DAP error and 004 as an NSP error.)

disposition

is a short message summarizing the completion status of an unsuccessful request. The message can be one of the following:

Discarded

The request was discarded but not requeued.

Requeued

The request was requeued with an /AF time value (see preceding description).

Queue hold

After being requeued a maximum number of times, a request was placed in a hold queue for manual release or abort.

Workfile access

The request failed because an error occurred while accessing the user request file from the queue entry.

Privileged information access

The request failed due to an error while accessing information on your access privileges to the local file specifications.

Buffer allocation

The request failed due to a lack of buffer space to store the file specifications (see the preceding FSB*nnn* error code).

file

identifies the output file, or the only file in the request when input and output have no distinction. The file descriptor is in the form

node-name::*dev*:[*uic*]*file-name.type*;*version*

Only the output file descriptor is displayed for a delete or execute operation. The node name is omitted for local file descriptors.

input

identifies the input file in the form

node-name::*dev*:[*uic*]*file-name.type*;*version*

The input file descriptor is repeated for concatenation requests involving multiple input files. The node name is omitted if the descriptor is for local files.

The following examples show the command lines and completion messages for single file copy, copy with concatenation, and remote-to-remote copy requests.

Example 1: Single file copy

```
FTS>

DENVER   TRL    Job=217444
Time:  19-APR-86    17:36:54    17:50:05
User:  [200,200] Blocks:  0
Status:  error=DENVER N03907 Requeued
File:  DB0:[200,1]INVENTORY.LST;4
Input:  DENVER::DK5:[200,200]INVENTORY.LST;4
```

One file was queued to transfer to the local system. FTS uses the local user's current log-in defaults to store the file. The completion message shows that the request was requeued because of an error on the remote system. The error code, N03907, identifies the error as no path to destination node.

Example 2: Copy with concatenation

```
FTS>

DALLAS   CLR    Job=382077
Time:  28-JUN-86    11:05:03    11:35:03
User:  [224,1] Blocks:  25
Status:  Success
File:  DALLAS::DK0:[225,225]TAXES.LST
Input:  DB0: [224,1]FICA.LST
        DB0:[224,1]FEDERAL.LST
        DB0:[224,1]STATE.LST
```

FTS transferred three input files and concatenated them to form the output file. The absence of a disposition code tells you that the transfer was successful. A file descriptor appears for each input file.

Example 3: Remote-to-remote copy

```
FTS>

BOSTON   TRR    Job=217526
Time:  12-DEC-86    13:15:04    13:45:05
User:  [200,1] Blocks:  0
Status:  error=BOSTON N03407 Discarded
File:  BOSTON::DK2:[224,223]PEN.INV
Input:  DENVER::DK3:[200,1]PEN.INV
```

This example illustrates a completion message for a remote-to-remote transfer that terminated in an error. The error code, N03407, indicates that the network rejected the connection.

C

QIO Functions Supported by the HT: Driver

This appendix is for programmers who write applications that may access HT: terminal drivers. It points out the differences in how the HT: and TT: terminal drivers handle QIO functions.

Table C–1 lists standard and device specific QIO functions that the HT: driver can execute over an RMT line. Each function must also be supported by the local terminal driver. For details on each function's parameters and use, refer to the *RSX–11M/M–PLUS I/O Driver's Reference Manual.* The last column of Table C–1 compares how the HT: driver and TT: driver perform the function.

C.1 HT: QIO Subfunctions

You can select many of the device specific functions in Table C–1 by using subfunction bits. You can select one or more subfunctions by performing an OR operation on their relative bits in a QIO macro. The *RSX–11M/M–PLUS I/O Driver's Reference Manual* outlines allowable function/subfunction combinations; this manual also lists additional subfunction codes. The HT: driver supports the following RSX–11M/M–PLUS subfunctions:

TF.AST
TF.BIN
TF.CCO
TF.NOT
TF.RAL
TF.RCU
TF.RNE
TF.RST
TF.TMO
TF.WAL
TF.WBT
TF.XCC

The HT: driver does not support TF.ESQ and TF.XOF.

Table C-1: HT: QIO Functions

Function	Description	Comments
Standard Functions		
IO.ATT IO.DET	Attach device Detach device	On TT: drivers, ATT and DET occur synchronously without fail. On HT: drivers there is a time lag across the network; specify an event flag to ensure that the task is synchronous with the function. Also check for error returns, as failures can occur.
IO.KIL	Cancel I/O requests	Same as for TT: driver. The connection may be aborted if the host system is low on pool.
IO.RLB	Read logical block (read typed input into buffer)	Same as for TT: driver.
IO.RVB	Read virtual block (read typed input into buffer)	Subfunction bits are not cleared as they are with TT: drivers.
IO.WLB	Write logical block (print buffer contents)	Same as for TT: driver.
IO.WVB	Write virtual block (print buffer contents)	Subfunction bits are not cleared as they are with TT: drivers.
Device Specific Functions		
IO.ATA	Attach device; specify unsolicited input character AST	See comments for IO.ATT.
IO.CCO	Cancel CTRL/O (if in effect); then write logical block	Same as for TT: driver.
SF.GMC	Get multiple characteristics	See Sections C.2 and C.3.
IO.GTS	Get terminal support	Same as for TT: driver.
IO.RAL	Read logical block; pass all bits	Same as for TT: driver.
IO.RNE	Read logical block; do not echo	Same as for TT: driver.
IO.RPR	Read logical block after prompt	Same as for TT: driver.
IO.RST	Read logical block ended by special terminators	Same as for TT: driver.

Table C-1 (Cont.): HT: QIO Functions

Function	Description	Comments
IO.RTT	Read logical block ended by specified terminator table	Same as for TT: driver.
SF.SMC	Set multiple characteristics	See Sections C.2 and C.3.
IO.WAL	Write logical block; pass all bits	Same as for TT: driver.
IO.WBT	Write logical block; break through any I/O conditions at terminal	Same as for TT: driver, except that a long WBT message may be broken into two or more pieces. For example, if a BRO program on the host writes a WBT message while you are typing input, the message may be broken into pieces and your input retyped between pieces.

C.2 Restrictions on Half Duplex Terminal Drivers

The SET CHARACTERISTIC command may not always work on HT: drivers if either the host or the local terminal driver is half duplex. The results also depend on whether the SET command originates as a QIO from a task or an MCR SET /*characteristic* command. Table C-2 shows the results in both cases. To be safe, write your programs to run in accordance with your local terminal driver.

NOTE

The SET /FDX command is not supported by any HT: driver.

Table C-2: HT: SET Command Support

Set Form	Local TTDRV[1]	Host TTDRV[1]	Supported Commands
Issued as QIO	Full duplex	Full duplex	All SET commands[2]
	Full duplex	Half duplex	All SET commands[2]
	Half duplex	Full duplex	Commands in Table A only[3]
	Half duplex	Half duplex	Commands in Table A only[3]
Issued as MCR SET */char*	Full duplex	Full duplex	All SET commands[2]
	Full duplex	Half duplex	Commands in Table B only[3]
	Half duplex	Full duplex	Commands in Table C only[3]
	Half duplex	Half duplex	Commands in Table B only[3]

[1] RSX-11M-PLUS always has full duplex TTDRV.
RSX-11S always has half duplex TTDRV.
RSX-11M has a generation option for TTDRV.

[2] The HT: driver does not support SET/SPEED (TC.RSP and TC.XSP).

[3]

Table A	Table B	Table C
TC.HFF	SET/BUF	SET/BUF
TC.HLD	SET/LOWER	SET/CRT
TC.NEC	SET/PRIV	SET/ECHO
TC.PRI	SET/SLAVE	SET/FORMFEED
TC.SCP		SET/HOLD
TC.SLV		SET/LOWER
TC.SMR		SET/PRIV
TC.TTP		SET/SLAVE
TC.WID		SET/TERM

C.3 HT: Status Returns

Table C-3 lists the error and completion status codes that the HT: driver can return. The HT: driver does not return any SE.*xxx* codes for SF.GMC and SF.SMC functions. Refer to the *RSX-11M/M-PLUS I/O Driver's Reference Manual* for more information.

Table C–3: HT: Status Returns

Code	Meaning
IS.SUC	Successful completion
IS.CR	Successful completion on a read
IS.ESC	Successful completion on a read
IS.PND	I/O request pending
IS.TMO	Successful completion on a read
IE.EOF	Successful completion on a read with end of file
IE.ABO	Operation aborted
IE.BAD	Bad parameter
IE.DAA	Device already attached
IE.DNA	Device not attached
IE.DNR	Device not read
IE.IFC	Illegal function
IE.NOD	Buffer allocation failure
IE.OFL	Device off line
IE.PRI	Privilege violation
IE.SPC	Illegal address space

Index

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA
call 800–DIGITAL

In Canada
call 800–267–6215

In New Hampshire
Alaska or Hawaii
call 603–884–6660

In Puerto Rico
call 809–754–7575 x2012

ELECTRONIC ORDERS (U.S. ONLY)

Dial 800–DEC–DEMO with any VT100 or VT200
compatible terminal and a 1200 baud modem.
If you need assistance, call 1–800–DIGITAL.

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: A&SG Business Manager

INTERNATIONAL

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

Internal orders should be placed through the Software Services Business (SSB)
Digital Equipment Corporation, Westminster, Massachusetts 01473

*Any prepaid order from Puerto Rico must be placed
with the Local Digital Subsidiary:
809–754–7575 x2012

DECnet–RSX
Guide to User Utilities
AA–H223F–TC

READER'S COMMENTS

What do you think of this manual? Your comments and suggestions will help us to improve the quality and usefulness of our publications.

Please rate this manual:

	Poor				Excellent
Accuracy	1	2	3	4	5
Readability	1	2	3	4	5
Examples	1	2	3	4	5
Organization	1	2	3	4	5
Completeness	1	2	3	4	5

Did you find errors in this manual? If so, please specify the error(s) and page number(s).

____

____

____

____

General comments:

____

____

____

____

Suggestions for improvement:

____

____

____

____

Name ______________________________ Date ______________

Title ______________________________ Department ______________

Company ______________________________ Street ______________

City ______________ State/Country ______________ Zip Code ________

DO NOT CUT – FOLD HERE AND TAPE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY LABEL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

digital™

Networks and
Communications Publications
550 King Street
Littleton, MA 01460–1289

DO NOT CUT – FOLD HERE

digital

Printed in U.S.A.