Networks • Communications

DECnet-RSX

Guide to User Utilities

digital software

DECnet-RSX

Guide to User Utilities

Order No. AA-H223D-TC

September 1985

This *DECnet-RSX Guide to User Utilities* instructs the terminal user in the use and application of the DECnet-RSX user utilities. This manual also describes QIO functions supported by local and host terminal drivers for the user/programmer.

Supersession/Update Information:	This is a new manual.
Operating System and Version:	RSX-11M V4.2 RSX-11S V4.2 RSX-11M-PLUS V3.0 Micro/RSX V3.0
Software Version:	DECnet-11M V4.2 DECnet-11S V4.2 DECnet-11M-PLUS V3.0 DECnet-Micro/RSX V1.0

digital

AA-H223D-TC
First Printing, September 1985

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.

Copyright © 1985 by Digital Equipment Corporation

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

The following are trademarks of Digital Equipment Corporation:

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

Ethernet is a trademark of Xerox Corporation.

This manual was produced by Networks and Communications Publications.

Contents

3 The Network File Transfer Utility (NFT)

4 The File Transfer Spooler Utility (FTS)

5 The Network Command Terminal Utility (NCT)

6 The Remote Terminal Utility (RMT)

7 The Phone Communications Utility (PHO)

8 The Terminal Communications Utility (TLK)

A QIO Functions Supported by the HT: Driver

Figures

Tables

Preface

The *DECnet-RSX Guide to User Utilities* describes procedures for determining node, line, and circuit states, for setting alias node names, for transferring files, for accessing the resources of remote DECnet nodes, and for communicating with other terminal users. This manual also discusses QIO functions supported by the HT: driver.

Intended Audience

This manual is intended for the DECnet-RSX terminal user. Appendix A applies, more specifically, to the user/programmer and lists the QIO functions supported by the HT: driver.

Structure of This Manual

Chapter 1 Outlines the DECnet-RSX user utilities and presents a sample application of the utilities in a terminal session.

Chapter 2 Describes the Network Control Program (NCP) which allows you to display node, line, and circuit information at your terminal and to set alias node names.

Chapter 3 Describes the Network File Transfer utility (NFT) which allows you to perform remote file operations such as file transfers between your local DECnet-RSX node and any other DECnet node that supports file operations.

Chapter 4 Describes the File Transfer Spooler utility (FTS) which allows you to perform the same file transfer and manipulation operations as NFT, while queuing and processing your requests according to specified dates and times.

Chapter 5 Describes the Network Command Terminal utility (NCT) which allows you to log on to any host operating system that supports network command terminals and use the host system's resources

Chapter 6 Describes the Remote Terminal utility (RMT) which allows you to access the resources of a remote DECnet-RSX node

Chapter 7 Describes the Phone Communications utility (PHO) which allows you to engage in an interactive dialog with a remote user on a DECnet-RSX, DECnet-VAX, or PRO/DECnet system. (PHO is available only for RSX-11M-PLUS and Micro/RSX users)

Chapter 8 Describes the Terminal Communications utility (TLK) which allows you to send single line or dialog messages to other terminal users that support the TLK utility.

Appendix A Describes the 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*, you should 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
`Monospace`	Monospace type depicts examples of user input or system output. User input is shown in red; system output is shown in black. For example, `NCP> HELP SHOW NODE`
UPPERCASE	Uppercase letters indicate characters that you must type exactly as shown.
italics	Italics represent variables for which you must supply actual values.
[*opt-argument*]	Square brackets enclose optional arguments. You can either include the item shown in brackets or omit it. DO NOT type the brackets in a command line.
{*option1* / *option2*}	Braces indicate that you must choose ONE of the options contained within them. DO NOT type the braces in a command line.
<CTRL/*x*>	<CTRL/*x*> indicates a control character keying sequence. The CTRL key on your keyboard and the appropriate character key should be pressed simultaneously. For example, to enter a <CTRL/C>, hold down the CTRL key and type a C at the same time. In displays, control characters are shown as ^*x*. For example, when you press <CTRL/C>, `^C` is displayed on your terminal.
<*KEY*>	This symbol indicates that you should type the specified key. For example, the <RET> symbol indicates that you should type the RETURN key.
`>NCP <RET>` `NCP>`	Most examples in this manual represent terminals in MCR mode. However, you need only type a utility name in response to an MCR or DCL prompt in order 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 provide the following capabilities:

- **Examination of node, line, and circuit states.** The Network Control Program (NCP) allows you to display node, line, and circuit information at your terminal. The information available for a local node includes the current state of the node and the name and address of the node. The information available for a remote node includes the type of node, the current status of the node, and the identification of all lines and circuits associated with the node. NCP also provides information on the state of a specific line or circuit. Chapter 2 explains NCP.

- **Specification of alias node names.** NCP also allows you to assign or change alias node names. Alias node names are abbreviated pseudonyms that you can specify in order to simplify a node name or to simplify node name and access control specification (see Section 2.7).

- **File transfer and manipulation.** The Network File Transfer (NFT) and the File Transfer Spooler (FTS) utilities allow you to transfer files to another user either on the same node or on a remote node. In addition, NFT and FTS provide 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. (The basic difference between NFT and FTS is that FTS queues and processes user requests according to user-supplied dates and times, while NFT executes user commands interactively.) Chapters 3 and 4 contain complete explanations of NFT and FTS capabilities, respectively.

NFT is accessible to both MCR and DCL users. FTS has support for MCR users, only.

- **Access to remote host node resources.** The Network Command Terminal utility (NCT) allows you to access the resources of remote nodes which support the Terminal Services Architecture (TSA). NCT enables you to log on to remote systems and communicate with all programs run under the host operating system. NCT is available on RSX-11M-PLUS systems, RSX-11M/RSX-11S V4.2 (or greater) systems, and Micro/RSX systems. Chapter 5 explains the NCT utility.

  The Remote Terminal utility (RMT) allows you to access the resources of remote DECnet-RSX nodes. RMT logically connects your terminal to a specific 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 6 explains the RMT utility.

- **Communication with DECnet-RSX, DECnet-VAX, and PRO/DECnet users.** The Phone Communications utility (PHO) allows you to engage in an interactive dialog with up to six users, concurrently. PHO allows you to "phone" any user on your local node, or on a remote DECnet-RSX, DECnet-VAX, or PRO/DECnet node. PHO provides many advanced features that are not available with the TLK utility (see the description which follows). PHO is available only on RSX-11M-PLUS and Micro/RSX systems. Chapter 7 explains the PHO utility.

  The Terminal Communications utility (TLK) allows you to send a single line message or to engage in a dialog with another terminal user. TLK enables you to communicate with another user on your local DECnet-RSX node or on a remote node that supports the TLK utility. Chapter 8 explains the TLK utility.

1.2 Sample Session with the DECnet-RSX User Utilities

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

Initially, you use NCP to confirm that remote node YUKON is "reachable" from your node.

```
>NCP<RET>
NCP>SHOW NODE YUKON<RET>

Node summary as of 14-JAN 84 13 12

Remote                          Active                          Next
Node                 State      Links     Delay     Circuit     Node

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

NCP> <CTRL/Z>
>
```

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

```
> PHONE <RET>

                RSX-11M PLUS Phone Facility      15-AUG-84

%DIRECTORY YUKON <RET>
Press any key to cancel directory listing and continue.
------------------------------------------------------------

Process Name          User Name      Terminal     Phone Status

BRAUNSTEIN            BRAUNSTEIN     unusable     ---
Larry                 AUGUSTUS       TT2          available
Joe Smith             SMITH          TT4          available
DALEY                 DALEY          TT7          available
Dave Garrod           GARROD         TT9          available
IAN                   IAN            TT12         /nobroadcast
Terri Buckley         BUCKLEY        TT18         available
Bob Fraser            FRASER         TT13         available

8 persons listed

<CTRL/Z>
>
```

After typing <CTRL/Z> to exit from PHO, you use TLK in dialog mode to converse with user Smith on node YUKON (another option would have been for you to "phone" Smith while you were still using the PHO utility, see Chapter 7). Once TLK successfully connects with YUKON and user Smith indicates a willingness to "talk" with you, you receive the <TLK> - START OF DIALOG message on your terminal. You can then engage in a dialog with Smith, instructing him to load pack FRED onto drive DR0:.

```
>TLK YUKON ·TT4:<RET>

<TLK> - START OF DIALOG
TLK>HELLO, IS AN RK06 DISK DRIVF AVAILABLE?<RET>

<TLK> YES.  YOU CAN USE DR0

TLK> PLEASE LOAD PACK FRED ONTO DRIVE DRO:.  BYE.<RET>
TLK> <CTRL/Z>
<TLK> - END OF DIALOG
```

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

```
>SET /HOST=YUKON <RET>

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

>HELLO<RET>

ACCOUNT OR NAME: JEFF<RET>
PASSWORD:   <RET> (the password does not echo)

RSX-11M-PLUS V3.0 [1,54] SYSTEM

GOOD MORNING
14-SEP-84 11:18 LOGGED ON TERMINAL RTO:

>MOUNT DRO:FRED<RET>
```

You can then use NCP to set the alias (H) for your original node name, HOME, and your password and account information, JEFF/SECRET. Use of the alias will simplify node and access control specification in the NFT command line used to transfer files to YUKON.

```
>NCP SET ALIAS H DESTINATION HOME/JEFF/SECRET<RET>
```

Next, you use NFT to transfer files from your original node, HOME, to your current node, YUKON, so that the files can be task built there. (You need not specify any output node name or directory; the output files will default to your current node and directory.) Then, you check the YUKON directory for the existence of the special library files used for assembling and task building files.

```
>NFT<RET>
NFT>=H::TEST1.MAC,TEST2.MAC,*.CMD<RET>
NFT><CTRL/Z>

>PIP DR0:[1,1]SPCLIB.OLB/LI<RET>

DIRECTORY DR0:[1,1]
14-JAN-84 13:18

SPCLIB.OLB       31.       C 22-NOV-84
```

Now you are ready to assemble and task build the files. (Remember, with NCT, your terminal is logically connected to YUKON.)

```
>MAC @TESTASM<RET>
>TKB @TESTBLD<RET>
```

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

```
>DISMOUNT DR0:<RET>
DMO -- RT0: DISMOUNTED FROM DR0: *** FINAL DISMOUNT ***
>BYE<RET>
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) allows you to obtain node, line, and circuit information and to set alias node names. NCP is a DECnet utility program that is normally used by the system manager to control the network. This chapter describes a subset of NCP commands that may be useful to RSX-11M/M-PLUS terminal users. RSX-11S users should refer to Chapter 1 of the *DECnet-RSX Guide to Network Management Utilities* for a description of NCP support for RSX-11S systems. For a complete discussion of all NCP capabilities refer to the *DECnet-RSX Network Management Concepts and Procedures* manual.

NCP now offers DCL support for the SHOW ACTIVE NODES command (see Section 2.4.6 for the DCL command syntax).

2.1 Invoking and Exiting NCP

To invoke NCP, type NCP after the MCR or DCL prompt and 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, type a <CTRL/Z> or specify 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 type NCP commands, you can abbreviate any keyword to the first three letters. For example, the format for the NCP command used for obtaining 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 commmands (see Section 2.4.2).

2.3 Obtaining Assistance in Selecting NCP Commands

If you need assistance in selecting NCP commands, use the HELP command. If you want to obtain a summary of all NCP commands, type

```
NCP>HELP<RET>
```

The HELP command can be used with a specific NCP command or component to display the relevant information about that command or component, for example:

```
NCP>HELP SHOW<RET>
NCP>HELP SHOW NODE<RET>
```

For the HELP command to work, the NCP HELP files must be located on LB:[1,2]. If they are not in the proper place, you will get the following message:

```
NCP - No HELP Available
```

If you get this message, ask your system manager to place the HELP files on 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 shown here:

Example:

```
NCP>SHOW<RET>
Show component: NODE<RET>
Node identification: YUKON<RET>
```

If you need more assistance about the choices available to you, press the <ESC> key in response to any of the NCP prompts, and relevant information will be displayed.

2.4 Obtaining Node Information

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

All NCP SHOW commands cause output to be displayed at your terminal. However, you can direct command output to an output file at your local node (see Section 2.4.1, which follows).

2.4.1 Directing NCP Command Output to an Output File

To direct command output to an output file at your local node, append the keyword TO, followed by the output file specification, to the NCP command. For example, to direct the SHOW NODE YUKON command output to the file YUKON.LST, type

```
NCP>SHOW NODE YUKON TO YUKON.LST<RET>
```

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

You can also use the Network Display utility (NTD) 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. Node names may not be abbreviated in NCP commands. (However, you can create alias node names, which can simplify node name specification, as described in Section 2.7.1.)

The format for a node address depends on whether or not a node has multiarea support. If a node has multiarea support, the node address consists of an area number identifying that node's area within the multiarea network, and a unique decimal integer assigned to the node at network generation (for example, 4.19 YUKON). If a node does not have multiarea support, the node address consists of an area number of 1 and a unique decimal integer (for example, 1.18 BASIN). Multiarea 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 those users with multiarea networks, the examples for the NCP SHOW ACTIVE NODES and SHOW KNOWN NODES commands contain multiarea node addresses.

In NCP commands, if you specify a node address for a multiarea node but do not supply a specific area number for that node, the area number that is returned 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 — In this state, communications paths (logical links) can be established with other nodes. This is the normal operating state.

SHUT — When the system manager (or any other privileged user) sets the executor to this state, the network closes down gracefully. This means that no new logical links are permitted, but existing links are allowed to complete. When all links are complete and disconnected, the node is set to the OFF state.

OFF — In this state, your node is isolated from the network. No logical links can exist and any previous links have been destroyed.

2.4.4 Obtaining Local Node Information

If you want to know the name of the node on which you are operating (the executor node), along with the node address and state and the system identification, use the SHOW EXECUTOR command, as shown here:

```
NCP>SHOW EXECUTOR<RET>
```

Example of a display:

```
Node summary as of 23-NOV-84 10:36

Executor node = 1.18 (HIRDAL)

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

2.4.5 Determining if a Specific Remote Node Is Reachable

If you want to know if a specific node is reachable (that is, whether a communication path can be established between that node and your node), use the NCP SHOW NODE command. All the 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.

For example, if you plan to send a TLK message to remote node YUKON or to use NFT to transfer files to YUKON, you can first check to see if YUKON is reachable. To do so, enter

```
NCP>SHOW NODE YUKON<RET>
```

Example of a display:

```
Node summary as of 23-NOV-84 10:46

Remote                          Active                          Next
Node                 State      Links      Delay    Circuit     Node

1.19 (YUKON)         Reachable  0          4        DMC-1       1.22 (YENHAM)
```

This display example reveals that YUKON is reachable, meaning that connections to YUKON can be attempted.

If your node is a nonrouting node, remote node state information is not available. You must have the SHOW NODE command executed at a reachable full-routing node (in a multiarea network, the full-routing node must reside in the same area as the node for which you are requesting information). To do so, use the TELL command shown below and detailed in Section 2.6. (See Section 2.4.8 for information on determining adjacent nodes and adjacent node types.) The format for the NCP TELL command is:

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

where

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

node-id specifies the name or address of the node for which you want information displayed.

In the following example, your node (HOME) is a nonrouting node, and you want to determine if remote node YUKON is reachable. Therefore, you must have the SHOW NODE command executed at node NEXT, a full-routing node accessible to node HOME:

```
NCP>TELL NEXT SHOW NODE YUKON<RET>
```

The information displayed at your terminal will reveal whether YUKON can be reached by your node.

2.4.6 Determining All Reachable Nodes

A reachable node is a node to which the executor node can establish a communication path. NCP will display names, addresses, and other information for all reachable nodes in the network. The SHOW ACTIVE NODES command is useful when you are planning to access several nodes (for example, when using NFT to make transfers to several nodes). This command displays information regarding all reachable nodes so that you do not have to use the SHOW NODE command separately for each node. (However, if your network is large, you may prefer to use the SHOW NODE command only for those specific nodes that you are interested in.)

NCP offers MCR and DCL support for the SHOW ACTIVE NODES command. Both the MCR and DCL command syntax are shown below:

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

Example of a display:

```
Active nodes summary as of 23-NOV-84 11:02

Executor node = 6.18 (HIRDAL)

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

Remote                      Active                     Next
Node            State       Links   Delay  Circuit    Node

2.2 (GRIBEL)    Reachable     2      4     DMC-2      6.17 (REXHUB)
2.3 (YUKON)     Reachable     0      4     DMC-1      8.22 (YENHAM)
4.7 (WOLERY)    Reachable     0      18    DMC-0      4.7 (WOLERY)
4.8 (GAMMA)     Reachable                  DMC-0      9.21 (ROCK)
4.9 (MARVIN)    Reachable                  DMC-0      11.30 (SCRIBE)
6.10 (FORD)     Reachable                  DMC-0      8.24 (CHIPS)
9.15 (QUASAR)   Reachable                  DMC-0      11.16 (WAREN)
```

2.4.7 Determining the Nodes Known to Your System

NCP will display all nodes whose names are known to your node and whether or not these nodes are currently reachable (or active, as defined in Section 2.4.6). The known nodes display format is identical to the display generated by the SHOW ACTIVE NODES command. To determine the nodes known to your system, enter

```
NCP>SHOW KNOWN NODES<RET>
```

Example of a display:

```
Known nodes summary as of 23-NOV 84 14:21

Executor node = 6.18 (HIRDAL)

  State = On, Identification - HIRDAL DISTRIBUTED SYSTEMS
  Active links = 0

Remote                      Active                    Next
Node           State        Links   Delay  Circuit    Node

2.1 (ORION)    Unreachable   0       6                2.1 (ORION)
2.2 (GRIBEL)   Reachable     2       4     DMC-2      6.17 (REXHUB)
2.3 (YUKON)    Reachable     0       4     DMC-1      8.22 (YENHAM)
4.6 (ZEUS)     Unreachable                            8.26 (RDVAX)
4.7 (WOLERY)   Reachable     0       18    DMC-0      4.7 (WOLERY)
4.8 (GAMMA)    Reachable                   DMC-0      9.21 (ROCK)
4.9 (MARVIN)   Reachable                   DMC-0      11.30 (SCRIBE)
9.10 (FORD)    Reachable                   DMC-0      8.24 (CHIPS)
9.12 (KREMIT)  Unreachable                            8.23 (VAXWRK)
```

Note that nodes 9.12 (KREMIT), 4.6 (ZEUS), and 2.1 (ORION) are unreachable.

2.4.8 Determining Adjacent Nodes

An adjacent node is a node that is either physically connected to your node or on the same Ethernet as your node. (All nodes on an Ethernet are considered adjacent to one another.) To determine all nodes adjacent to your node, use the NCP SHOW ADJACENT NODES command:

```
NCP>SHOW ADJACENT NODES<RET>
```

Example of a display:

```
Adjacent nodes status as of 10-NOV-84 15:22:07

Remote                        Active                        Next
Node           State          Links     Delay   Circuit     Node

1.6 (GRIBEL)   Reachable 1              3       UNA-0       1.6 (GRIBEL)
1.9 (ORI)      Reachable                        UNA-0       1.9 (ORI)
1.20 (REX)     Reachable                        UNA-0       1.20 (REX)
1.22 (YENHAM)  Reachable                        DMC-2       1 22 (YENHAM)
1.24 (CHIPS)   Reachable                        UNA-1       1.24 (CHIPS)
1.30 (SCRIBE)  Reachable 1              2       DMC-0       1.30 (SCRIBE)
1.32 (VAX4)    Reachable                        UNA-0       1.32 (VAX4)
```

To determine the node type for all nodes adjacent to your node, use the NCP SHOW ADJACENT NODES STATUS command:

```
NCP>SHOW ADJACENT NODES STATUS<RET>
```

Example of a display:

```
Adjacent nodes status as of 10 AUG 84 16 23 07

Remote                           Active
Node                 State       Links   Delay   Type             Cost    Hops    Circui

1.3                  Reachable                   Routing IV       3       1       UNA-1
   Next node = 1.3
1.6 (GRIBEL)         Reachable                   Nonrouting IV    3       1       UNA-0
   Next node = 1.6 (GRIBEL)
1.8 (GAMMA)          Reachable   1       3       Area             3       1       UNA-0
   Next node = 1.8 (GAMMA)
1.16 (WAREN)         Reachable                   Routing IV       4       1       DMC-2
   Next node = 1.16 (WAREN)
1.19 (RONDEL)        Reachable   1       2       Nonrouting IV    3       1       UNA-1
   Next node = 1.19 (RONDEL)
1.22 (YENHAM)        Reachable                   Routing IV       4       1       DMC-0
   Next node = 1.22 (YENHAM)
1.24 (CHIPS)         Reachable                   Nonrouting IV    3       1       UNA-0
   Next node = 1.24 (CHIPS)
```

2.5 Obtaining Line and Circuit Information

The following sections contain information pertaining to 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.

If you want to obtain information regarding 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

A line or circuit is identified by a unique identification that is assigned to the line or circuit 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 identification, you can type the NCP SHOW KNOWN LINES (or CIRCUITS) command. The identification of all lines (or circuits) known to your node will be displayed.

Using the KNOWN LINES or KNOWN CIRCUITS keywords with the SHOW command is one method of identifying more than one line or circuit. You can also use the ACTIVE LINES and ACTIVE CIRCUITS keywords with the SHOW command. Active lines or active circuits are all those lines and circuits known to your node that are currently turned on (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 that may be of interest to you. For a discussion of all possible 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 the 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 Determining Line and Circuit States

If you want to know the state of a specific line or circuit, use the appropriate NCP command format, shown below:

SHOW LINE *line-id*

SHOW CIRCUIT *circuit-id*

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

For example, to determine the states of line DMC-0 and circuit PCL-0.1, enter the following commands:

```
NCP>SHOW LINE DMC-0<RET>
```

Example of a display:

```
Line summary as of 23-NOV-84 16:15

Line            State
DMC-0           On


NCP>SHOW CIRCUIT PCL-0.1<RET>
```

Example of a display:

```
Circuit summary as of 23-NOV-84 16:43

Circuit = PCL-0.1

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

If you want to know the state of all lines or circuits known to the executor node and currently working, use the SHOW ACTIVE LINES or SHOW ACTIVE CIRCUITS command for example.

```
NCP>SHOW ACTIVE CIRCUITS<RET>
```

Example of a display:

```
Active circuits summary as of 23-NOV-84 17:09

Circuit = DMC-0

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

Circuit = DMC-1

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

If you want to know the state of all lines or circuits known to the executor node, use the SHOW KNOWN LINES or SHOW KNOWN CIRCUITS command:

```
NCP>SHOW KNOWN LINES<RET>
```

Example of a display:

```
Known lines summary as of 23-NOV-84 17:31

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 have an NCP command executed at a remote node by using the TELL prefix with the command. For example, if you want to know the network status of your node (that is, if you want to know how another node "sees" your node), you can have the NCP SHOW NODE command executed at another node (specifying your node name with the SHOW NODE command). The information displayed at your terminal would be the same as the information displayed at the remote node. The format for using the TELL prefix is:

TELL *node-id command*

Example:

```
NCP>TELL YUKON SHOW LINE DMC-3<RET>
```

In this example, the NCP SHOW LINE command is executed at the remote node named YUKON, but the resulting information is displayed at your terminal. The display lists the line information as maintained by YUKON.

The remote node at which the command is to be executed may require that you specify access control information. Access control information should follow the node identification, as shown in the following format:

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

where

user-id is a 1- to 16-character string identifying the user at the remote system.

passwd is a 1- to 8-character password needed to gain access to the remote system.

accnt is a 1- to 16-character string used to specify an account at the remote system. (This field is not used by RSX systems.)

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

2.7 Setting and Obtaining Information about Alias Node Names

An alias node name is a user-assigned logical name for a network node that allows you to refer to a node independently of the actual node name. You assign an alias to a specific node by entering a command at your terminal, as described in the following sections.

You can increase the flexibility of your programs by using logical node names that will also be assigned as alias node names. Therefore, each time you run a program, you can interact with a different node without having to modify the node name within the program, and therefore, without having to rebuild the program. Before running the program, you enter a command at your terminal to assign to the desired node an alias node name that is identical to the logical node name used within the program.

You can also use alias node names to abbreviate node specifications by specifying the node name and access control information within the alias.

The following sections describe the commands used to set, change, and remove alias node names and to obtain information about aliases. Further information on aliases is provided 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 the node to which the alias is to refer, plus any network access control parameters associated with this alias for that node.

Nonprivileged users can set aliases to apply to their own terminal 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 causes all references to node name ME to apply to node YUKON:

```
NCP>SET ALIAS ME DESTINATION YUKON<RET>
```

In this example, all connect requests to node ME from programs associated with your terminal will go to node YUKON.

Example 2:

In this example, without the use of alias node names, you would have to type the following command lines, each of which includes the node name MEXICO and access control information:

```
>NFT MEXICO/[1,1]/SECRET::DTO:BUENO.MAC=GOOD.MAC<RET>
>NFT MEXICO/[1,1]/SECRET::DTO:DULCE.MAC=YUKON::SWEET.MAC<RET>
>NFT MEXICO/[1,1]/SECRET::DTO:VERDE.MAC=QUEBEC::GREEN.MAC<RET>
```

To simplify data entry, you can set an alias node name (M) and specify the access control information as follows:

```
NCP>SET ALIAS M DES MEXICO/[1,1]/SECRET<RET>
```

Now, using the alias node name M, you can type these shorter versions of the same three lines shown above:

```
>NFT M::DTO:BUENO.MAC=GOOD.MAC<RET>
>NFT M::DTO:DULCE.MAC=YUKON::SWEET.MAC<RET>
>NFT M::DTO:VERDE.MAC=QUEBEC::GREEN.MAC<RET>
```

Example 3:

By using aliases, you can create a command file that will apply to any node you choose. All you have to do is use a logical node name in the command file (REM, in this example) and, before running the command file, specify the logical node name as the alias of the node for which you want the command file to apply. The following sample command file can be used to transfer test files to any remote node and to send a message to the remote node's operator.

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

To transfer all test files to node YUKON, you would assign alias REM to YUKON before executing the command file:

```
NCP>SET ALIAS REM DES YUKON<RET>
```

2.7.2 Changing and Removing Alias Node Names

To change the current definition of any 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 the test files transferred to another node (for example, to one named QUEBEC), you can use the same command file without having to modify it: just assign alias node name REM to QUEBEC

To remove a particular alias or all aliases known to your terminal, use the CLEAR ALIAS command. The format for the NCP CLEAR ALIAS command is:

CLEAR ALIAS *alias*

For example, to remove the alias ME, enter

```
NCP>CLEAR ALIAS ME<RET>
```

To remove all aliases known to your terminal, enter

```
NCP>CLEAR KNOWN ALIASES<RET>
```

2.7.3 Displaying Alias Node Names

You can display information about aliases with the NCP SHOW ALIAS command.

Displaying Your Alias Node Names

To display all your alias node names, enter

```
NCP>SHOW KNOWN ALIASES<RET>
```

Example of a display:

```
Known aliases summary as of 27-NOV-84 16:17

Alias
Name      Scope           Destination

ME        Terminal TT2:   HIRDAL
```

Displaying All System Alias Node Names

To display all aliases systemwide, enter

```
NCP>SHOW ALL ALIASES<RET>
```

Example of a display:

```
All aliases summary as of 27-NOV 84 16:25

Alias
Name      Scope              Destination

ME        Terminal TT2:      HIRDAL
B         Terminal TT4:      GRIBEL/MILGROM/...
EF        Terminal TT4:      YUKON/FAL
H         Terminal TT4:      HIRDAL/PHILPOTT/...
W         Terminal TT4:      WOLERY/DECNET/...
HIRDAL    Terminal TT10:     HIRDAL/MILGROM/...
YENHAM    Terminal TT10:     YENHAM/MILGROM/...
KRYPTN    Terminal TT10:     KRYPTN/MILGROM/...
WOLERY    Global             WOLERY/DECNET/...
KREMIT    Global             KREMIT/DECNET/...
QUASAR    Global             QUASAR/DECNET/...
```

Global aliases (such as WOLERY, KREMIT, and QUASAR in the preceding display) are aliases set by the system manager that can be used by all tasks and users. Local aliases are those set by nonprivileged users and apply only at the user's terminal (for example, alias ME may be used only at terminal TT2:).

Displaying Information about a Particular Alias

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

SHOW ALIAS *alias*

where

alias identifies a particular alias for which information is to be displayed.

For example, if you want information about the alias ME, enter

```
NCP>SHOW ALIAS ME<RET>
```

Example of a display:

```
Alias summary as of 27-NOV-84 17:07

Alias
Name      Scope            Destination

ME        Terminal TT2:    HIRDAL
```

2.8 NCP Error Messages

Some common NCP error messages are listed on the following pages in two categories: those that are specific to DECnet-RSX and those that are standard for all DECnet implementations. For a complete list of NCP errors, refer to the *DECnet-RSX Guide to Network Management Utilities*.

2.8.1 RSX System-specific Error Messages

NCP - Error reading command

NCP could not read a command from either 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 specified with the TO option in a SHOW command is write protected against the user of NCP.

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 HEI P 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 device name or unit number in a command file or TO file specification could not be found by the RSX operating system.

NCP - Unrecognized file, device, or directory

The file name, device name, or directory (UIC) in a command file or a TO file specification could not be found by the RSX operating system.

2.8.2 Standard DECnet Error Messages

NCP - Invalid identification format

The identification of the component involved in the requested operation did not have proper syntax (for example, a nonalphanumeric character was used in a node name).

This message will include the type of component in error (for example, line, node, and so forth).

NCP - Invalid parameter grouping

The parameters furnished by the user for the requested operation cannot be included in 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 will have one of the following secondary messages:

Access control rejected

The remote node or network management listener either 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 (for example, the name contained illegal characters or was too long).

Local node shut down

The executor node is in the process of shutting down and will not accept any more logical link connections.

Network resources

Either the local or the remote node had insufficient network resources to connect the logical link.

No response from object

The network management listener did not respond. This could be due, for example, to its responding too slowly or terminating abnormally.

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 resources available to handle the connect request.

Remote node shut down

The remote node is in the process of shutting down and will accept no more 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 either 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

The network management software could not internally buffer all the aliases that could be displayed.

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

The user did not supply a parameter that is necessary for the requested operation. This message will include the type of parameter missing.

NCP - Parameter not applicable

The user supplied a parameter that is not applicable to the requested operation on the specified component. This message will include the type of parameter in error.

NCP - Privilege violation

The user does not have sufficient privilege to perform the requested operation.

NCP - System-specific management function not supported

The requested operation is RSX system specific and is not supported by the network management listener.

NCP - Unrecognized command

NCP does not have the command that the user typed.

NCP - Unrecognized component

The component specified by the user does not exist. This message will include the type of component in error (for example, line, node, and so forth).

NCP - Unrecognized function or option

The requested operation is not implemented by the executor.

NCP - Unrecognized keyword

One of the keywords in a command is not known by NCP.

NCP - Unrecognized parameter type

One of the parameters given by the user is not implemented by the executor. This message will include the type of parameter in error.

NCP - Unrecognized value

A parameter value given by the user is not known to NCP.

NCP - Value out of range

A parameter value given by the user is out of the range accepted by NCP.

3
The Network File Transfer Utility (NFT)

The Network File Transfer utility (NFT) allows you to perform file operations on accessible network nodes. Certain NFT operations are specific to either the MCR or DCL command language interpreters (CLIs). (Users with DECnet-11M-PLUS and DECnet-Mirco/RSX software now have access to NFT functions through DCL.)

Because the NFT utility addresses two types of users (MCR and DCL users), this chapter is divided into three major sections:

- Section 3.1 describes the NFT file descriptor syntax, the use of wildcards in file specifications, foreign file formats, and NFT file descriptor defaults. Both MCR and DCL users should read this section.

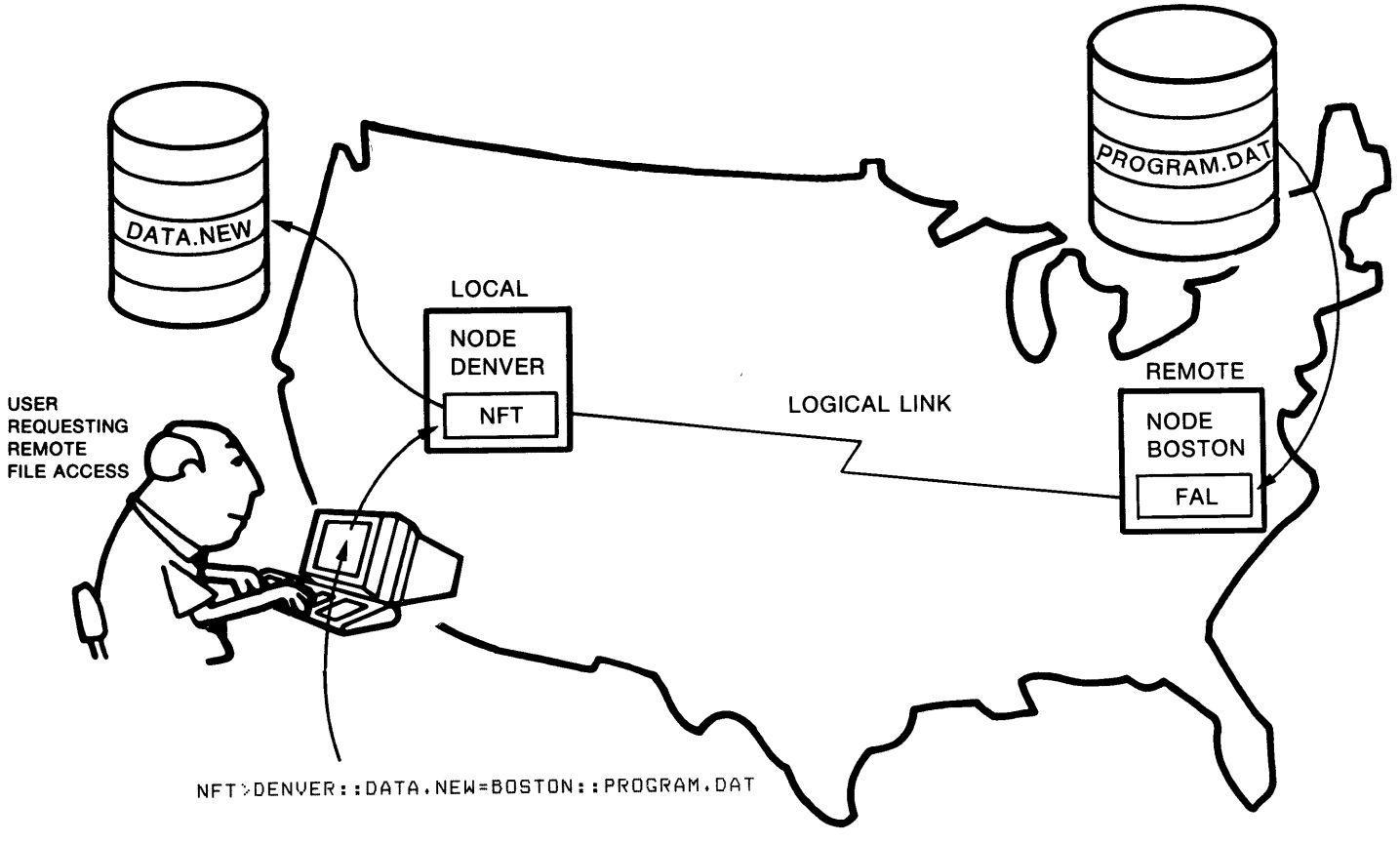
- Section 3.2 describes, in detail, the NFT operations that can be performed through the MCR command language. MCR users should read this section. (Table 3-2 summarizes the NFT operations for MCR users.)

- Section 3.3 describes, in detail, the NFT operations that can be performed through the DCL command language. DCL users should read this section. (Table 3-3 summarizes the NFT operations for DCL users.)

In DECnet-RSX systems, the NFT server task is called the File Access Listener (FAL). FAL performs whatever actions are required of the remote node to execute the file-handling requests that it receives from NFT (see Figure 3-1).

NOTE

Refer to the *DECnet-RSX Release Notes* for information regarding cross-system file operations in the heterogeneous DECnet environment.



NOTE: NFT and FAL transfer file PROGRAM.DAT from node BOSTON to node DENVER to create file DATA.NEW. In this example, access control information does not appear in the NFT command line, since it has previously been entered in an alias node name block.

TW137

Figure 3-1: The NFT Utility: File Transfer

3.1 NFT File Descriptor Syntax

NFT commands operate on file descriptors that identify files in the network A file descriptor consists of a node specification and a file specification The format for a file descriptor is.

[*node specification* .][*file specification*]

3.1.1 Node Specification

A node specification consists of the name of the node and optional access control information for that node, followed by two colons:

nodename[*access-control*]::

where

nodename is a 1- to 6-alphanumeric character name that includes at least 1 alphabetic character. If a node name is an alias that includes associated access control information, you can omit all access control fields, as they will default to the information associated with the alias. For information on how to specify alias node names, see Section 2.7

access-control is a user identification, password, and account specification (appended to a node name) used to gain access to a remote file system. Access control information can be specified using one of the two following formats.

Format 1:

/userid/passwd/accnt::

Format 2:

"*userid passwd accnt*"::

where

userid is a 1- to 16-character string identifying the user at the remote system.

passwd is a 1 to 8 character password needed to gain access to the remote file system

accnt is a 1 to 16-character string used to specify an account at the remote system. (This field is not used by RSX systems.)

NOTE

When passing access control information to case sensitive systems (for example, ULTRIX-32/ULTRIX-32m systems), you must use Format 2, shown here. (Format 1 passes access control information in UPPERCASE.) The embedded spaces in Format 2 are mandatory

Both the interpretation of the access control fields and the access control mechanism depend on the type of remote system used. If the remote system is a DECnet-RSX node supporting access control verification, the *userid* field is used to specify 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 additional 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) will permit either privileged or nonprivileged access control. For privileged access, use the privileged password that was specified for FAL during generation of the remote node (the default password is PRIV) and 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 and password are [200,200] and GUEST, respectively. The *accnt* field can be omitted for both privileged and nonprivileged access.

Examples of Node Specifications:

Examples of valid node specifications follow. Missing fields can be omitted from the right. For example, the node specification QUEBEC/[310,2]// can be written QUEBEC/[310,2]. Missing fields on the left must be marked as missing by a slash or a blank (depending upon the format used), as shown in the fourth example.

NOTE

> The use of brackets ([]) around the user ID is optional (as shown in the last two examples), if the user ID is supplied in the form of a UIC.

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

3.1.2 File Specification

A file specification identifies either the local or the remote file on which NFT is to operate. NFT can parse many foreign file specifications (see Section 3.1.4). RSX file specifications have the following format:

dev:[*ufd*]*filename.type*;*ver*

NOTE

> When you enclose a file specification in quotes, the file spec string is passed to the remote system in a case sensitive manner. This is necessary for transferring files to and from case sensitive systems such as ULTRIX-32/ULTRIX-32m systems.

RSX-11M-PLUS and Micro/RSX systems support the definition of logical names for all or part of a file specification The DECnet-RSX FAL can translate only global logical names. For a discussion of logical names and how to define them, refer to the *RSX 11M/M PLUS I/O Operations Reference Manual.*

For additional information on RSX files, refer to the *RSX-11 Utilities Manual* For information on how to specify file names that do not conform to RSX conventions, see Section 3.1.4. For a description of the use of wildcard characters within file specifiers, see Section 3.1.3.

Depending upon its position in a command string, a file specification is either an input (source) or an output (destination) file specification. Furthermore, the node names define the file specification as local or remote. Therefore, NFT files can fall into four categories:

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

Local or remote output files are files that receive data from an NFT operation, and local or remote input files are files that contain the data to be operated on and/or transmitted.

Defaults for DECnet-RSX NFT file descriptors are summarized in Table 3-1. You can change default values listed in Table 3-1 by using the set default operation (/DF). See Section 3.2.12.

Table 3-1: Summary of DECnet-RSX NFT File Descriptor Defaults

Field	Conditions	Default Value
nodename	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).
userid	The *userid* has been assigned using an alias node name. (See Chapter 2 for information on the NCP SET ALIAS command.)	Value of *userid* 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 *userid* given for the node name.
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.

(continued on next page)

Table 3-1 (cont.): Summary of DECnet-RSX NFT File Descriptor Defaults

Field	Conditions	Default Value
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 defined above.	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.)

(continued on next page)

Table 3-1 (cont.): Summary of DECnet-RSX NFT File Descriptor Defaults

Field	Conditions	Default Value
filename	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.
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.

If you are using a format that is not described here, you should enclose the foreign file directory and file specification with double quotes, for example, "*filespecification*". This directs NFT to transmit the file specification to the foreign node without checking its syntax or applying defaults for missing fields. File specifications within double quotes are not used in determining default values for output files

NOTE

When you enclose a file specification in quotes, the file spec string is passed to the remote system in a case sensitive manner. This is necessary for transferring files to and from case sensitive systems such as ULTRIX-32/ULTRIX-32m systems.

The use of wildcards in foreign file specifications is subject to the restrictions of the foreign operating system. The use of wildcards in RSX-11 file specifications is described in the following section.

3.1.3 Wildcards in NFT File Specifications

NFT allows you to type wildcards (asterisks and/or percent signs), within file specifications for existing local or remote files located on DECnet-RSX systems. Wildcard specifiers that are not accepted by NFT can be used for remote files if they are enclosed in quotes (as described in Section 3.1.2).

An asterisk (*) inserted in a field means that any value will be accepted within that field. Allowable wildcard field specifiers are listed below:

[*,*m*]*filename.type;ver*	means all group numbers will be accepted.
[*g*,*]*filename.type;ver*	means all member numbers will be accepted.
[*]*filename.type;ver*	means all named directories will be accepted.
[*g,m*]*.type;ver*	means all file names will be accepted.
[*g,m*]*filename.*;ver*	means all file types will be accepted.
[*g,m*]*filename.type;**	means all version numbers will be accepted.

In addition, except when renaming files (see the rename operation, Section 3.2.11), NFT permits wildcard specifiers for part of a field in the file name and file type, as defined here:

- An asterisk in a portion of a field means that that part of the field 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 means that any one character will be accepted in that position (nulls are not permitted). RSTS/E uses the question mark (?) rather than the percent sign to indicate 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

Multiple wildcard specifiers can be combined in a single file specifier, as shown in the following 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>ELROND/TERRI/ACCT  DB0 [100 1]=DM1 [*,10]FIL%%.MAC<RET>
```

In this sample, all files that meet the following qualifications are transferred to node ELROND:

- Files located on device DM1: on the local node
- Files listed under any group number for UFD member 10
- Files having a 5-character file name beginning with the characters FIL
- Files having the file type MAC

3.1.4 Foreign File Formats

Files that reside on non-RSX nodes are referred to as foreign files and must use syntax compatible with the systems on which they are located. NFT has the ability to parse the foreign file formats listed here:

Directory formats:		**File specification formats:**
[*ufd*]	IAS	*filename.type;ver* IAS, VMS
[*directory*]	VMS	*filename.type ver* TOPS-20, VMS
[*ppn*] or (*ppn*)	RSTS/E	*filename.type* RT-11, RSTS/E
<*directory*>	TOPS-20, VMS	*/pathname/filename* ULTRIX-32, ULTRIX-32m
(not applicable)	RT-11	
/pathname	ULTRIX-32/ULTRIX-32m	

NOTE

An ULTRIX-32/ULTRIX-32m directory is specified by a path name, such as /usr/users/buckley where "/" denotes the root.

An ULTRIX-32/ULTRIX-32m file specification must be enclosed in quotation marks in order to be passed to a remote system in a case sensitive manner.

Examples

VMS:

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

TOPS-20:

PS:[MITTON]MAIL.TXT.1439

RSTS/E:

SY:(1,4)SYSTEM.HLP

RT-11:

DK:TECO.SAV

ULTRIX-32/ULTRIX-32m

"/usr/users/buckley/main.c"

3.2 NFT Operations for MCR Users

MCR NFT operations act on primary switches which specify the operation to be performed (for example, /AP specifies the append operation). Some MCR NFT operations have optional qualifying switches that further define the action of the primary switch (for example, when /AS/AP are appended to an input file specification, the /AS qualifying switch causes the input file to be appended to an output file in ASCII record mode). The MCR NFT operations are summarized in Table 3-2. Sections 3.2.4 - 3.2.15 describe each MCR NFT operation in detail.

3.2.1 Rules for Specifying MCR Switches in a Command Line

MCR primary switches can be inserted anywhere in a command line.

MCR qualifying switches fall into one of two categories, each having different rules for insertion in a command line:

1. **Command Qualifying Switches.** Command qualifying switches do not affect input or output files and, therefore, can be specified anywhere in a command line.

2. **File Qualifying Switches.** File qualifying switches apply only to each specific input file to which they are appended. When file qualifying switches are specified for an output file, the switch (or switches) achieves global status, and becomes the default for all input file(s) in the command, unless overridden on a per file basis.

3.2.2 NFT Command Line Continuer

You can continue an NFT command line onto an additional input line by using a hyphen (-) before the line terminator (see the example below). The hyphen, followed by the terminator (<RET>), acts as a command line continuer. A command line continuer causes NFT to prompt you for the remainder of the command line. You can insert a continuer at any point in a command line, and you can use any number of continuing lines within a command.

Example:

```
NFT>NODEX/[200,30]/DECNET::DTO:[BUCKLEY]MAGIC.MAC;3=-<RET>
NFT>NODEY/200,10/PASME::SYO:[30,60]MYSTIC.MAC;2<RET>
```

Table 3–2: Summary of NFT Operations for MCR Users

NFT Operation	MCR Primary Switch	MCR Qualifying Switches	(Scope of Qualifier)
Append: The append operation adds 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: The copy operation transfers one or more existing files to one or more files on the local or remote node.	*default*	/AS	(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	(c md)
Delete: The delete operation deletes one or more files.	**/DE**	/LO	(cmd)
		/NM	(cmd)
Directory: The directory operations list a directory of one or more files.	**/LI** **/BR** **/FU** **/AT**		
Execute: The execute operation executes a command file stored on a remote or local node.	**/EX**	/LO	(cmd)
		/NM	(cmd)

(continued on next page)

Table 3–2 (cont.): Summary of NFT Operations for MCR Users

NFT Operation	MCR Primary Switch	MCR Qualifying Switches	(Scope of Qualifier)
Help: The help operation displays a descriptive message about switches and commands.	**/HE** **HELP [*subject*]**		
Identify: The identify operation identifies the version of NFT being used.	**/ID**		
Rename: The rename operation changes the name of an existing file.	**/RE**	/NV /LO /NM	(file) (cmd) (cmd)
Set defaults: The default operation sets certain default values for input and output files.	**/DF**		
Set protection: The protect operation 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: The spool operation spools 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: The submit operation copies a command file to a local or a 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)

3.2.3 Definition of MCR Qualifying Switches

/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, the file is created as a variable length format implied LF-CR attribute file.

The /AS switch should be used only if translation is desired. File transfers to remote nodes that have similar file systems (for example, VMS) are more efficiently handled in image block mode (the default transfer mode).

/AX The automatic transfer selection switch causes NFT to use its default algorithms in selecting the mode in which to transfer a file to a remote node. 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 of transfer, you will want to use the /AX switch to override a specified mode of transfer.

/BK The block mode switch transfers files using 512.-byte blocks, which is a more efficient method than the usual record mode transfer. Block mode can be used with any file organization type on transfers to other Files-11 or RMS-11 systems that support block mode transfers (for example, RSX, VMS, or the RSTS/E RMS FAL). When transferring files to or from a foreign file system using block mode, 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 version number support on the remote system. For example, RT-11 and RSTS/E do not support version numbers, and therefore, this switch would be ignored on those systems.

The negative version of the /NV switch (/-NV) specifies that an output file will retain the current version number. If another file with the same version number already exists, an error may result

/PR/SY/OW/GR/WO The protection switch (/PR) allows a file's protection status to be set or changed. File protection is provided at four levels (/SY:*p*, /OW:*p*, /GR:*p*, /WO:*p*):

- **/SY: (System)** specifies file access that you allow to the system UICs (that is, UICs with group numbers of 10 octal or less).
- **/OW: (Owner)** specifies file access that you allow to yourself.
- **/GR: (Group)** specifies file access that you allow to other members of your group.
- **/WO: (World)** specifies file access that you allow for all other UICs.

The *p* variable signifies the protection settings R, W, E, and D. The R, W, E, and D protection settings specify whether each level can read, write, extend, or delete a file, respectively. (See Section 3.2.13 for a full description of the set protection operation.)

/FO:[uic] The file owner switch allows you to specify the owner of an output file.

/RAT:MACY11 The MACY11 switch causes a file to be stored in MACY11 format on a TOPS-10 or TOPS-20 system.

/RAT:NOSPAN The no span blocks switch specifies that the ouput file will not span block boundaries.

/RC The record mode transfer switch directs NFT to transfer a file one record at a time, instead of using block mode. You would use the /RC switch when copying to unit record equipment or to override NFT's default selection of the transfer mode.

/SB The submit switch submits a file to a batch or indirect command file processor on a local or a remote node, executes the file, and then deletes it.

/SP The spool switch directs 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. (See Section 3.2.14 for a full description of the spool operation.)

/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 that are in the directory, but the file ID is new.

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

/LO The log switch displays the names of files (copied, deleted, and so forth) as the operation is performed. The log switch applies to all files involved in the current command.

/NM The no messages switch suppresses output messages that would otherwise display at your terminal during NFT operations.

/ME The merge switch merges two or more input files into one output file.

If you specify wildcards in one or more fields of the output file specification, the /ME switch causes the resulting output file specification to contain the information found in the corresponding fields of the first input file specification in the command line.

3.2.4 The Append Operation

MCR Primary Switch: /AP

The append operation adds the contents of one or more input files to the end of an existing output file. The output file retains its original attributes independent of the attributes of the input file(s). However, if the attributes of the input files are different from the attributes of the output file, the append operation will appear to succeed, but the resulting output file may not read correctly.

Format:

NFT>*outfile*[*/sw*] = *infile*(*s*)[*/sw*]/AP

where

outfile is the output file descriptor. No wildcard specifiers are allowed in the output file descriptor. The file type and record attributes are taken from the existing file. The file name and file type for the output file must be explicitly specified.

infile(*s*) is one (or more) input file descriptor(s).

/sw is one (or more) of the following qualifying switches (defined in Section 3.2.3):

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

NOTE

Refer to Section 3.2.1, which contains the rules for specifying both file and command qualifying switches in a command line.

Example:

```
NFT>LOWELL::DK1:FIL1.DAT;1=ESTER::FIL2.DAT;1/AS,YUKON::-<RET>
NFT>FIL3.DAT;1/AS/AP<RET>
```

In this example, FIL1.DAT;1, which is stored on DK1: at node LOWELL, is opened. FIL2.DAT;1, located at node ESTER, and FIL3.DAT;1, located on node YUKON, are then appended to FIL1.DAT;1 in ASCII record mode.

3.2.5 The Copy Operation (File Transfer)

MCR Primary Switch: (*default operation*)

The NFT copy operation can transfer files in the following ways:

- **Single file transfer**. Transfers a single file from one node to another.

- **File transfer with concatenation.** Transfers more than one input file, from the same or different nodes, to create a single output file.

  The attributes of the output file are determined by the attributes of the first input file that you specify in the command line. If the attributes of the input files differ, the concatenation may appear to succeed, but the output file may not be correctly readable.

- **Multiple file transfer.** Transfers more than one input file 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:

NFT>[*outfile*][/*sw*] = *infile*(*s*)[/*sw*]

where

outfile is an optional output file descriptor to which the input file(s) are copied.

infile(*s*) is one (or more) input file descriptor(s).

/sw is one (or more) of the following qualifying switches (defined in Section 3.2.3):

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

NOTE

Refer to Section 3.2.1, which contains the rules for specifying both file and command qualifying switches in a command line.

Example 1: Single file transfer

```
NFT>=YUKON::THURS.TXT<RET>
```

In this example, the file THURS.TXT, located on remote RSX node YUKON, is transferred 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 YUKON. Therefore, missing fields are assumed to be defined either in the respective default node names or in alias node names.

Example 2: Single file transfer between remote nodes

```
NFT>NODEX/[200,30]/DECNET::DK0:[BUCKLEY]MAGIC.MAC;3=-<RET>
NFT>NODEY/200,10/PASME::SY0:[30,60]MYSTIC.MAC;2<RET>
```

In this example, the file MYSTIC.MAC;2, located on NODEY, an RSX-11 node, on the system device under UFD [30,60] is transferred to NODEX, also an RSX-11 node. The file is renamed MAGIC.MAC;3, and is stored on DK0: in the UFD [BUCKLEY]. 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 transfer using a foreign file

```
NFT>NODEA::MARY.MAC=NODEB::SYS$SYSTEM:BOOTS.EYE<RET>
```

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. Therefore, it is possible that much of the file descriptor information discussed is not even required by NODEB. (You should refer to Section 3.1.4 to determine what foreign file specification format is required.) If, however, you omit any required argument in the command line and do not adhere to default regulations, you will receive an error message.

Example 4: Transfer with file concatenation

```
NFT>NODEA::DT1:[50,10]SUM.TIM=NODEY :DKO:[50,10]-<RET>
NFT>SAM.LIT,ONE.WAY<RET>
```

In this example, SAM.LIT and ONE.WAY located on RSX NODEY, are transferred to RSX NODEA. The two input files are concatenated into a single output file named SUM.TIM, which is stored on DECtape under UFD [50,10]. Notice how ONE.WAY inherits default values from the information specified for SAM.LIT. (ONE.WAY and SAM.LIT are located on NODEY on DK0: 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>NODEA::=NODEB/[FRASER]/MULTI::-<RET>
NFT>DXO:BEE.FIL/AS,CEE.FIL/AS,DEE.FIL/AS<RET>
```

In this example, BEE.FIL, CEE.FIL, and DEE.FIL are transferred to NODEA in ASCII record mode and stored under their original names (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>=BOSTON/310,2/MGR::*.*<RET>
```

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

NOTE

The use of brackets ([]) around the user ID is optional (as shown in the preceding example), if the user ID is supplied in the form of a UIC.

3.2.6 The Delete Operation

MCR Primary Switch: /DE

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

Format:

NFT>*filedescriptor(s)*[*/sw*]/DE

where

filedescriptor(s) is any valid file descriptor(s), as defined in Section 3.1. The version number must always be specified for files deleted on RSX nodes.

/sw is one (or both) of the following qualifying switches (defined in Section 3.2.3):

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

NOTE

Command qualifying switches do not affect input or output files and, therefore, can be specified anywhere in a command line.

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>MAIN::DK:[100,10]TAX.LST;5,BUDGET.LST;1/DE<RET>
```

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

3.2.7 The MCR Directory Operations

Directory Switches: /LI[:*width*]
/FU[:*width*]
/AT
/BR

MCR has four types of file directory listings. The /LI, /FU, /AT, and /BR switches are used to obtain each type of directory listing and are described here:

- **/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 note. 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:

NFT>[*outfile* =]*infile*(*s*) { /LI[:*width*] | /FU[:*width*] | /BR | /AT }

where

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

infile(*s*) is the file descriptor for each file for which you want directory listings. If no file name or type is specified in 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>X::DBO:[SMITH]/LI<RET>
```

The directory of DB0:[SMITH] on node X is displayed on TI: as shown here:

```
Directory X::DBO:[SMITH]
3-FEB-83 13:26:02

X.MAC;3          2./2.       29-JAN-83 17:46:23
X.OBJ;4          1./1.       29-JAN-83 17:46:34
X.TSK;2          4./4.    C  29-JAN-83 17:46:51
Y.TSK;3          0./0.    CL 29-JAN-83 17:47:08
```

where the ordering of the fields from left to right is 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>X::DBO:[7,7]/BR<RET>
```

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

```
Directory X::DBO:[7,7]
3-FEB-83 13:21:22

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

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

Example 3: The Full Directory Listing Switch (/FU)

```
NFT>X::DB0:[7,7]/FU<RET>
```

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

```
Directory X::DB0:[7,7]
3-FEB-83 13:27:39

X.OBJ;4          1./1.                 29-JAN-83 17:46:34
[7,7]            [RWED,RWED,RWED,R]    01-FEB-83 09:58:32(2.)
```

where the ordering of the fields from left to right by line is:

First Line	Second Line
file name	file owner (UIC)
blocks used/allocated	file protection
creation date and time	last modified date and time revision number

If the file is stored on contiguous blocks, an uppercase character C appears to the left of the creation date, as shown in the example below.

```
X.TSK;2          4./4.                 C  29-JAN-83 17:46:51
[7,7]            [RWED,RWED,RWED,R]       01-FEB-83 09:58:36(2.)
```

If the file is locked, an uppercase L appears to the left of the creation date (see previous example for the /LI switch). (Not all fields are displayed for all file systems; for example, RT-11 directories do not include the file owner and the modification date and time.)

Example 4: The Attributes Listing Switch (/AT)

```
NFT>ELROND::LOGIN.CMD/AT<RET>
```

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

```
Directory DBO:[DALEY]
15-JUN-83 12:00:06

LOGIN.CMD;22
                        Size:           2./2.    Created: 15-JUN-1983 10:40
                        Owner:      [DALEY]      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
```

3.2.8 The Execute Operation

MCR Primary Switch: /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. If you want to execute a command file that does not already exist on a remote node, you must use the submit operation (see Section 3.2.15) to copy the command file to the remote node, and subsequently submit the file for execution.

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

Format:

NFT>*commandfile(s)*[*/sw*]/EX

where

commandfile(s) is the output file descriptor(s) for each remote node command file that you want to execute. Command files are not deleted after execution.

/sw is one (or both) of the following qualifying switches (defined in Section 3.2.3):

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

NOTE

Command qualifying switches do not affect input or output files and, therefore, can be specified anywhere in a command line.

Example:

```
NFT>WASH::DB0:[200,200]NAMES.CMD/EX<RET>
```

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

3.2.9 The Help Operation

MCR Primary Switch: **/HE**
HELP[*subject*]

The NFT help operation displays information on how to use NFT. To print the list of subjects for which there is help information available, type either version of the switch:

Format:

NFT>/HE

NFT>HELP

To obtain information about one of the subjects listed, type

NFT>HELP[*subject*]

Example:

```
NFT>HELP TRANSFER<RET>
```

This help request displays text describing the syntax for transferring files.

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

```
NFT - No HELP Available
```

If your help files are not on the proper device, consult your system manager.

3.2.10 The Identify Operation

MCR Primary Switch: /ID

The identify operation displays the current NFT and NFARs version numbers on your terminal. The type of operating system and file support for your node, as well as the DAP buffer size, are also displayed. If a remote node is specified in the command, information is displayed for the remote FAL.

Format:

NFT>[*node specification*::]/ID

where

node specification is a remote node name and access control information (see Section 3.1 for information on specifying node names and access control information).

Example 1: Local Node

```
NFT>/ID<RET>

NFT - Version 4.1
Local NFARs V4  DAP V7.0  Buffer size=528.  OS=RSX-11M-PLUS  FS=FCS-11
```

Example 2: Remote RSX Node

```
NFT>ELROND::/ID<RET>

NFT - Version 4.1
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>BERGIL::/ID<RET>

NFT - Version 4.1
Local NFARs V4  DAP V7.0  Buffer size=528.  OS=RSX-11M-PLUS  FS=FCS-11
Remote FAL  V4  DAP V6.0  Buffer size=4160.  OS=VAX/VMS  FS=RMS-32
```

3.2.11 The Rename Operation

MCR Primary Switch: /RE

The rename operation changes the name(s) of the input file(s) to that of the output file specifier. The new file name is entered into the new directory and the existing or current file name(s) are deleted from the old directory. Because no data is transferred on a rename operation, both directories must be on the same node and same device. Since the operation is performed on the source node, access control information need be given for the input file(s) only. If the rename is to a different directory, the target directory must also allow access. (Some systems do not support the DAP rename operation. Refer to the *Introduction to DECnet* or to the individual system documentation for more information.)

Format:

NFT>*outfile*[*/sw*] = *infile*(*s*)[*/sw*]/RE

where

outfile is the file specification to be given to the new file. (See Section 3.1.2 for a description of the 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. Thus, one or more fields can be changed while preserving the others. (See Section 3.1.3 for more information on wildcard specifications.)

infile(s) is the file descriptor of the file(s) to be renamed.

/sw is one (or more) of the following qualifying switches (defined in Section 3.2.3):

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

NOTE

Refer to Section 3.2.1, which contains the rules for specifying both file and command qualifying switches in a command line.

Example 1:

```
NFT>REM::DB3:[100,1]NEWNAME.FIL=REM/ACCOUNT/PASS::-<RET>
NFT>DB3:[100,1]OLDNAME.FIL/RE<RET>
```

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

Example 2:

```
NFT>BILBO::DL1:[FRASER]SONGS.OLD=BILBO::-<RET>
NFT>DL1:[BUCKLEY]SONGS.NEW/RE<RET>
```

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

3.2.12 The Set Default Operation

MCR Primary Switch: /DF

The set default operation allows you to define your own 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, the file name, type, and version number cannot be assigned default values. Default values can be set for input files by specifying IN, and for output files by specifying OUT. You can define the same default values for both input and output files by using one command and omitting the IN/OUT arguments. Defaults remain in effect for subsequent files until they are changed or until you exit from NFT.

Format:

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

where

defaults is any string of default argument values that you wish to define in the command line. If omitted, NFT displays the last defaults set.

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

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

Any file descriptor arguments that are not set with the /DF switch are defined by the standard NFT defaults listed in Table 3-1. If you want to know which defaults are currently defined, type the following command:

```
NFT>/DF<RET>
```

NFT will display the current default values at your terminal.

Example:

```
NFT>MASTER/GUMBO/BLUE::[4,5]/DF:IN<RET>
NFT>TI:/DF:OUT<RET>
NFT>=TIN.MAN<RET>
```

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

NFT>TI: = MASTER/GUMBO/BLUE::[4,5]TIN.MAN

If you wished to know the defaults set in this example, /DF would print the following itemization in response to NFT>/DF<RET>:

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

Output defaults =TI:
```

Notice that the default password is printed out as ellipses (...) to maintain security.

3.2.13 The Set Protection Operation

MCR Primary Switch: **/PR[:***n***]**

[/SY:*p***]**

[/OW:*p***]**

[/GR:*p***]**

[/WO:*p***]**

The set protection operation sets or changes a file's protection status. File protection is provided at four levels (/SY:*p*, /OW:*p*, /GR:*p*, /WO:*p*):

- **/SY:*p* (System)** specifies file access that you allow to the system UICs (that is, UICs with group numbers of 10 octal or less).
- **/OW:*p* (Owner)** specifies file access that you allow to yourself.
- **/GR:*p* (Group)** specifies file access that you allow to other members of your group.
- **/WO:*p* (World)** specifies file access that you allow for all other UICs.

The *p* variable signifies the protection settings R, W, E, and/or D. The R, W, E, and D protection settings specify whether each level can read, write, extend, or delete a file, respectively. The R, W, E, and/or D protection settings apply to all input files in a command line, unless overridden.

To alter the protection level of a file, you can use either the /PR qualifying switches (/SY:*p*, /OW:*p*, /GR:*p*, /WO:*p*) or an octal representation (/PR:*n*). In either case, if you are the file owner or have a system level UIC, you can alter the protection level without having read or write access. However, because the protection level of a file is a protected attribute, you cannot alter the protection level 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.)

Format:

NFT>*filedescriptor(s)*/PR[:*n*][/SY:*p*][/OW:*p*][/GR:*p*][/WO:*p*][/FO:*uic*][/*sw*]

where

filedescriptor(s) is the file descriptor(s) for each file in the command line.

/PR is the protect switch.

:*n* is an optional octal value that can be used to 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 here and in Section 3.2.3).

/OW: is the owner level protection qualifying switch (defined here and in Section 3.2.3).

/GR: is the group level protection qualifying switch (defined here and in Section 3.2.3).

/WO: is the world level protection qualifying switch (defined here and in Section 3.2.3).

p is R, W, E, and/or D, signifying the type of access that is to be allowed. R, W, E, and D signify the following types of access:

R - read allowed E - extend allowed

W - write allowed D - delete allowed

Only those types of access that you specify (R, W, E, and/or D) will be allowed; all others remain protected. If you specify one of the four qualifying switches, (/SY:*p*, /OW:*p*, /GR:*p*, /WO:*p*), but do not specify a value for *p*, no access is allowed at the level of that qualifying switch.

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

/sw is one (or both) of the following qualifying switches (defined in Section 3.2.3):

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

NOTE

Command qualifying switches do not affect input or output files and, therefore, can be specified anywhere in a command line.

Example 1: Changing protection status on a transfer operation

```
NFT>HALDIR::ALIAS.CMD=ALIAS.CMD/PR/SY/WO/GR/OW:RWD<RET>
```

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

Example 2: Changing protection status on an existing file

```
NFT>HALDIR::ALIAS.CMD/PR/OW:R<RET>
```

File ALIAS.CMD allows the owner to read only.

3.2.14 The Spool Operation

MCR Primary Switch: /SP

The spool operation directs 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. When format conversion is necessary, use the /AS and /RC switches.

Format:

NFT>[*outfile*[*/sw*] =]*infile*(*s*)[*/sw*]/SP

where

outfile is an optional file descriptor that specifies the output file to which the file will be copied and the system on which it will be spooled. If *outfile* is omitted, the file(s) specified by *infile*(*s*) is spooled on the input node.

infile(*s*) is the file descriptor for each file to be spooled for printing.

/sw is one (or more) of the following qualifying switches (defined in Section 3.2.3):

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

NOTE

Refer to Section 3.2.1, which contains the rules for specifying both file and command qualifying switches in a command line.

Example 1:

```
NFT>SITKA::COLD.FIL;1,HOME::SNOW.FIL;1/SP<RET>
```

In this example, one copy of the file COLD.FIL;1 is spooled to the line printer on node SITKA, and one copy of the file SNOW.FIL;1 is spooled to the line printer on node HOME.

Example 2:

```
NFT>HOME::SITKA::COLD.FIL;1,HOME::SNOW.FIL;1/SP<RET>
```

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 input file names default to the output file name.

3.2.15 The Submit Operation

MCR Primary Switch: /SB

The submit operation transfers a command file to a remote node and submits the file to either a remote command file processor or a batch file processor for execution. The command file is deleted (by the remote node) after the file executes. The remote node must support command file submission and execution (for example, RT-11 nodes do not support command file submission). When you perform a submit operation, remember that the syntax of the command lines within the command file must conform to the syntax standards of the remote system on which the file is to be executed. To execute a command file that is already on the remote node, you can use the execute (/EX) operation (see Section 3.2.8).

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.

The success of the submit operation does not guarantee that the batch or command file was executed successfully, only that the file was transferred and that the execution request was successfully given to the MCM.

Format:

NFT>*outfile*[*/sw*] = *commandfile*[*/sw*]/SB

where

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

commandfile is the input file descriptor of the file containing the commands to be executed by the destination node's operating system. Only one command file can be specified.

/sw is one (or more) of the following qualifying switches (defined in Section 3.2.3):

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

NOTE

Refer to Section 3.2.1, which contains the rules for specifying both file and command qualifying switches in a command line.

Example:

```
NFT>DENVER::DBO:[100,100]SVEFIL.CMD=INST.CMD/SB<RET>
```

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.

3.3 NFT Operations for DCL Users

DCL NFT operations act on DCL command verbs which specify the operation to be performed (for example, the APPEND command specifies the append operation). Most DCL NFT operations have optional qualifiers that can further define the action of the command verb (for example, APPEND/DATA_TYPE=ASCII causes an input file to be appended to an output file in ASCII record mode). The DCL NFT operations are summarized in Table 3-3. Sections 3.3.4 - 3.3.14 describe each DCL NFT operation in detail.

3.3.1 Rules for Specifying DCL Qualifiers in a Command Line

An NFT DCL command line consists of a command verb, one or more input and/or output file specifications, and optional qualifiers (see Section 3.3). The optional qualifiers fall into two categories:

1. **Command Qualifiers.** A command qualifier does not affect input or output files and, therefore, can be specified anywhere in a command line.

2. **File Qualifiers.** A file qualifier applies only to each specific input file to which it is appended.

   A file qualifier achieves global status, and becomes the default for the rest of the command line when:

- a file qualifier is appended to an output file specification.

- a file qualifier is appended to a command verb.

A global qualifier can be overridden on a per file basis.

3.3.2 DCL Command Line Continuer

You can continue a DCL command line onto an additional input line by using a hyphen (-) before the line terminator (see the example below). The hyphen, followed by the terminator (<RET>), acts as a command line continuer. A command line continuer causes DCL to prompt you for the remainder of the command with an underscore followed by an angle bracket (_>). You can insert a continuer at any point in a command line, and you can use any number of continuing lines within a command.

Example:

```
$COPY NODEY/200,10/PASME::SYO:[BUCKLEY]MYSTIC.MAC;2-<RET>
_> NODEX/[200,30]/DECNET::DTO:[DALEY]MAGIC.MAC;3<RET>
```

Table 3–3: Summary of NFT Operations for DCL Users

NFT Operation	DCL Command	DCL Qualifiers	(Scope of Qualifier)
Append: The append operation adds 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: The copy operation transfers one or more existing files to one or more files on the local or remote node.	**COPY**	/TRANSFER_MODE = 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: The create operation 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: The delete operation deletes one or more files.	**DELETE**	/LOG	(cmd)
		/NOWARNINGS	(cmd)

(continued on next page)

Table 3–3 (cont.): Summary of NFT Operations for DCL Users

NFT Operation	DCL Command	DCL Qualifiers	(Scope of Qualifier)
Directory: The directory operation lists a directory of one or more files.	**DIRECTORY**	/BRIEF	(cmd)
		/FULL	(cmd)
		/ATTRIBUTES	(cmd)
		/WIDTH = *n*	(cmd)
		/OUTPUT = *outfile*	(cmd)
		/NOWARNINGS	(cmd)
Print: The print operation 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: The rename operation changes the name of an existing file.	**RENAME**	/NEW_VERSION	(file)
		/LOG	(cmd)
		/NOWARNINGS	(cmd)
Set protection: The protect operation sets or changes a file's protection status.	**SET PROTECTION**	/OWNER = [*uic*]	(file)
		/LOG	(cmd)
		/NOWARNINGS	(cmd)
Set file: The set file operation 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)
Submit: The submit operation 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: The type operation types the contents of a file to your terminal screen.	**TYPE**	/MACY11	(file)
		/LOG	(cmd)
		/NOWARNINGS	(cmd)

3.3.3 Definition of DCL Qualifiers

NOTE

There are abbreviated synonyms for the data type and transfer mode qualifiers which you can substitute in order to simplify a command line. The qualifiers and their respective synonyms are listed below:

- /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, the file is created as a variable length format, implied LF-CR attribute file.

The /DATA_TYPE = ASCII qualifier should be used only if translation is desired. File transfers to remote nodes that have similar file systems (for example, VMS) are more efficiently handled in image block mode (the default transfer mode).

/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 causes NFT to use its default algorithms in selecting the mode in which to transfer a file to a remote node. 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 of transfer, you will want to use the /TRANSFER_MODE=AUTO qualifier to override a specified mode of transfer.

/TRANSFER_MODE=BLOCK (/BLOCK)

The block mode qualifier transfers files using 512.-byte blocks, which is a more efficient method than the usual record mode transfer. Block mode can be used with any file organization type on transfers to other Files-11 or RMS-11 systems that support block mode transfers (for example, RSX, VMS, or the RSTS/E RMS FAL). When transferring files in block mode either to or from a foreign file system, there is no guarantee that the files will be usable, since all foreign systems do not store records compatibly.

/TRANSFER_MODE = RECORD (/RECORD)

The record mode transfer qualifier directs NFT to transfer a file one record at a time, instead of using block mode. You would use this qualifier when copying to unit record equipment or to override NFT's default selection of the transfer mode.

/CONTIGUOUS

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

/MACY11

The MACY11 qualifier causes a file to be stored in MACY11 format on a TOPS-10 or a TOPS-20 system.

/NEW_VERSION

The new version qualifier specifies that an output file will have the same or latest version number, whichever is higher. However, the effect of the /NEW_VERSION qualifier depends on the version number support on the remote system. For example, RT-11 and RSTS/E do not support version numbers; therefore, this qualifier would be ignored on those systems.

There is no negative version of the /NEW_VERSION qualifier. However, you can use the REPLACE command to specify that an output file will retain the current version number. If another file with the same version number already exists, an error will result.

/NO_SPAN

The no span blocks qualifier specifies that an ouput file will not span block boundaries.

/PROTECTION = *level*

The protect qualifier allows a file's protection status to be set or changed. File protection is provided at four levels (SYSTEM: RWED, OWNER:RWED, GROUP:RWED, WORLD:RWED):

- **SYSTEM:** specifies file access that you allow to the system UICs (that is, UICs with group numbers of 10 octal or less).
- **OWNER:** specifies file access that you allow to yourself.
- **GROUP:** specifies file access that you allow to other members of your group.
- **WORLD:** specifies file access that you allow for all other UICs.

RWED specifies whether each level can read, write, extend, or delete a file, respectively. (See Section 3.3.11 for a full description of the set protection operation.)

/OWNER = **[***uic***]**

The owner qualifier allows you to specify the file owner for an output file.

/SUBMIT

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

/PRINT

The print qualifier directs a file to a line printer for printing.

/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 retains the current file name, type, and version number that are in the directory, but the file ID is new.

If an error occurs during the file transfer, both the new output file and the one 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 names of files (copied, deleted, and so forth) as the operation is performed. The log qualifier applies to all files involved in the current command.

/NOWARNINGS

The no warnings qualifier suppresses output messages that would otherwise be displayed at your terminal during NFT operations.

3.3.4 The Append Operation

DCL Command: APPEND

The append operation adds the contents of one or more input files to the end of an existing output file. The output file retains its original attributes independent of the attributes of the input file(s). However, if the attributes of the input files are different from the attributes of the output file, the append operation will appear to succeed, but the resulting output file may not read correctly.

Format:

$APPEND[/*qualifier*(*s*)] *infile*(*s*)[/*qualifier*(*s*)] *outfile*[/*qualifier*(*s*)]

where

outfile is the output file descriptor. No wildcard specifiers are allowed in the output file descriptor. The file type and record attributes are taken from the existing file. The file name and file type for the output file must be explicitly specified.

infile(*s*) is one (or more) input file descriptor(s).

/*qualifier*(*s*) is one (or more) of the following qualifiers (defined in Section 3.3.3):

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	

NOTE

Refer to Section 3.3.1, which contains the rules for specifying both file and command qualifiers in a command line.

Example 1:

```
$APPEND/ASCII/LOG ESTER::FIL2.DAT;1,YUKON::FIL3.DAT;1-<RET>
_> LOWELL::DK1:FIL1.DAT;1<RET>
```

In this example, the /ASCII qualifier has been appended to the command verb. Therefore, both input files in the command line (FIL2.DAT;1 located on node ESTER, and FIL3.DAT;1 located on node YUKON) are appended to FIL1.DAT;1 stored on DK1: on node LOWELL, in ASCII record mode. The command qualifier, /LOG, causes the names of the two files being appended to FIL1.DAT to be displayed at your terminal as each append operation takes place.

Example 2:

```
$APPEND ESTER::FIL2.DAT;1,YUKON::FIL3.DAT-<RET>
_> LOWELL::DK1:FIL1.DAT/ASCII/LOG<RET>
```

In this example, the /ASCII qualifier has been appended to the output file. Therefore, both input files in the command line (FIL2.DAT;1 located on node ESTER, and FIL3.DAT;1 located on node YUKON) are appended to FIL1.DAT;1 stored on DK1: on node LOWELL, in ASCII record mode. The command qualifier, /LOG, causes the names of the two files being appended to FIL1.DAT to be displayed at your terminal as each append operation takes place.

Example 3:

```
$APPEND ESTER::FIL2.DAT;1/ASCII/LOG,YUKON::FIL3.DAT;1-<RET>
_> LOWELL::DK1:FIL1.DAT;1<RET>
```

In this example, the /ASCII qualifier has been appended to the FIL2.DAT;1 input file. Therefore, although both input files in the command line (FIL2.DAT;1 located on node ESTER, and FIL3.DAT;1 located on node YUKON) are appended to FIL1.DAT;1 stored on DK1: on node LOWELL, only FIL2.DAT;1 is transferred in ASCII record mode. The command qualifier, /LOG, causes the names of the two files being appended to FIL1.DAT to be displayed at your terminal as each append operation takes place.

3.3.5 The Copy Operation (File Transfer)

DCL Command: COPY

The NFT copy operation can transfer files in the following ways:

- **Single file transfer.** Transfers a single file from one node to another.

- **File transfer with concatenation.** Transfers more than one input file, from the same or different nodes, to create a single output file.

  The attributes of the output file are determined by the attributes of the first input file that you specify in the command line. If the attributes of the input files differ, the concatenation may appear to succeed, but the output file may not read correctly.

- **Multiple file transfer.** Transfers more than one input file 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:

$COPY[/*qualifier(s)*] *infile(s)*[/*qualifier(s)*] *outfile*[/*qualifier(s)*]

where

infile(s) is one (or more) input file descriptor(s).

/qualifier(s) is one (or more) of the following qualifiers (defined in Section 3.3.3):

File Qualifiers	Command Qualifiers
/DATA_TYPE = ASCII	/CONCATENATE
/DATA_TYPE = IMAGE	/LOG
/TRANSFER_MODE = AUTO	/NOWARNINGS
/TRANSFER_MODE = BLOCK	
/TRANSFER_MODE = RECORD	
/CONTIGUOUS	
/MACY11	
/NEW_VERSION	
/NO_SPAN	
/PROTECTION = *level*	
/OWNER = [*uic*]	
/PRINT	
/REPLACE	
/SUBMIT	

NOTE

Refer to Section 3.3.1, which contains the rules for specifying both file and command qualifiers in a command line.

Example 1: Single file transer

```
$COPY YUKON::THURS.TXT THUR.TXT<RET>
```

In this example, THURS.TXT located on remote node YUKON, is transferred to the local node and renamed THUR.TXT.

Example 2: Single file transfer between remote nodes

```
$COPY/NEW_VERSION YUKON::THURS.TXT RASTOR::DR5:[MITTON]<RET>
```

In this example, THURS.TXT located on node YUKON, is transferred to node RASTOR, to be stored on DR5: under UFD [MITTON]. Once THURS.TXT is transferred to RASTOR, the file will have the same or latest version number, whichever is higher.

Example 3: Single file transfer using a foreign file

```
$COPY YUKON::THURS.TXT/MACY11 THUR TXT<RET>
```

In this example, THURS.TXT located on remote node YUKON, is transferred in MACY11 format to the local TOPS-20 node and renamed THUR.TXT.

Example 4: Transfer with file concatenation

```
$COPY/CONCATENATE YUKON::THURS TXT,FRI.TXT WEEK.TXT<RET>
```

In this example, THURS.TXT and FRI.TXT located on remote node YUKON, are transferred to the local node and concatenated to a single output file named WEEK.TXT.

Example 5: Multiple file transfer

```
$COPY YUKON::THURS.TXT/REPLACE,RASTOR::FRI.TXT-<RET>
_> ALEN::DRA1:[DATES]/NEW_VERSION<RET>
```

In this example, THURS.TXT located on node YUKON, and FRI.TXT located on node RASTOR, are transferred to node ALEN, and stored on DRA1: in the UFD [DATES]. Both files will have the same or latest version number, whichever is higher.

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

Example 6: Multiple file transfer using wildcards

```
$COPY/NEW_VERSION RASTOR::DR5:[MITTON]*.OBJ,FILES.*;*-<RET>
_> KREMIT::DB3:[ALBANESE]<RET>
```

In this example, the latest version of all files having the file type, OBJ and, all versions of all files having the filename, FILES that are stored on DR5:[MITTON] on node RASTOR, are transferred to node KREMIT, and stored on DB3:[ALBANESE]. The resulting output files will have the same names as the input files. The resulting output files will have the same or latest version number, whichever is higher.

3.3.6 The Create Operation

DCL Command: CREATE

The create operation creates an output file that previously did not exist. The output file is created (using ASCII record mode), from data that you enter at your terminal. The data that you enter is copied into the file until you type a <CTRL/Z> (end-of-file indicator).

Format:

$CREATE *outfile*[*/qualifier*(*s*)]

where

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

/qualifier(*s*) is one (or more) of the following qualifiers (defined in Section 3.3.3):

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

NOTE

Refer to Section 3.3.1, which contains the rules for specifying both file and command qualifiers in a command line.

Example:

```
$CREATE RASTOR::DR5:[7,132]MYFILE.TXT<RET>
```

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

3.3.7 The Delete Operation

DCL Command: DELETE

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

Format:

$DELETE[/*qualifier(s)*] *filedescriptor(s)*[/*qualifier(s)*]

where

filedescriptor(s) is any valid file descriptor(s), as defined in Section 3.1. The version number must always be specified for files deleted on RSX nodes.

/*qualifier(s)* is one (or both) of the following qualifiers (defined in Section 3.3.3):

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

NOTE

Command qualifiers do not affect input or output files, and therefore, can be specified anywhere in a command line.

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:

```
$DELETE MAIN::DK:[100,10]TAX.LST;5,BUDGET.LST;1<RET>
```

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

3.3.8 The DCL Directory Operations

DCL Command: DIRECTORY

DCL Directory Qualifiers

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

DCL has four types of file directory listings. The DIRECTORY command (specified alone, or in conjunction with the /FULL, /ATTRIBUTES, or /BRIEF qualifiers) is used to obtain each type of directory listing as described here:

- The **DIRECTORY** command (when specified without any of the optional qualifiers), lists the name, size, creation date, and optional contiguous and locked indicators for the files specified in the input file descriptor (which 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 file organization, record format, and record attributes, as well as the information included in a full (/FU) directory listing.

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

Format:

```
$DIRECTORY  [/FULL          ]        infile(s)
            [/ATTRIBUTES    ]
            [/BRIEF         ]
            [/WIDTH = n     ]
            /OUTPUT = outfile
```

where

/OUTPUT = *outfile* is an optional file descriptor (that consists of a *node specification*:: and associated *file specification*, see Section 3.1), to which the directory listing will be written. If omitted, the output defaults to TI:.

/WIDTH = *n* 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. The /WIDTH = *n* qualifier cannot be used in conjunction with either the /ATTRIBUTES or the /BRIEF qualifiers.

infile(*s*) is the input file descriptor (that consists of a remote *node specification*:: and associated *file specification*, see Section 3.1), for each remote file for which you want directory listings. If no file name or type is specified in the descriptor, you will receive a directory listing of all files in the specified directory.

Example 1: A Sample DIRECTORY Listing

```
$DIRECTORY WASH::DBO:[BUCKLEY]<RET>
```

The directory of DB0:[BUCKLEY] on node WASH is displayed on TI: as shown here:

```
Directory DBO:[BUCKLEY]
7-AUG-84 22:06

REMINDER.CMD;1                3.  07-AUG-84 22:02
MASTERINI.CMD;1               11. 07-AUG-84 22:02
TELI.CMD;1                    2.  07-AUG-84 22:02
TESTFILE.COM;1                0.  07-AUG-84 22:02
LOGIN.CMD;1                   1.  07-AUG-84 22:02
PEOPLE.LST;1                  1.  07-AUG-84 22:02
NULLTEST.LST;1                2.  07-AUG-84 22:02
COPY.LST;1                    5.  07-AUG-84 22:02
NFTTEST.LOG;1                 13. 07-AUG-84 22:04
TSETCMP.LOG;1                 10. 07-AUG-84 22:04
SETCMP.LOG;1                  9.  07-AUG-84 22:04
```

Example 2: A Sample DIRECTORY/FULL Listing

```
$DIRECTORY/FULL WASH::DBO:[200,200]*.LST<RET>
```

The directory of all files having the file type .LST that are located on DB0:[200,200] on node WASH, are displayed on TI: as shown here:

```
Directory DRO:[200,200] 7-AUG-84 22:07

PEOPLE.LST;1    (40345,1)  1./1.  07-AUG-84 22:02 [7,132]  [RWED,RWED,RWED,R]
NULLTEST.LST;1 (40356,1)  2./2.  07-AUG-84 22:02 [7,132]  [RWED,RWED,RWED,R]
COPY.LST;1      (40375,1)  5./5.  07-AUG-84 22:02 [7,132]  [RWED,RWED,RWED,R]
```

Example 3: A Sample DIRECTORY/ATTRIBUTES Listing

```
$DIRECTORY/ATTRIBUTES/OUTPUT=DIR.LST-<RET>
_> WASH::DBO:[BUCKLEY]LOGIN.CMD,TESTFILE.CMD<RET>
```

The attributes of the files LOGIN.CMD and TESTFILE.CMD located on DB0:[BUCKLEY] on node WASH, are written to the file DIR.LST.

```
DBO:[BUCKLEY]LOGIN.CMD;1
                        Size:      1./1.          Created: 7-AUG-1984 22:02
                        Owner:     [BUCKLEY]      Revised: 7-AUG-1984 22:02
                        File ID:   (40320,1,0)    Expires: <nonespecified>
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

DBO:[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: <nonespecified>
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 Sample DIRECTORY/BRIEF Listing

```
$DIRECTORY/BRIEF BOSTON::DR5:[305,344]*.LOG<RET>
```

The directory of all files having the file type .LOG that are located on DR5:[305,344] on node BOSTON, are displayed on TI: as shown here:

```
Directory DR5:[305,344]

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

3.3.9 The Print Operation

DCL Command: PRINT/REMOTE

The print operation prints a file which exists on a remote node on that remote node's line printer. When you specify the PRINT/REMOTE command, the input file(s) are printed on the remote node that you specify in the command line. (To copy a file to a remote node and have it printed, you would have to specify the COPY/PRINT command. See Section 3.3.5 for a description of the copy operation, listing /PRINT as a valid qualifier.)

Format:

$PRINT/REMOTE [*/qualifier(s)*] *infile(s)*[*/qualifier(s)*]

where

/REMOTE is the qualifier used to print a file on a remote node.

infile(s) is the file descriptor for each file to be printed.

/qualifier(s) is one (or more) of the following qualifiers (defined in Section 3.3.3):

File Qualifiers	Command Qualifier
/DATA_TYPE = ASCII	/LOG
/TRANSFER_MODE = AUTO	
/TRANSFER_MODE = BLOCK	
/DATA_TYPE = IMAGE	
/NEW_VERSION	
/TRANSFER_MODE = RECORD	
/REPLACE	

NOTE

Refer to Section 3.3.1, which contains the rules for specifying both file and command qualifiers in a command line.

Example:

```
$PRINT/REMOTE HOME::SNOW.FIL;1,COLD.FIL;1<RET>
```

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.

3.3.10 The Rename Operation

DCL Command: RENAME

The rename operation changes the name(s) of the input file(s) to that of the output file specifier. The new file name is entered into the new directory and the existing or current file name(s) are deleted from the old directory. Because no data is transferred on a rename operation, both directories must be on the same node and same device. Since the operation is performed on the source node, access control information need be given for the input file(s) only. If the rename is to a different directory, the target directory must also allow access. (Some systems do not support the Data Access Protocol (DAP) rename operation. Refer to the *Introduction to DECnet* or to the individual system documentation for more information.)

Format:

$RENAME[/*qualifier*(*s*)] *infile*(*s*)[/*qualifier*(*s*)] *outfile*[/*qualifier*(*s*)]

where

outfile is the file specification to be given to the new file. (See Section 3.1.2 for a description of the 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. Thus, one or more fields can be changed while preserving the others. (See Section 3.1.3 for more information on wildcard specifications.)

infile(*s*) is the file descriptor of the file(s) to be renamed.

/*qualifier*(*s*) is one (or more) of the following qualifiers (defined in Section 3.3.3):

File Qualifier	Command Qualifiers
/NEW_VERSION	/LOG /NOWARNINGS

NOTE

Refer to Section 3.3.1, which contains the rules for specifying both file and command qualifiers in a command line.

Example:

```
$RENAME WASH::DB0:[200,200]*.MAR,*.MAL *.MAC<RET>
```

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).

3.3.11 The Set Protection Operation

DCL Command:

SET PROTECTION =
[(SYSTEM:RWED,OWNER:RWED,GROUP:RWED,WORLD:RWED)]
[/OWNER = [*uic*]]

The set protection operation sets or changes a file's protection status. File protection is provided at four levels (SYSTEM:RWED, OWNER:RWED, GROUP:RWED, WORLD:RWED):

- **SYSTEM:** specifies file access that you allow to the system UICs (that is, UICs with group numbers of 10 octal or less).
- **OWNER:** specifies file access that you allow to yourself.
- **GROUP:** specifies file access that you allow to other members of your group.
- **WORLD:** specifies file access that you allow for all other UICs.

The R, W, E, and D protection settings specify whether each level can read, write, extend, or delete a file, respectively. The R, W, E, and/or D protection settings apply to all input files in a command line, unless overridden.

To alter the protection level of a file, you specify the SET PROTECTION = command along with any/all of the four protection level qualifiers that you choose (SYSTEM:RWED, OWNER:RWED, GROUP:RWED, WORLD:RWED).

If you are the file owner or have a system level UIC, you can alter the protection level of a file without having read or write access. However, because the protection level of a file is a protected attribute, you cannot alter the protection level 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.)

Format:

$SET PROTECTION = -<RET>
_>[(SYSTEM:RWED,OWNER:RWED,GROUP:RWED,WORLD:RWED)]-<RET>
_>*filedescriptor(s)*[/*qualifier(s)*][/OWNER = [*uic*]]

where

filedescriptor(s)	is the file descriptor(s) for each file in the command line.
SYSTEM:RWED	is the system level qualifier (defined here and in Section 3.3.3).
OWNER:RWED	is the owner level qualifier (defined here and in Section 3.3.3).
GROUP:RWED	is the group level qualifier (defined here and in Section 3.3.3).
WORLD:RWED	is the world level qualifier (defined here and in Section 3.3.3).
RWED	are the protection settings allowing read, write, extend, and/or delete privileges, respectively.
/OWNER = [*uic*]	is the file owner qualifier which allows you to set the owner of a file to a specified UIC, in the form: [*g,m*].
/*qualifier(s)*	is one (or both) of the following qualifiers (defined in Section 3.3.3):

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

Example:

```
$SET PROTECTION=(SYSTEM:RWED,WORLD:RWED,GROUP:RWED,OWNER:RWD)-<RET>
_> HALDIR::ALIAS.CMD<RET>
```

File ALIAS.CMD allows complete file access on the system, world, and group levels, and allows the owner only to read, write, and delete the file.

3.3.12 The Set File Operation

DCL Command: SET FILE

The set file operation allows you to set the owner and protection for a file. The set file operation is an alternative method of specifying owner and protection values for a file. (Owner and protection values can also be specified with the set protection operation; see Section 3.3.11.)

Format:

$SET FILE[/*qualifier(s)*] *infile(s)*[/*qualifier(s)*]-<RET>
_>[/PROTECTION = *level*] [/OWNER = [*uic*]

where

infile(s)	is one (or more) input file(s) for which protection and/or owner values are specified.
/PROTECTION = *level*	is the protect qualifier which allows a file's protection status to be set or changed. File protection is provided at four levels; SYSTEM: RWED, OWNER:RWED, GROUP:RWED, WORLD:RWED, where RWED specifies read, write, extend, and delete privileges, respectively. (See the set protection operation, Section 3.3.11, for more information.)
/OWNER = [*uic*]	is the file owner qualifier which allows you to set the owner of a file to a specified UIC, in the form: [*g,m*].
/*qualifier(s)*	is one (or both) of the following qualifiers (defined in Section 3.3.3):

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

NOTE

Command qualifiers do not affect input or output files and, therefore, can be specified anywhere in a command line.

Example:

```
$SET FILE RASTOR::DR5:[MITTON]MY.FIL/PROTECTION=(GR:RW),-<RET>
_>HIS.FIL/OWNER=[MITTON]<RET>
```

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

3.3.13 The Submit Operation

DCL Command: SUBMIT/REMOTE

The submit operation executes the contents of a command file located on a remote node. You must specify the SUBMIT command along with the /REMOTE qualifier for all network file specifications. If you specify only the SUBMIT command and omit the /REMOTE qualifier, you will receive the following error message:

Must supply REMOTE qualifier with DECnet node specification

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.

If you want to execute a command file that does not already exist on a remote node, you must use the COPY/SUBMIT command to copy the command file to the remote node, and subsequently submit the file for execution. See Section 3.3.5 for a description of the copy operation listing /SUBMIT as a valid qualifier.

The success of the submit operation does not guarantee that the batch or command file was executed successfully, only that the execution request was given successfully to the MCM.

Format:

$SUBMIT/REMOTE [*/qualifier(s)*] *commandfile(s)*

where

/REMOTE	is the qualifier which executes a command file on a remote node.
commandfile(s)	is the output file descriptor(s) for each remote node command file that you want to execute. Command files are not deleted after execution.
/qualifier(s)	is one (or both) of the following qualifiers (defined in Section 3.3.3):

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

NOTE

Command qualifiers do not affect input or output files and, therefore, can be specified anywhere in a command line.

Example:

```
$SUBMIT/REMOTE/NOWARNINGS WASH::DBO:[200,200]NAMES.CMD<RET>
```

In this example, the command file NAMES.CMD located on node WASH, is executed (while suppressing the display of any output messages to your terminal).

3.3.14 The Type Operation

DCL Command: TYPE

The type operation types the contents of a file to your terminal screen.

Format:

$TYPE[/*qualifier(s)*] *infile(s)*[/*qualifier(s)*]

where

infile(s)	is one (or more) input file descriptor(s) for file(s) that will be displayed on your terminal screen.
/*qualifier(s)*	is one (or more) of the following qualifiers (defined in Section 3.3.3):

File Qualifier	Command Qualifiers
/MACY11	/LOG /NOWARNINGS

NOTE

Refer to Section 3.3.1, which contains the rules for specifying both file and command qualifiers in a command line.

Example:

```
$TYPE WASH::DBO:[ALBANESE]FIL2.DAT<RET>
```

This command causes the contents of FIL2.DAT located on node WASH, to be displayed on your terminal screen.

3.4 Exit with Status

NFT exits with a status code that may be useful in the following circumstances:

- In an indirect command file
- In a batch stream job (for RSX-11M-PLUS systems only)
- When spawned by a parent task or 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.

3.5 NFT Error Messages

When an NFT error occurs, you should receive one or more of the error messages listed in the following sections. (For DCL users, NFT parsing and syntax errors will generate appropriate DCL error messages.) NFT error messages fall into three categories:

- **Command error messages** pertain to the NFT interface (network or operating system interface), including I/O errors to the command device and errors caused by trying to run NFT on a system without network support enabled (see Section 3.5.1).

- **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 prints an error message specifying the name of the file and the operation that NFT was performing when the error occurred (see Section 3.5.2). A secondary message generally follows, giving the cause of the error.

  When a command line specifies multiple operations, NFT tries to execute subsequent operations, even after the error has been detected. In cases involving file transfers (except when appending files), NFT deletes the output file being created if an error occurs.

- **Secondary error messages** print out following primary error messages to provide specific information on the nature of the error. Error types fall into two categories: network errors and file I/O errors, (see Section 3.5.3). The message may be text, or it may be a DAP, FCS, or NSP error code. For example, the following error messages were returned in response to a command line in which an incorrect node name was specified in the input file specification:

```
NFT - Error in opening input file YOKON::FIL.CMD
      Unrecognized node name
```

The second line is the secondary error message, which helps isolate the cause of the error.

The NFT error messages are listed alphabetically in the following sections, which correspond to the three error categories described here.

3.5.1 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 in 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 has occurred while attempting to use the help operation (see Section 3.2.9). 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 specified command file cannot be found.

NFT - Syntax error

Either NFT cannot determine the operation being requested, or the command line is ambiguous. Reenter the faulty command line echoed after the error message, using the proper syntax.

NFT - Syntax error in file name *filedescriptor*

The specified file name did not conform to RSX syntax for file specifiers.

NFT - Unable to access network

NFT was invoked, but did not find network support.

3.5.2 Primary Error Messages

Whenever one of the primary error messages listed below is issued, it is followed by a secondary error message, which further defines the nature of the error (see Section 3.5.3).

NFT - Error in changing file protection *filedescriptor*

NFT - Error in closing input file *filedescriptor*

NFT - Error in closing output file *filedescriptor*

NFT - Error in deleting file *filedescriptor*

NFT - Error in getting record from file *filedescriptor*

NFT - Error in opening input file *filedescriptor*

NFT - Error in opening output file *filedescriptor*

NFT - Error in purging output file *filedescriptor*

When an error occurs in an NFT file operation, NFT attempts to delete the output file being created. If that attempt fails, NFT displays the preceding message.

NFT - Error in putting record in file *filedescriptor*

NFT - Error in reading directory *filedescriptor*

NFT - Error in renaming file *filedescriptor*

This message is displayed if there is a problem with either an input or an output file name. On local operations, NFT will display the erring file descriptor. For remote rename operations, NFT will display only the input file descriptor (even though it may be the output file name that is in error).

NFT - Error in spooling file *filedescriptor*

NFT - Error in submitting file *filedescriptor*

3.5.3 Secondary Error Messages

Secondary error messages provide additional information on the nature of an error reported by a preceding primary error message. The primary error message indicates the source of the error. Secondary error messages fall into two categories: network errors and file I/O errors.

Network errors pertain to network operation failures or rejections. For example, connect requests can be rejected by the network due to insufficient network resources or due to an invalid node name format being specified in the file descriptor.

File I/O errors pertain to file input or output errors on the local or the remote node. Examples of file I/O errors are an improper file descriptor specification or a file read error.

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 either the remote node or the object could not understand or would not accept the access control information (that is, the access control information -- user ID, password, and account -- does not match 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 the remote node.

Internal consistency error

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 (for example, the name contained illegal characters or was too long).

Invalid object name format

The network (NSP) rejected an attempted connect because the node did not understand the object name format.

Network already open

The user is attempting to access the network for a second time.

Network error code: *n*

The specified error occurred during a network operation. The value reported (*n*) is a signed octal number. If this error occurs, file a software performance report (SPR) documenting the error. 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 maximum number of connections has been made at the remote node. 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 the logical link could not be connected, or the connected logical link was aborted because no path existed to the remote node.

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 in the process of shutting down and would not accept any more 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 has developed in communications between NFT and the remote server task (FAL), causing 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

No room was found in which to create the specified file. The cause of this error pertains to the method of file allocation. For example, there may not be enough room for allocation of a contiguous file. In contrast, a device full error (described in this section) occurs when there is no room available, regardless of how the file is to be allocated.

Bad device name

The specified device name does not conform to the syntax of the target system.

Bad directory syntax

The specified directory does not conform to the syntax of the target system.

Bad file name

The specified file name does not conform to the syntax of the target system.

Bad record size

The specified record size is either invalid or illegal for the specified operation.

Bad version number

Either the version number is out of range, or it has been illegally specified in a delete (see Section 3.2.6) or a rename (see Section 3.2.11) 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 specified 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 to determine 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

There is an error in the contents of the DAP message field. Refer to the microcode for more information.

Message format error

There is an error in the DAP message format. Refer to the microcode for more information.

Message out of sync

There is an error in the DAP message sequence. 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 specified output file cannot be created because no room is available on the output device or there is not enough contiguous space to contain a contiguous file. (In the latter case, some systems will respond with the allocation failure on device message.)

Device off line

The specified file cannot be accessed because the device on which it resides is either 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 specified directory does not exist on the node/device that was specified or defaulted in the file specification.

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 in the process of being modified.

File already exists

The specified file name is already assigned to an existing 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 has been encountered that is illegal for this operation.

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 (that is, sequential, relative, or indexed) is invalid or unsupported by NFT or the remote FAL.

Invalid or unsupported record attributes

The file's record attributes (that is, FORTRAN, implied LF-CR, 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 (that is, fixed, variable, VFC, or stream) is invalid or unsupported by NFT or the remote FAL.

Invalid wildcard operation

The wildcard specification is inappropriate for the specified operation (for example, a wildcard specifier has been included for a partial field on a /RE operation). See Section 3.1.3 for a description of wildcard usage.

No such file(s)

The specified file or files do not exist.

Privilege violation

An operation was specified for which the user does not have privileges.

Rename using two different devices

An attempt has been made 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 needed for the requested operation are not available. 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 has occurred on a spool (/SP), submit (/SB), or execute (/EX) operation when an error has occurred between the remote FAL and the print spooler or the command file/batch submission task. A possible cause of this error is that 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 that may not be supported by the remote FAL are those that represent the following options: contiguous file, maximum version, spool on close, submit on close, and delete on close.

Unsupported file organization

The file organization value (that is, sequential, relative, or indexed) is valid, but is unsupported by either NFT or the remote FAL.

Unsupported record attributes

The file's record attributes (that is, FORTRAN, implied LF-CR, embedded, or VMS print file) are valid, but are unsupported by either NFT or the remote FAL.

Unsupported record format

The file's record format (that is, fixed, variable, VFC, or stream) is valid, but is unsupported by either NFT or the remote FAL.

Wildcard syntax error

There is an error within a wildcard specification (see Section 3.1.3).

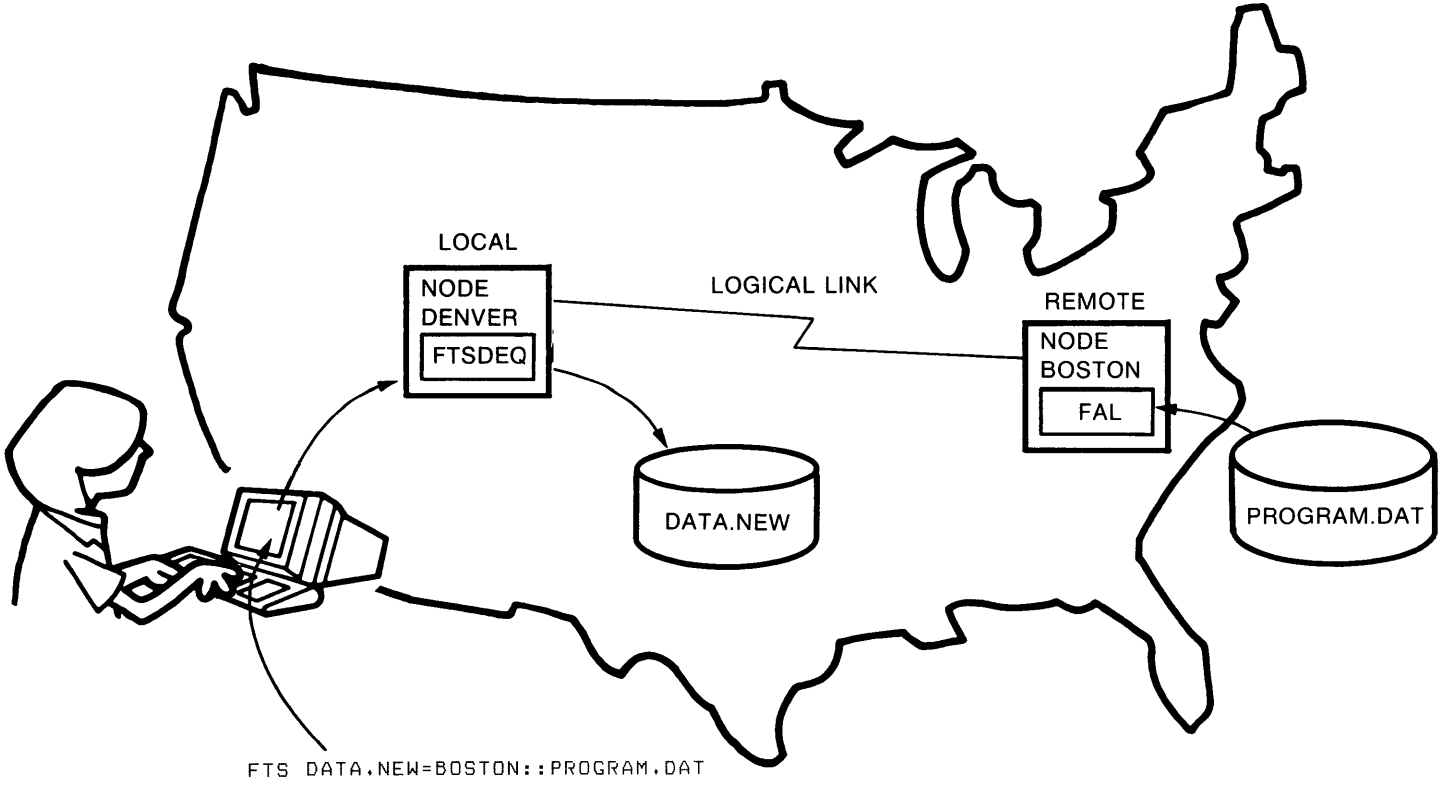
4
The File Transfer Spooler Utility (FTS)

The File Transfer Spooler (FTS) utility has a set of file transfer and manipulation capabilities that are the same as those provided by NFT (see Chapter 3). However, unlike NFT, FTS supports only the MCR Command Language Interpreter (CLI). Those file operations that are valid for both NFT and FTS are:

- 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 provides the following control functions not available in NFT:

- Queue and process user requests according to user-supplied dates and times
- List pending user requests
- Kill pending requests
- Log user request completion



NOTE: FTSDEQ establishes a logical link to FAL on
remote node BOSTON and transfers file PROGRAM.DAT
to node DENVER to create file DATA.NEW.

Figure 4-1: The FTS Utility: Queuing File Transfer Requests

When you type commands to FTS, it builds a queue entry for the FTS dequeuer task on your local node. The FTS dequeuer communicates with the DECnet server task File Access Listener (FAL) on the remote node (see Figure 4-1).

4.1 FTS File Descriptor Syntax

FTS commands operate on file descriptors that identify files in the network. A file descriptor consists of a node specification and a file specification. The format for a file descriptor is:

[*node specification*][*file specification*]

4.1.1 Node Specification

A node specification consists of the name of the node and optional access control information for that node, followed by two colons.

nodename[*access control*]::

where

nodename — is a 1- to 6-alphanumeric character name that includes at least 1 alphabetic character. If the node name is an alias that includes associated access control information, you can omit all access control fields, as they will default to the access control information associated with the alias. For more information on aliases, refer to Section 2.7.

access control — is a user identification, password, and account specification (appended to a node name) used to gain access to a remote file system. Access control information can be specified using one of the two following formats.

NOTE

When passing access control information to case sensitive systems (for example, ULTRIX-32/ULTRIX-32m systems), you must use Format 2, shown here. (Format 1 passes access control information in UPPERCASE.) The embedded spaces in Format 2 are mandatory.

Format 1:

/userid/passwd/accnt.

Format 2:

"userid passwd accnt".

userid is a 1- to 16-character string identifying the user at the remote system.

passwd is a 1- to 8-character string password needed to gain access to the remote file system.

accnt is a 1- to 16-character string used to specify an account at the remote system. (This field is not used for RSX systems.)

The interpretation of both the access control fields and the access control mechanism depends on the type of remote system used. If the remote system is a DECnet-RSX node supporting access control verification, the *userid* field is used to specify 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 additional 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) will permit either privileged or nonprivileged access control. For privileged access, use the privileged password that was specified for FAL during generation of the remote node (the default password is PRIV) and 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 and password are [200,200] and GUEST, respectively. The *accnt* field can be omitted for both privileged and nonprivileged access.

Examples of Node Specifications:

Examples of valid node specifications follow. Missing fields may be omitted from the right. For example, the node specification QUEBEC/[310,2]// can be written QUEBEC/[310,2]. Missing fields on the left must be marked as missing by a slash or a blank (depending upon the format used), as shown in the fourth example, which follows.

NOTE

The use of brackets ([]) around the user ID is optional (as shown in the last two examples, which follow), if the user ID is supplied in the form of a UIC.

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

4.1.2 File Specification

A file specification identifies either the local or the remote file on which FTS is to operate. FTS can parse the same foreign file specifications as NFT (see Section 3.1.4). RSX file specifications have the following format:

dev:[*ufd*]*filename.type*;*ver*

RSX-11M-PLUS and Micro/RSX systems support the defintion of logical names for all or part of a file specification. The DECnet-RSX FAL can translate only global logical names. For a discussion on logical names and how to define them, refer to the *RSX-11M/M-PLUS I/O Operations Reference Manual.*

Depending upon its position in a command string, a file specification is either an input (source) or an output (destination) file specification. Furthermore, the node names define the file specification as local or remote. Therefore, FTS files can fall into four categories:

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

Local or remote output files are files that receive data from an FTS operation, and local or remote input files are files that contain the data to be operated on and/or transmitted. FTS always verifies the existence of local input files before allowing a specified request to be queued. For additional information on RSX files, refer to the *RSX-11 Utilities Manual.*

NOTE

When you enclose a file specification in double quotes, the file spec string is passed to the remote system in a case sensitive manner. This is necessary for transferring files to and from case sensitive systems such as ULTRIX-32/ULTRIX-32m systems.

Defaults for FTS file descriptors are the same as those for NFT file descriptors, which are summarized in Table 3-1. You can change the default values listed in Table 3-1 by using the FTS set default operation (/DF) (see Section 4.2.13).

4.1.3 Wildcards in FTS File Specifications

FTS permits the use of wildcards in local input file specifications only (see the allowable wildcard field specifiers defined for NFT in Section 3.1.3). FTS differs from NFT in that it looks up wildcards immediately and records applicable files in the user request block (URB) file for subsequent execution. Therefore, files that meet the wildcard classification, but that are created after the FTS wildcard request, cannot be included in the URB.

4.2 FTS Operations

FTS operations act on primary switches which specify the operation to be performed (for example, /AP specifies the append operation). Some FTS operations have optional qualifying switches that can further define the action of the primary switch (for example, when /AS/AP are appended to an input file specification, the /AS qualifying switch causes the input file to be appended to an output file in ASCII record mode). The FTS operations are summarized in Table 4-1. Sections 4.2.4 - 4.2.15 describe each FTS operation in detail.

4.2.1 Rules for Specifying FTS Switches in a Command Line

Primary switches can be inserted anywhere in a command line.

Qualifying switches, however, apply only to each specific input file to which they are appended. When qualifying switches are specified for an output file, the switch (or switches) becomes the default for all input file(s) in the command, unless overridden on a per file basis.

When a switch modifies a file specification, the switch cannot appear before the file name, type, version, or UIC of the file specification.

4.2.2 FTS Command Line Continuer

You can continue a command line onto an additional input line by using a hyphen (-) before the line terminator (see the example which follows). The hyphen, followed by the terminator (<RET>), acts as a command line continuer. A command line continuer causes FTS to prompt you for the remainder of the command line. You can insert a continuer at any point in a command line, and you can use any number of continuing lines within a command. The maximum length of a command string is 256 characters.

Example:

```
FTS>DALLAS::DK0:[BUCKLEY]TAXES.LST=FICA.LST,-<RET>
FTS>FEDERAL,STATE<RET>
```

Table 4-1: Summary of FTS Operations

Operation	Primary Switch	Qualifying Switches
Append: The append operation adds files to the end of an existing file.	/AP	/AF /AS /IM /LO /PR /SE
Copy: The copy operation 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: The delete operation deletes one or more files.	/DE	/AF /LO /PR /SE
Execute: The execute operation executes a command file stored on a local or remote node.	/EX	/AF /LO /PR /SE
Help: The help operation displays a descriptive message about switches and commands.	/HE HELP[*subject*]	

(continued on next page)

Table 4-1 (cont.): Summary of FTS Operations

Operation	Primary Switch	Qualifying Swithces
Identify: The identify operation identifies the version of FTS being used.	/ID	
Kill: The kill operation removes a request from the FTS queue.	*/KI	
List: The list operation lists your pending user requests.	*/LI	
Set default: The set default operation sets certain default values for input and output files.	/DF	
Spool: The spool operation spools files to a line printer for printing.	/SP	/AF /LO /PR /SE
Submit: The submit operation copies a command file to a local or remote node, executes the file, and then deletes it.	/SB	/AF /LO /PR /SE

---

*Switches supported by FTS that are not supported by NFT or that have functions different from those for NFT.

4.2.3 Definition of FTS Qualifying Switches

/AS The ASCII switch transfers files in ASCII record mode. The switch applies only to the current command line and must be reset, if desired, for each command.

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

The /AS switch should be used only if translation is desired. File transfers to remote nodes that have similar file systems (for example, VMS) are more efficiently handled in image record mode.

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

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

Format:

FTS>*commandline*/AF[:*dd-mmm-yy*]:hr:mn

where

commandline is the user request that is queued for execution at the specified time.

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

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

Example 1:

```
FTS>DALLAS::DK1:[200,201]=LONDON::SY:[200,200]-<RET>
FTS>PAYROLL.TSK;4/AF:21-MAY-83:15:30<RET>
```

File PAYROLL.TSK;4 will transfer from node LONDON to node DALLAS on May 21, 1983, after 3:30.

Example 2:

```
FTS>LONDON::DK1:[200,200]SALES.LST=SALES.LST;2/AF:17:15<RET>
```

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

/LO The log switch returns a completion message into your file area after processing your request. (A separate message is logged for each file in a multiple file transfer.) FTS logs completion messages by default and stores messages in a user log file. The /LO switch allows you to redefine the user log in which messages are to be placed. If you do not want a completion message to be returned for a user request, disable logging by negating the /LO switch (using /NOLO or /-LO) for each command line.

Format:

FTS>/LO:*filespec*

This format allows you to enter a file specification for the user log file. The specification you enter overrides the default file specification that FTS uses to store completion messages when logging is enabled by the system manager. (This format also causes logging when logging is not enabled.) Any log file specification that you enter with this format defines the user log file only for the current command line. Subsequent lines will use the default or an entered file specification. (The default file type is .LOG.)

FTS>/LO

This format causes logging to default to the default file specification SY0:FTS.LOG when logging is enabled by the system manager.

FTS>/-LO
or
FTS>/NOLO

These negative formats are used to disable logging for the current request when logging is enabled. FTS creates a user log file if the specified file does not exist. If the user log file does exist, FTS appends completion messages to it.

You can use /LO in conjunction with a sequence of user requests (see the /SE switch, which follows). However, you must place the /LO switch on the first command line of the sequence.

Examples:

```
FTS>=TACOMA::DK1:[200,200]BANK.LST<RET>
FTS>=DALLAS::DK1:[200,200]TAXES.LST/LO<RET>
FTS>DENVER::SY0:[200,200]=PAYROLL.CBL,.LST/LO:PAYROLL.LOG<RET>
FTS>BOSTON::DK0:[200,200]=MISC.DOC;4/NOLO<RET>
FTS>GENEVA::DB0:[100,100]NEWSOFT.MAC=RELEASE.MAC/LO:CO:<RET>
FTS>LONDON::DB2:[100,100]=REALTIM.MAC/LO:TI:<RET>
```

The first command logs a message by default to the common system user log file on LB:[1,4]FTSSYS.LOG. The second command logs a message by default to the user's own log file, FTS.LOG. The third command defines the user log file with the specification PAYROLL.LOG. The fourth command line suspends logging for this request only with the /NOLO switch. The fifth and sixth commands illustrate the use of logical device names in the 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.

Completion Message Format:

A completion message has the following format:

nodename type Job = *queue job number*
Time: *dd-mmm-yy start time end time*
User: *userid* Blocks: *size*
Status: *status* [*error code disposition*]
File: *file*
Input: *input*

where

nodename identifies the remote node. If there is no remote node, *nodename* identifies the local node. If the user request was a remote-to-remote file transfer, *nodename* identifies the destination node.

type is a character code that identifies the type of user request issued. The codes are summarized in Table 4-2 in Section 4.2.12.

queue job number is the job number of the user request as listed by the /LI switch (see Section 4.2.12).

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.

userid 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 indicates whether or not the user request completed successfully. If the request did complete with success, the message success is printed in the status field, and the error code and disposition field are omitted. If the request did not complete successfully, the status field contains the phrase error=*nodename*. The node on which the error occurred can be indicated in one of two ways:

LOCAL
LOCAL is printed to indicate that the error occurred at the local node.

errnodnam
The actual name of the remote node on which the error occurred is printed.

The unsuccessful completion of a queued user request returns information in the error code and disposition fields as well.

error code is a 6-character DAP, NSP, or FCS error code that is returned on an unsuccessful operation. Error codes fall into one of the categories described in the following paragraphs. In each case, *nnn* is a positive decimal number that defines a specific set of error codes that each program returns.

FCS*nnn*
describes an error that occurred while 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 that occurred while 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 Appendix C of the *DECnet-RSX Programmer's Reference Manual* for a list of these codes.

N*nnn*07
returns 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 Appendix A of 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 is 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 that an FTS buffer allocation error occurred. 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 Appendix C of 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 Appendix C of 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 that summarizes the completion status of an unsuccessful request. When returned, this field will contain one of the following messages:

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 to check the user's 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 in cases where input and output have no distinction). The file descriptor is in the form

nodename::*dev*:[*uic*]*filename.type*;*version*

If the user request is a delete or execute request, this is the only file descriptor that is printed (that is, the input field is omitted). The node name is omitted in local file descriptors.

input identifies the input file in the form

nodename::*dev*:[*uic*]*filename.type*;*version*

The input file descriptor is repeated for concatenation requests that involve multiple input files. The node name is omitted if the descriptor is for local files.

The following examples show the command lines and resulting completion messages for a single file transfer, concatenation transfer, and remote-to-remote transfer requests.

Example 1: Single file transfer

```
FTS>=DENVER::DK5:[200,200]INVENTORY.LST;4/LO<RET>

DENVER TRL Job=217444
Time: 19-APR-83 17:36:54 17:50:05
User: [200,200] Blocks: 0
Status: error=DENVER N03907 Requeued
File: DBO:[200,1]INVENTORY.LST;4
Input: DENVER::DK5:[200,200]INVENTORY.LST;4
```

One file was queued to transfer to the local system. FTS will use 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 that occurred on the remote system. The error code, N03907, identifies the error as no path to destination node.

Example 2: Concatenation transfer

```
FTS>DALLAS::DKO:[225,225]TAXES.LST=FICA.LST,FEDERAL,STATE/LO<RET>

DALLAS     CLR              Job=382077
Time:   28-JUN-83 11:05:03     11:35:03
User:   [224,1] Blocks:   25
Status:   Success
File:   DALLAS::DKO:[225,225]TAXES.LST
Input:   DBO:   [224,1]FICA.LST
         DBO:[224,1]FEDERAL.LST
         DBO:[224,1]STATE.LST
```

Three input files were transferred and concatenated 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 transfer

```
FTS>BOSTON::DK2:[224,223]=DENVER::DK3:[200,1]PEN.INV<RET>

BOSTON              TRR Job=217526
Time: 12-DEC-83    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.

/PR The priority switch directs FTS to queue a user request at a given priority.

Format:

FTS>/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>MAIN::LB:[200,202]=FISCAL.DOC;1/PR:65<RET>
```

The user request to transfer FISCAL.DOC;1 to MAIN is placed in the queue with 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:

FTS>*commandline1*<RET>
FTS>*commandline2*/SE<RET>
FTS>*commandline3*/SE<RET>
FTS>*commandline4*/SE<RET>
FTS>*commandline5*<RET>
FTS>*commandline6*<RET>

The requests specified in command lines 2 through 5 will be processed in the order that they are input. (Command line 5 is the line subsequent to command line 4 and is, therefore, included in the sequence even though there is no /SE switch typed for command line 5.) If an error is found at any point in the sequence, FTS stops and will not queue the requests for processing. In the above sequence, command lines 1 and 6 are processed independently.

Example:

```
FTS>DALLAS::DK2:[200,202]INVENTORY.LST=*.LST/AS<RET>
FTS>BOSTON::SY:[200,200]PAYROLL.LST=DALLAS::DK0:[200,200] - <RET>
FTS>WAGES.LST;5/AS/SE<RET>
FTS>BOSTON::DK1:[200,200]TAXES.LST;4/DE/SE<RET>
FTS>BOSTON::DK1:[200,200]TAXES.LST=LB:TAXES.CBL,.TSK,.TXT<RET>
FTS>DENVER::DB0::[200,200]PATCH.TSK=FIX.*;*/IM<RET>
```

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

4.2.4 The Append Operation

FTS Primary Switch: /AP

The append operation adds the contents of one or more input files to the end of an existing output file. The output file retains its original attributes independent of the attributes of the input files. However, if the attributes of the input files are different from the attributes of the output file, the append operation will appear to succeed, but the resulting output file may not read correctly.

Format:

FTS>*outfile* = *infile*(*s*)[/*sw*]/AP

where

outfile is the output file descriptor. No wildcard specifiers are allowed in the output file descriptor. The file type and record attributes are taken from the existing file. The file name and file type for the output file must be specified explicitly.

infile(*s*) is one (or more) input file descriptor(s).

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

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

Example:

```
FTS>LOWELL::DK1:FIL1.DAT;1=ESTER::FIL2.DAT;1/AS,YUKON::-<RET>
FTS>FIL3.DAT;1/AS/AP<RET>
```

In this example, FIL1.DAT;1, which is stored on DK1: at node LOWELL, is opened. FIL2DAT;1, located at node ESTER, and FIL3.DAT;1, located on node YUKON, are then appended to FIL1.DAT;1 in ASCII record mode.

4.2.5 The Copy Operation (File Transfer)

FTS Primary Switch: (default operation)

The FTS copy operation can transfer files in the following ways:

- **Single file transfer.** Transfers a single file from one node to another.

- **File transfer with concatenation.** Transfers more than one input file, from the same or different nodes, to create a single output file.

  The attributes of the output file are determined by the attributes of the first input file that you specify in the command line (see the command format, which follows). If the attributes of the input files differ, the concatenation may appear to succeed, but the output file may not be correctly readable.

- **Multiple file transfer.** Transfers more than one input file to the same number of output files, so that each input file has a corresponding output file.

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

If you specify a name and type in the output file specification and there is more than one input file, the input files will be concatenated to form the output file. If an error occurs while processing a concatenation file transfer request, FTS stops processing the request without transferring any of the files.

To specify a multiple file transfer (more than one output file and more than one input file), omit the output file name and type. A node name, device, and UIC may be specified.

Format:

FTS>*outfile* = *infile*(*s*)[/*sw*]

where

infile(*s*) is one (or more) input file descriptor(s).

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

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

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

```
FTS>=DENVER::DK5:[BUCKLEY]INVENTORY.LST;4<RET>
```

In this example, INVENTORY.LST;4 stored on node DENVER is transferred to the local node and stored under the same file name in the UFD [BUCKLEY].

Example 2: Concatenation transfer from the local node to a remote node

```
FTS>DALLAS::DK0:[225,225]TAXES.LST=FICA.LST,-<RET>
FTS>FEDERAL,STATE<RET>
```

In this example, FICA.LST, FEDERAL.LST, and STATE.LST are transferred and concatenated to form TAXES.LST on DALLAS.

Example 3: Multiple transfer from the local node to a remote node

```
FTS>DENVER::DK3:[221,221]=PENS.INV,PAPER,PENCILS<RET>
```

In this example, PENS.INV, PAPER.INV, and PENCILS.INV are transferred to node DENVER and retain the same file names.

Example 4: Single file transfer between two remote nodes

```
FTS>BOSTON::DK2:[FRASER]=DENVER::DK3:[DALEY]-<RET>
FTS>COMPUTERS.LST;4<RET>
```

In this example, COMPUTERS.LST;4 is transferred to node BOSTON from node DENVER, and is stored under the same file name and type. If a file of the same name and type exists on BOSTON, a new version of the file is created.

4.2.6 The Delete Operation

FTS Primary Switch: /DE

The delete operation deletes a list of files (see Format A) or a single input file after it has been transferred (see Format B). The Format B (transfer and delete) option is not offered by NFT.

Format A:

Use the following format to delete one or more files when no transfer operation is being performed:

FTS>*filedescriptor(s)*[*/sw*]/DE

where

filedescriptor(s) — is any valid file descriptor(s), as defined in Section 4.1. The version number must always be specified for files deleted on RSX nodes.

/sw — is one (or more) of the qualifying switches listed under Format B.

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). (See Section 3.1.3 for more information on wildcard specifications.)

Other systems may have different requirements for delete requests. Check the user documentation for each operating system to determine its requirements.

Format B:

Use the following format to transfer a single input file and then delete it from the source. (If the file transfer fails, the input file is not deleted.) Multiple file transfers are not accepted in this format.

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

where

outfile is a file descriptor for the file receiving the data.

infile is a file descriptor for the file that initially contains the data. FTS deletes the input file after it is transferred.

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

/AF
/LO
/PR
/SE

Example 1: Request to delete a list of files

```
FTS>DALLAS::DK1:[FRASER]TAXES.LST;4,DK2:[BUCKLEY]-<RET>
FTS>FICA.LST;3/DE<RET>
```

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

Example 2: Request to delete a file after transfer

```
FTS>LYNN::DK1:[200,200]=USELESS.LST;4/DE<RET>
```

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

4.2.7 The Edit Operation

FTS Primary Switch: /ED

The edit operation inhibits the queuing of user requests. In this state, FTS accepts and checks your command lines for syntax but does not queue them for execution. To return to queuing mode, enter a <CTRL/Z>.

When in edit mode, FTS does detect nonexistent local input files or illegal local input file descriptors. However, when FTS checks the syntax of a user request, it makes no attempt to validate the remote file descriptors. Therefore, an error in the remote file descriptor will go undetected until execution time.

Example:

```
FTS>/ED<RET>

FTS - Entering Edit mode

FTS> <CTRL/Z>

FTS - Leaving Edit mode
```

4.2.8 The Execute Operation

FTS Primary Switch: /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. If you want to execute a command file that does not already exist on a remote node, you must use the submit operation (see Section 4.2.15) to copy the command file to the remote node, and subsequently submit the file for execution.

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

Format:

FTS>*commandfile(s)*[*/sw*]/EX

where

commandfile(s) is the output file descriptor(s) for each remote node command file that you want to execute. Command files are not deleted after execution.

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

/AF
/LO
/PR
/SE

Example:

```
FTS>BOSTON::SY0:[100,100]RETRIEVE.CMD/EX<RET>
```

In this example, the command file RETRIEVE.CMD is to be executed on node BOSTON.

4.2.9 The Help Operation

FTS Primary Switch: /HE
HELP

The help operation displays information on how to use FTS. To print the list of subjects for which there is help information available, type either version of the switch:

FTS>/HE

FTS>HELP

To obtain information about one of the subjects listed, type:

FTS>HELP[*subject*]

Example:

```
FTS>HELP SWITCHES<RET>
```

This help request lists the valid FTS switch options.

Help files must be located on device LB:[1,2]. If they are not on this device, the following message will appear on your terminal:

```
FTS - No help available
```

If your help files are not on the proper device, consult your system manager.

4.2.10 The Identify Operation

FTS Primary Switch: /ID

The identify operation displays the current FTS version number on your terminal.

Example:

```
FTS>/ID<RET>

FTS - VERSION 4.1
```

4.2.11 The Kill Operation

FTS Primary Switch: /KI

The kill operation deletes a pending user request or aborts a request already in progress. Use the kill switch in any of the following forms:

FTS>*job#/KI*	deletes the specified user request.
FTS>*nodename*::/KI	deletes all user requests involving the specified node.
FTS>*/KI	deletes all user requests you have queued.

where

job# is a 6-digit number that FTS assigns to a user request when it is issued. To find the FTS job number, inspect output initiated by /LI (see Section 4.3.12).

nodename is the name of the node for which the request(s) should be killed.

The kill switch deletes or aborts only requests that you have specified. If you attempt to kill a request from a UIC other than the requesting UIC, you will receive an error message.

Killing a job entry involves sending a message to the queue manager to kill the queue entry and deleting the user request file which defines the job to FTSDEQ. When the FTS /KI command is issued, FTS attempts to delete the user request file and to send a request to the queue manager to kill the queue entry. If the request has not been requeued by FTSDEQ due to some failure, the kill request will work. However, when a job has been requeued by FTSDEQ, the queue request is entered under the owner UIC [1,1]. FTS can no longer request the queue manager to kill the queue entry because the queue entry is now owned by [1,1] and not by the original user. FTS may still delete the user request file.

If the queue entry is not in the HELD state and the user request file has been deleted, the entry is deleted the next time FTSDEQ is invoked. However, if the queue entry has been placed in the HELD state, the entry must be killed by a privileged user running under the UIC [1,1]. To kill the queue entry, issue the queue manager command

QUE /DEL/EN:*entry*

Example 1: Delete a single user request

```
FTS>162880/KI<RET>
```

Removes user request with job number 162880 from the queue.

Example 2: Delete all user requests involving a specific node

```
FTS>BOSTON::/KI<RET>
```

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

4.2.12 The List Operation

FTS Primary Switch: /LI

The list operation displays all pending user requests from your UIC on your terminal.

Format:

FTS>[*nodename*::]/LI

where

nodename specifies that only user requests for that node be listed. If *nodename* is omitted, all user requests from your UIC are listed.

When FTS lists pending requests, it prints two header lines. The first contains the message FTS user request listing, the current date, and the time. The second contains the job number, type/priority, and options headings. If there are no queued requests, FTS prints the message no requests found. If there are requests in the queue, but they do not involve the node name specified in your command, FTS prints the headings and the message no requests found.

On the first line of each request description, FTS prints the job number, followed by the code for the function to be performed. (Function codes are defined in Table 4-2.) The function code is followed by a slash and the request's priority. Following the priority are the options the user specified. /IM and /LO are printed by default if logging has been enabled by the system manager and the user did not use the /AS switch to define ASCII transfer mode. The /AF switch with a date and time appears under the options header when a request has been requeued or the user has specified the switch.

The second line of the request description contains the date and time that the request was queued and the number of times that the request 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.) Under the request description are two lines describing the input (source) file and the output (destination) file. A third line describing the user log file is printed if the user has specified a user log file with the /LO:*filespec* switch.

Example 1: Print user requests involving specified node name

```
FTS>DALLAS::/LI<RET>

FTS USER REQUEST LISTING                    28-Jun-83     11:30
Job #          Type/Priority       Options

141954         TLR/50              /IM/LO
 Queued 28-Jun-83 10:45                     Active
 Source File - DBO:[224,1]FICA.LST;1
 Destination File - DALLAS::DK1:[200,200]TAXES.LST

382077         TLR/50              /AF:28-Jun-83:11:35/IM/LO
 Queued 28-Jun-83 10:50                     Tried 1 Time
 Source File - DBO:[224,1]STATE.LST;1
 Destination File - DALLAS::DK1:[200,200]TAXES.LST
 Log File - DBO:DALLAS.LOG
```

Example 2: Print all user requests

```
FTS>/LI<RET>

FTS USER REQUEST LISTING                    28-Jun-83     11:45
Job #          Type/Priority       Options

234892         DR/50               /IM/DE/LO
 Queued 28-Jun-83 9:00                      Active
 Source File - BOSTON:DBO:[224,1]USELESS.LST;4

415121         TLR/50              /AF:28-Jun-83:11:50/IM/LO
 Queued 28-Jun-83 9:15                      Tried 2 Times
 Source File - DBO:[224,1]PAYROLL.CBL
 Destination File - DENVER::SY0:[200,200]PAYROLL.CBL
 Log File - PAYROLL.LOG

415220         TRR/50              /AF:28-Jun-83:11:55/IM/LO
 Queued 28-Jun-83 9:20                      Tried 1 Time
 Source File - BOSTON::DBO:[200,1]TAXES.CBL
 Destination File - LONDON::DK4:[221,221]TAXES.CBL

410455         SLR/50              /AF:28-Jun-83:12:00/IM/LO/SB
 Queued 28-Jun-83 9:25                      Tried 1 Time
 Source File - DBO:[372,1]SALES.LST
 Destination File - GENEVA::SALES.LST

463340         ER                  /AF:28-JUN-83:15:30/IM/EX/LO
 Queued 28-Jun-83 9:30                      Tried 0 Times
 Source File - BURMA::COMMANDS.EXE
```

Table 4-2: 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 file(s) on local node
DR	Delete file(s) on remote node
EL	Execute command file(s) issued for the local node
ER	Execute command file(s) issued for a remote node
PL	Print a file on the local node
PR	Print a file on the remote 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 file(s) to local file(s)
TLR	Transfer local file(s) to remote file(s)
TRL	Transfer remote file(s) to local file(s)
TRR	Transfer remote file(s) to remote file(s)

4.2.13 The Set Default Operation

FTS Primary Switch: /DF

The set default operation allows you to define your own 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, the file name, type, and version number cannot be assigned default values. Default values can be set to affect input files only (by specifying :IN) or output files only (by specifying :OUT). You can define the same default values for both input and output files by using one command and omitting the :IN/:OUT specifiers. Defaults remain in effect for subsequent files until they are changed or until you exit from FTS.

Format:

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

where

defaults is any string of default argument values that you wish to define in the command line. If omitted, FTS displays the last defaults set.

:IN is the input file specifier. When you add this to the end of the command line, :IN indicates that all the default values in the command apply to input files only.

:OUT is the output file specifier. When you add this to the end of the command line, :OUT indicates that all default values defined in the command apply to output files only.

Any file descriptor arguments that are not set with the /DF switch take the standard defaults defined in Table 3-1. To determine what default values are in effect at a given time, type the following command:

```
FTS>/DF<RET>
```

FTS will display the current /DF default values at your terminal.

Example:

```
FTS>INVENT/STATUS/GEARS.:[224,1]/DF:IN<RET>
FTS>DBO:/DF:OUT<RET>
FTS>=INVENTORY.LST<RET>
```

In this example, the first command line defines the input file defaults for node name, access control information (*/userid* and */password*), and UIC. The second command line defines the 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

If you wished to know the default values set in this example, /DF would print the following itemization in response to FTS>/DF<RET>.

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

Destination defaults = DBO:
```

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

4.2.14 The Spool Operation

FTS Primary Switch: /SP

The spool operation directs a file to a line printer on a local or a remote node for printing. When a file is spooled to a remote node, its format must be compatible with the remote system.

Format:

FTS>[*outfile* =]*infile*(*s*)[/*sw*]/SP

where

outfile is an optional file descriptor that specifies the output file to which the file will be copied and the system on which it will be spooled. If *outfile* is omitted, the file specified by *infile* is spooled on the source node.

infile(*s*) is the file descriptor for each file to be spooled for printing.

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

/AF
/LO
/PR
/SE

Example 1:

```
FTS>SITKA::COLD.FIL;1,HOME::SNOW.FIL;1/SP<RET>
```

In this example, one copy of the file COLD.FIL;1 is spooled to the line printer on node SITKA, and one copy of the file SNOW.FIL;1 is spooled to the line printer on node HOME.

Example 2:

```
FTS>HOME::=SITKA::COLD.FIL;1,NOME::SNOW.FIL;1/SP<RET>
```

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 input file names default to the output file name.

4.2.15 The Submit Operation

FTS Primary Switch: /SB

The submit operation transfers a command file to a remote node and submits the file to either a remote command file processor or a batch file processor for execution. The command file is deleted (by the remote node) after the file executes. The remote node must support command file submission and execution (for example, RT-11 nodes do not support command file submission). When you perform a submit operation, remember that the syntax of the command lines within the command file must conform to the syntax standards of the remote system on which the file is to be executed. To execute a command file that is already on a remote node, you can use the execute (/EX) operation (see Section 4.2.8).

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

The success of the submit operation does not guarantee that the batch or command file was executed successfully, only that the file was transferred and that the execution request was successfully given to the MCM.

Format:

FTS>*outfile* = *commandfile*[/*sw*]/SB

where

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

commandfile is the input file descriptor of the file containing the commands to be executed by the destination node's operating system. Only one command file can be specified.

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

/AF
/LO
/PR
/SE

Example:

```
FTS>GENEVA::DK1:[BUCKLEY]NEWFILE.CMD=COMMANDS.CMD;5/SB<RET>
```

File COMMANDS.CMD;5 is transferred to GENEVA as NEWFILE.CMD and is executed, and NEWFILE.CMD is deleted after execution.

4.3 FTS Error Messages

If an error occurs during the processing of a user request, FTS prints one of the following messages:

FTS -- Bad indirect file name

The indirect file name did not conform to the conventions used by the operating system. RSX-11M systems use the at sign (@), followed by the file specification.

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 (see Figure 4-1). 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

One or more of the values that you specified with the /AF switch is illegal. Check your switch specification for an incorrect time or date argument. (/AF is described in Section 4.2.3.)

FTS -- Error in destination file *filedescriptor*

An error occurred in accessing the specified local file. Check the output file descriptor to see if it is correct. The variable *filedescriptor* 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 *filedescriptor*

An error occurred in accessing the specified local file. Check the input file descriptor to see if it is correct. The variable *filedescriptor* 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. Possible secondary error messages are:

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

An operation was specified for which the user does not have privileges.

QMG error code: *n*

The error specified in the primary error message occurred during an operation involving the queue manager (QMG). See Section 4.4 for a listing of the values for *n*, and refer to the *RSX-11 Utilities Manual* for more detailed information.

FTS -- Help file error: *n*

An error has occurred while attempting to use the help function (see Section 4.2.9). The variable *n* is a standard RSX I/O error code.

FTS -- Illegal command

A command line without file descriptors must have one of the following options specified: /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 specified FTS job number does not represent a job currently in the pending request queue.

FTS -- Maximum indirect file depth exceeded

FTS allows only three indirect files to be specified 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 indicated by a /LI switch (see Section 4.2.12).

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 number reported is an unsigned decimal value. Possible values are listed in Section 4.4.

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 legal FTS format.

FTS -- Syntax error in option argument

A switch argument does not conform to legal 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.

4.4 Queue Manager Error Codes

When a queue manager error occurs during the queuing of an FTS user request, the following message prints out:

QMG error code: *n*

The value of *n* can be any of the decimal values listed in Table 4-3.

Table 4-3: Queue Manager Error Codes

Error Code	Meaning
1.	Illegal function code
43.	Privilege violation
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
75.	No queue file space available

(continued on next page)

Table 4-3 (cont.):Queue Manager Error Codes

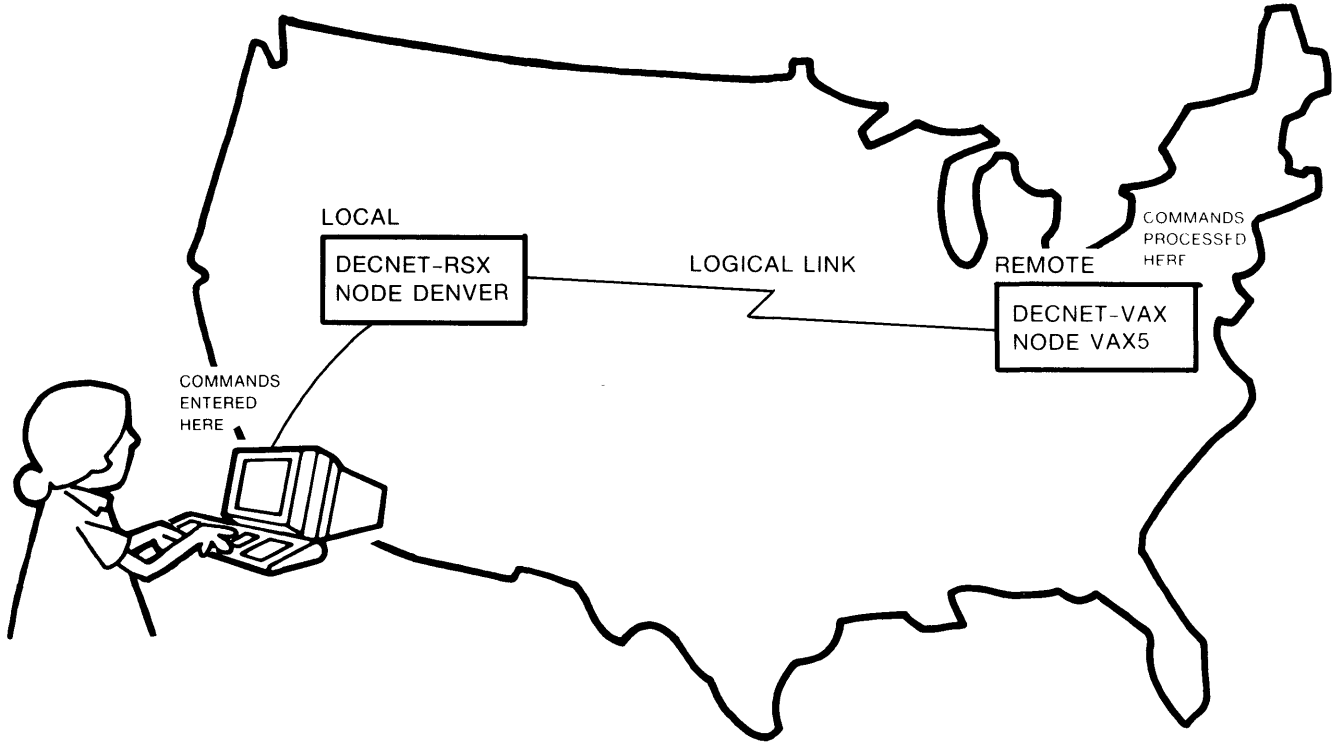
Error Code	Meaning
76.	I/O error on queue file
77.	Redundant operation
78.	Illegal argument value
79.	Executive directive failure
80.	Spooled device bad type
81.	Spooled device does not exist
82.	Spooled device driver off line
83.	Spooled device not loaded
84.	Spooled device redirected
85.	Spooled device 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

5
The Network Command Terminal Utility (NCT)

The Network Command Terminal utility (NCT) allows you to log on to a host system and communicate with all programs that are run under that remote operating system. When using NCT, your terminal is physically connected to the host system (see Figure 5-1). Once you connect to a host, NCT requires that you specify system-specific commands.

NCT functions using the heterogeneous command terminal protocol, CTERM. NCT, or the network command terminal server process, resides on your local node and allows you to logically connect your terminal to any host node which supports the Terminal Services Architecture (TSA). (RTH, the remote command terminal host process on RSX nodes, will also reside on your local node, so that remote users can connect to your local node, via NCT.)

NCT is available on RSX-11M-PLUS, RSX-11M/RSX-11S V4.2 (or greater), and Micro/RSX systems. You should 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.



NOTE: ONCE CONNECTED to the remote VMS host operating system, you must specify operating system level commands to the host

TW139

Figure 5-1: The NCT Utility: Access to Remote Host Operating System Resources

5.1 Connecting to a Host System

To connect to a VMS or RSX host operating system, you must set host to the host node, using the appropriate command format for your local CLI shown here:

MCR Format for DECnet-11M/M-PLUS Systems:

>SET /HOST = *hostnode*[::]

MCR Format for DECnet-11S Systems:

>RUN SETHST <RET>
Host: *hostnode*[::]

Where *hostnode* is the 1- to 6-alphanumeric-character node name (containing at least 1 alphabetic character) of the VMS or RSX host node.

MCR Example (Connect to RSX Host):

```
>SET /HOST=RAISIN <RET>

Connected to "RAISIN", System type = RSX-11M-PLUS
System ID: RSX-11M-PLUS V3.0

>HELLO FRASER <RET>
Password:        <RET> (the password does not echo)

GOOD AFTERNOON

>
```

DCL Format:

$SET HOST *hostnode*

Where *hostnode* is the 1- to 6-alphanumeric-character node name (containing at least 1 alphabetic character) of the VMS or RSX host node.

DCL Example (Connect to VMS Host):

```
$SET HOST VAX5 <RET>

Username: FRASER <RET>
Password:       <RET> (the password does not echo)

                  Welcome to VAX/VMS V4.0 on node VAX5
$
```

5.2 Disconnecting from a Host System

There are two methods of disconnecting from a host system. You can either log off the host system or enter terminal management mode and use the CLEAR HOST command. Both methods are described below.

5.2.1 Logging Off a Host System

If you are connected to an RSX host system, type BYE to log off the host. For example:

```
>BYE <RET>
>
Connect time: 1 minute
CPU time: 1 second
Task total: 7
Have a good afternoon
7-AUG-84 22:02 RT2: logged off RAISIN
>

NCT -- Control returned to node "local node"

>
```

Where *local node* is the name of your local DECnet-RSX node.

NOTE

If you are connected to an RSX host node, you also have the option of specifying BYE/HOLD. Typing BYE/HOLD will return you to the CLI prompt, and although you are still connected to the host, you must log back in if you still want to use the host system resources.

If you are connected to a VMS host system, type LOGOUT to log off the host. For example:

```
$LOGOUT <RET>
FRASER logged out at 7-AUG-84 22:02:58.31

NCT -- Control returned to node "local node"

>
```

Where *local node* is the name of your local DECnet-RSX node.

5.2.2 Entering Terminal Management Mode

Terminal management is a control task which resides on your local node. Terminal management mode allows you to:

- Continue your session on the host node.
- End your session on the host node.

Type <CRTL/\> <RET> to enter terminal management mode. (<CTRL/\> means that you type the <CTRL> key and the back slash (\) key, simultaneously.) When you enter terminal management mode, your local node prompts you for input, as shown in the example below:

```
$<CTRL/\> <RET>
RAISIN>
```

If you want to continue your session on the host node, type CONTINUE <RET> in response to your local node prompt.

If you want to end your session on the host node, type CLEAR HOST <RET> in response to your local node prompt. When you enter the CLEAR HOST command, you will receive a disconnect message from NCT confirming that control has been returned to your local node, as shown in the example below:

```
RAISIN> CLEAR HOST <RET>

NCT -- Control returned to node "RAISIN"

>
```

5.3 Sample NCT Session

In the following NCT session, you connect from your local DECnet-RSX node, RAISIN, to a remote VMS host node, VAX5. Once logged on to the VMS host, you specify the DIRECTORY command to obtain a list of files. You then specify the COPY command to copy PEOPLE.LST;1 to your account on node RAISIN.

```
>SET /HOST=VAX5 <RET>

Username: FRASER <RET>
Password:       <RET> (the password does not echo)

                  Welcome to VAX/VMS V4.0 on node VAX5

$DIRECTORY <RET>


Directory USER$DISK:[FRASER]

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

$COPY <RET>
_From:          VAX5"FRASER PASSWORD"::PEOPLE.LST;1 <RET>
_To:            RAISIN"FRASER PASS2"::PEOPLE.LST;1 <RET>
$LOGOUT <RET>
FRASER logged out at 7-AUG-84 22:02:58.31

NCT -- Control returned to node "RAISIN"

>
```

5.4 NCT Error Messages

NCT returns both connect and disconnect errors. These error types are defined here:

- **Connect Errors.** These errors can be returned by either your local node or the host node, depending on the reason for the error.
- **Disconnect Errors.** These errors occur when a logical link to the host node is aborted by the network.

5.4.1 Connect Errors

NCT -- No more LUNs

There are no available outgoing ports on your system. You can try the connect again. (A port will not be available until another user on your system who is engaged in a remote session ends that session.)

NCT -- Terminal is already attached

Your terminal is attached by another task. You can find out what task is running and abort the task to free up your terminal.

NCT -- No pool available

There is no system pool available for you to use NCT. You can wait awhile and try the connect 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 deemed 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 existed to the host node.

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. You can 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. In this case, NCT will attempt to use RMT as an alternative method of connecting to the host.

If your node supports RMT, you will receive the following message:

```
NCT -- Using RMT
```

If your node does not support RMT, you will receive the following message:

```
NCT -- RMT not available on local node
```

NOTE

RMT cannot be used to connect to a VMS system (see Chapter 6). Therefore, if NCT attempts to use RMT as an alternative means of connecting to a VMS host, an error will result.

5.4.2 Disconnect Errors

NCT -- Connection aborted, Incompatible CTERM versions

The version of CTERM on your local node cannot communicate with the version of CTERM on the host node.

NCT -- Connection aborted, CTERM protocol error

A CTERM protocol error occurred on either the local or host node. 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, 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 path that existed to the host node was broken.

6
The Remote Terminal Utility (RMT)

The Remote Terminal utility (RMT) logically connects your terminal to another DECnet-RSX target node. This connection allows you to log on to the host node and to use its resources as though your terminal were physically connected to the host node. Whenever you use RMT, the local node is the node to which your terminal is physically connected. The host node is the node to which you are logically connected by means of RMT. Figure 6-1 illustrates the RMT utility.

The RMT utility allows you to connect to any host in the network that supports the RMT host facility (RMHACP). When you are connected to the host system by means of RMT, your terminal device is designated HT:. With RMT, you can use any facility that is mutually supported by HT: and the local terminal driver. This excludes the TLK utility, which supports only the TT: device. See Appendix A for QIO functions supported by the HT: driver.

The RMT utility is supported on RSX-11M, RSX-11M-PLUS, and RSX-11S local node operating systems. Likewise, the host system can be any of these three systems.

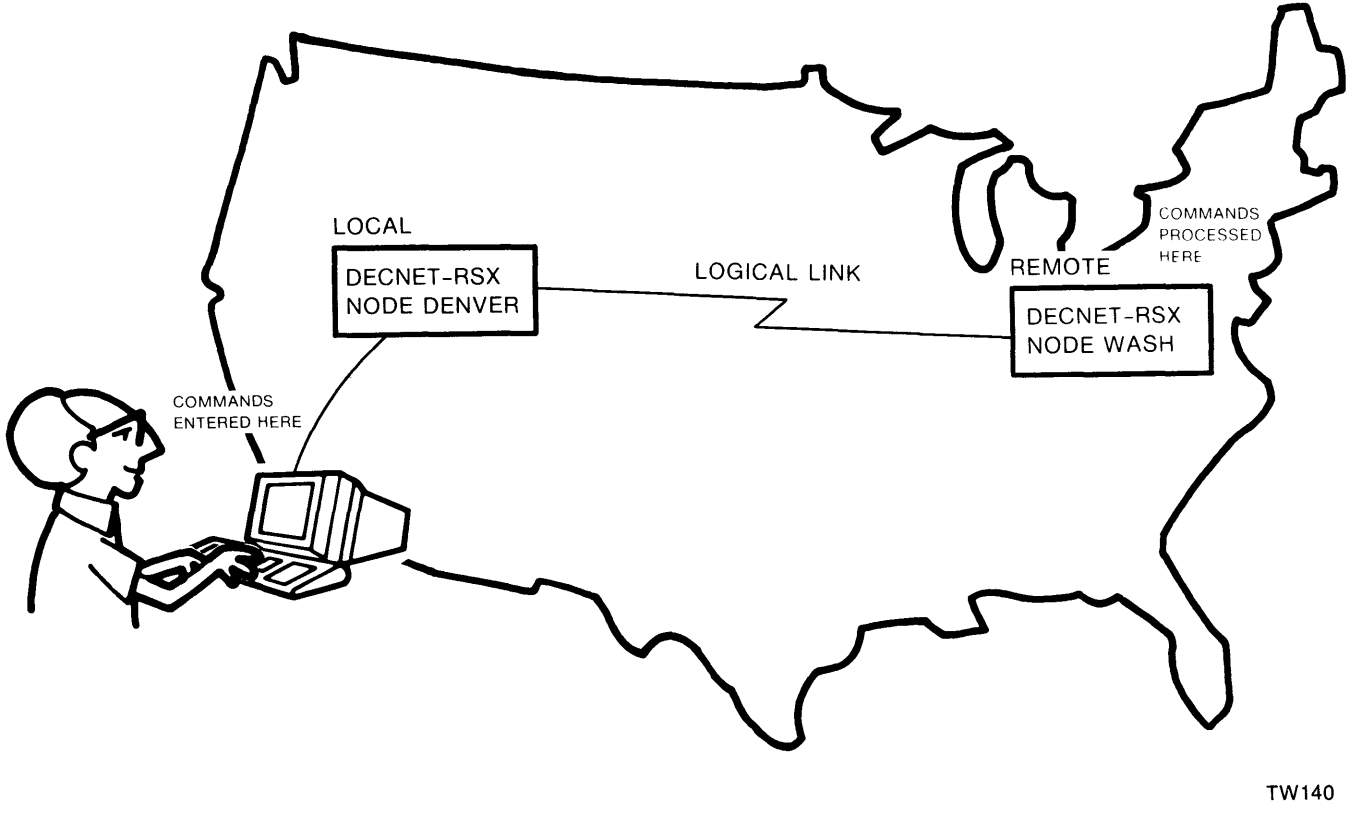


Figure 6-1: The RMT Utility: Access to Remote DECnet-RSX Node Resources

6.1 Connecting to a Host Node

To connect to a host node, you must invoke RMT and supply the name of the node to which you want to connect. There are three methods by which you can do this. Methods 1 and 2 apply to DECnet-11M or DECnet-11M-PLUS operating systems; method 3 applies to DECnet-11S operating systems.

NOTE

> You should not invoke the RMT utility from an indirect command file or from a batch job. The results are unpredictable and depend heavily upon the system and the timing involved.

Method 1: DECnet-11M and DECnet-11M-PLUS Systems

Type RMT after the MCR (or DCL) prompt. RMT then prompts you for a host name. After you enter a host name, RMT establishes the connection and identifies the type of operating system to which you are connected.

```
>RMT<RET>
Host: BASIN<RET>

Connected to "BASIN", System type = RSX11M-Plus
System ID: *DECNET-11M-Plus V3.0 5/24/84*
```

Method 2: DECnet-11M and DECnet-11M-PLUS Systems

Invoke RMT and enter the host node name on the same line:

```
>RMT BASIN<RET>

Connected to "BASIN", System type = RSX11M-Plus
System ID: *DECNET-11M-Plus V3.0 5/24/84*
```

Method 3: DECnet-11S Systems

Issue the RUN command for RMT after an MCR prompt (>):

```
>RUN RMT<RET>
Host: BASIN<RET>

Connected to "BASIN", System type = RSX11M-Plus
System ID: *DECNET-11M-Plus V3.0 5/24/84*
```

The host node name is defined by $HOST and can be changed by the system manager using either the Network Control Program (NCP) or the virtual network processor (VNP). If you omit the host node name, the default host node name is substituted for it. The default host node name is initialized to be 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 detailed information on setting the default host node name.

The following example illustrates a sample RMT log-in to remote host node, MANCE.

Example RMT Log-in:

```
>RMT<RET>
Host: MANCE<RET>

Connected to "MANCE", System type = RSX11M
System ID: MAPPED RSX11M V4.2

>HELLO CASEY<RET>
PASSWORD:        <RET> (the password does not echo)

RSX-11M V4.2 MULTI-USER

GOOD MORNING
3-MAY-85 10:12 LOGGED ON TERMINAL HT3:

>
```

6.2 Disconnecting from a Host Node

When you finish remote terminal operations, type BYE to log off the host system. For multiuser host systems, typing BYE automatically disconnects you from the host node and returns control to the local operating system.

If you want to stay connected to the host node, you can specify BYE/HOLD. Typing BYE/HOLD will return you to the MCR prompt, and although you are still connected to the host, you must log back in if you still want to use the host system resources.

Example:

```
>BYE<RET>
>
HAVE A GOOD AFTERNOON
21-MAY-84 14:20 HT2: LOGGED OFF

RMT -- Control returned to node local node::

>
```

You can also disconnect from the host node by typing EXIT RMT in response to an MCR prompt.

If your terminal is externally forced to log off the local system without first exiting from the RMT task, RMT automatically terminates your connection with the host node.

If, for some reason, the RMT connection to the host node is terminated before you can log off, the HT: on the host will automatically be logged off.

6.3 Aborting RMT

On DECnet-11M-PLUS systems, the Remote Terminal task (RMTACP) is made abortable 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 is aborted, the system data structures may be left corrupted.

If such an abort occurs the users that are connected through RMT can enter a "garbage" line. This line causes RMTACP to be invoked again and resets their terminals to the original state.

If RMTACP is aborted, it should not be removed from the system until all terminals are reconnected to the local system. Premature removal of RMTACP after an abort may lead to a system crash.

6.4 RMT Control Character Handling

RMT handles most control characters (<CTRL/O>, <CTRL/S>, <CTRL/Q>, and <CTRL/R>) according to standard RSX-11 conventions. However, <CTRL/C> causes RMT to prompt for input (RMT>) and then pass that input to the CLI on the host system (generally either MCR or DCL). This procedure differs from conventional RSX-11 usage, where <CTRL/C> directly initiates a local CLI prompt (for example, MCR>).

Example:

```
>MAC @BCASMASM<RET>
<CTRL/C>
RMT>ACT<RET>
MCR...
ACTH3
MACH3
```

6.5 Sample RMT Session

The following example depicts a typical RMT session. The example shows a connect to host node ALIEN from local node TERRAN, the issuing of a PIP command for a directory list, and a subsequent exit from RMT.

```
>RMT ALIEN<RET>

Connected to "ALIEN", System type = RSX11M
System ID: REMOTE TERMINAL TEST SITE

>HELLO DALEY<RET>
PASSWORD:        <RET> (the password does not echo)

RSX-11M V4.2 MULTI-USER SYSTEM

GOOD AFTERNOON
15-MAR-84 14:23 LOGGED ON TERMINAL HT3:

>PIP [354,20]/LI<RET>

DIRECTORY DB0:[354,20]
15-MAR-84 14:27

RMHACP.OLB;10                110.        04-MAR-84 11:52
HTDRV.OLB;10                 19.      C  04-MAR-84 11:53
RMHACP.CMD;7                 14.         06-MAR-84 12:32
<CTRL/C>
RMT>ABO PIP<RET>
14:28:57              TASK "PIPH3" TERMINATED
                      ABORTED VIA DIRECTIVE OR MCR
                      AND WITH PENDING IO REQUESTS
>
>EXIT RMT<RET>

RMT -- Control returned to node TERRAN::

>
```

6.6 RMT Error Messages

If one of the following errors occurs during an RMT session, you are automatically logged off the host system and disconnected from the host node.

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

The node name was entered in an illegal format.

RMT -- Illegal RMT terminal device

The terminal used to invoke 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 number of connections established at the host node exceeds the number allowed.

RMT -- No more remote connections available

The number of local users simultaneously using RMT exceeds the number of remote connections available to RMT.

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 has been abnormally terminated. (When this error occurs, it is likely that the link has been aborted.)

RMT -- Remote connection already established

The terminal used to invoke RMT is already connected to a host node by means of RMT.

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.

7
The Phone Communications Utility (PHO)

The Phone Communications utility (PHO) allows you to engage in an interactive dialog with one or more users on your node or on 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. In addition, PHO provides many advanced features that are not available with the TLK utility (see Chapter 8) such as the ability to "phone" a person by name, to leave a message with someone who was not in when you "phoned" them, or to include the contents of a file in your conversation.

The advanced features of PHO require the use of resource accounting information not available on RSX-11M or RSX-11S systems. Therefore, PHO is available only on RSX-11M-PLUS and Micro/RSX systems.

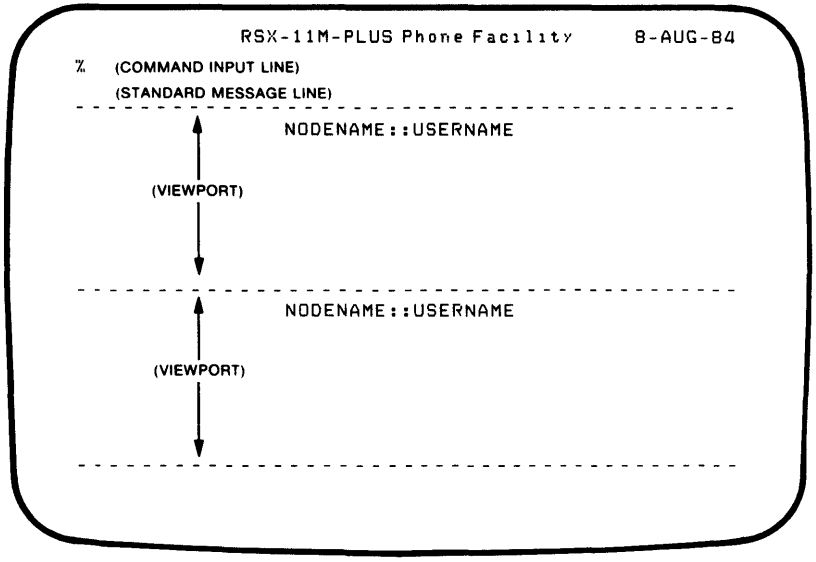
Tables 7-1 and 7-2 list and describe the PHO qualifiers and commands, respectively. Section 7.4 describes each PHO command in detail. Table 7-3 describes the control characters used for formatting your PHO sessions. Figure 7-1 depicts the screen format for the PHO utility. Figure 7-2 depicts a sample PHO conversation.

7.1 PHO Screen Format

PHO can be used only on video terminals with direct cursor positioning, such as VT200, VT100, and VT52 terminals. When PHO is invoked, your screen will appear much like the sample screen shown in Figure 7-1.

Each participant in a phone conversation has his/her own viewport area on the screen. Each participant's viewport is marked with his/her DECnet node name and user name and is separated from other participants' viewports with a broken line (see Figures 7-1, 7-2). Each participant's conversation text appears in his/her own viewport (along with various status messages, such as HOLD messages).

PHO can display up to six viewports at one time. Therefore, people that you have on hold can be temporarily eliminated from your screen to make room for new participants.



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 7-1: The PHO Utility: Sample Screen Format

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.
----------------------------------------
```

Figure 7-2: The PHO Utility: Sample PHO Conversation

7.2 Conference Calls

Conference calls allow three or more users to engage in an interactive, concurrent PHO session. When setting up a conference call, one participant should act as the operator. This person should set up the conference by calling the other participants in turn and waiting for a response.

Only users who are currently participating in a conference call can bring new participants into the conference with the DIAL command (see Section 7.4.2). If you (or any of your fellow conference call participants) should dial a person who is currently engaged in his/her own conference call, you will talk only to that person.

7.3 Invoking Phone

To invoke PHO, use the following command format:

PHONE[*/qualifier(s)*] [*command*]

where

/qualifier(s) is one or more of the qualifiers that modify the characteristics of a PHO session. Table 7-1 lists and describes the PHO qualifiers.

command is one of the PHO commands that is to be executed before PHO prompts you for additional commands. Table 7-2 lists and briefly describes each PHO command. Section 7.4 describes each PHO command in detail.

NOTE

When invoking PHO or when specifying any PHO command, you can truncate the word PHONE and any of its command keywords to the first three letters.

7.3.1 Exiting from PHO

Typing either <CTRL/Z> or the EXIT command will exit you from PHO and return control to your Command Language Interpreter (CLI). (If you are engaged in a PHO conversation, typing <CTRL/Z> will hang up your phone, and you will then have to type either <CTRL/Z>, again, or use the EXIT command to exit from PHO.)

Table 7-1: Summary of PHO Qualifiers

Qualifier	Function
/SCROLL /NOSCROLL	Specifies how the text in your viewport (the area on your screen where your conversation text is displayed) is scrolled when your viewport becomes full. When /SCROLL is specified, your conversation text is scrolled up one line, and a new line of text appears on the bottom line of your viewport. When /NOSCROLL is specified, new text entering your viewport is wrapped and appears on the top line of your viewport. (Specifying /NOSCROLL improves the 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 bottom line of dashes.) The valid range of lines in a viewport is 3 to 10, with a default of 10 lines. (Smaller viewports improve the response time on slow-speed terminals.)
/SWITCH_HOOK=*c*	Specifies the PHO switch hook character for a particular PHO session. The switch hook character is typed in a PHO session to instruct the PHO utility to treat the text following the character as a PHO command. Text typed without this preceding character is considered part of the PHO conversation. The default switch hook character is the percent sign (%). For example, if you are engaged in a PHO conversation and you want to know if Joe Smith is available to phone on node YUKON, you can type the switch hook character and enter the command DIR YUKON <RET>, and the phone directory for node YUKON will be displayed on your screen. When you subsequently type any character key, you will be returned to your phone conversation. The switch hook character is always displayed in column 1 of the command input line (see Figure 7-1).

Table 7-2: Summary of PHO Commands

Command	Meaning
ANSWER	Answers the phone when you receive a call.
DIAL	Phones another user.
DIRECTORY	Displays a list of users on a specified node that you can phone.
EXIT	Exits you from PHO. (Typing <CTRL/Z> is equivalent to entering the EXIT command.)
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 entering the HANGUP command.)
HELP	Displays information on how to use PHO.
HOLD UNHOLD	Places all users who are currently engaged in a PHO session with you on hold. UNHOLD negates a previous HOLD command.
MAIL	Sends a short MAIL message (no longer than a single line) to another user.
REJECT	Rejects a phone call from another user.

Table 7-3 lists the control characters that can be used for formatting a PHO session. Control characters that are not listed in Table 7-3 are ignored by the PHO utility.

Table 7-3: Summary of PHO Control Characters

Character	Formatting Function As Part of a Command	Formatting Function As Part of 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 both your terminal and the terminal of all participants involved 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 your CLI. (Equivalent to EXIT command.)	Hangs up your own phone. (Equivalent to HANGUP command.)

7.4 PHO Commands

This section contains a detailed description and format for each PHO command. All PHO command keywords can be truncated to the first three letters. Some PHO commands require additional information after the command keyword.

If you are using PHO but are not engaged in a conversation, you can enter PHO commands after the switch hook character prompt (see Figure 7-1). In this case, you need not type the switch hook character before entering a PHO command. However, if you are engaged in a phone conversation, you must type the appropriate switch hook character before specifying a PHO command. (Typing the switch hook character instructs the PHO utility to treat the text following that character as a command. Once the command has been executed, you can type <RET> and you will be returned to your phone conversation.)

Table 7-3 describes the control characters used for formatting your PHO session.

7.4.1 ANSWER Command

The ANSWER command answers the phone when you receive a call from another user. If someone calls you when you are not using the PHO utility, the following message will be broadcast to your terminal (at approximate 10 second intervals) indicating who is phoning you and the exact time that you are being phoned. This message will continue to be broadcast to your terminal until you answer the phone, or until the person calling you hangs up the phone:

```
BASIN::SMITH is phoning you on node YUKON:: (14:00:05)

BASIN::SMITH is phoning you on node YUKON:: (14:00:11)
```

(If someone calls you while you are using PHO, the above message(s) will appear in PHO's standard message line, located under the command input line; see Figure 7-1.)

If you want to prevent any calls from reaching your terminal, you should set your terminal to /nobroadcast status.

When you receive a message that someone is phoning you, you can do one of the following three things:

- Answer the phone, using the following command:

PHONE ANSWER

- Reject the call (see Section 7.4.10)

- Ignore the call

Note that the message notifying you that you are being phoned continues to be broadcast to your terminal until you answer the phone, or until the person calling you hangs up the phone.

7.4.2 DIAL Command

The DIAL command places a call to another user. If the user you want to call is on the same node as you, then you need only specify the DIAL command along with the user name. If the user you want to call is on a remote node, you must specify the DIAL command along with the person's DECnet node name and user name. The format for the DIAL command is:

PHONE DIAL [*nodename*::]*user-name*

An alias can be substituted for the *nodename*:: and/or *user-name* parameters.

Example:

```
>PHONE DIAL BASIN::SMITH <RET>
```

When you enter the above command, PHO broadcasts a message to Smith's terminal that he is being phoned by you. (See the ANSWER command, Section 7.4.1 for a more detailed description of this broadcast message.) This message will be broadcast to Smith's terminal at approximate 10 second intervals until one of the following happens:

- Smith answers your call.

- Smith rejects your call.

- You hang up the phone.

NOTE

If you enter a user name without specifying any PHO command, the DIAL command is assumed. Therefore, it is possible to phone another user (for example user Smith on node BASIN), by deleting the word DIAL from the command and using a simplified command such as the one shown below:

PHONE BASIN::SMITH

This command will assume the DIAL command and notify user Smith on node BASIN that you are phoning him.

7.4.3 DIRECTORY Command

The DIRECTORY command displays a list of users that you can phone on a given node. If you specify the DIRECTORY command without any additional information, PHO will display the list of users on your node. If you specify the DIRECTORY command along with a remote node name, PHO will display the list of users on that node. The format for the DIRECTORY command is:

PHONE DIRECTORY [*nodename*[::]]

An alias can be substituted for the *nodename*[::] parameter.

The directory is displayed on your screen, line by line, until the entire user list is displayed or until you type any key on the keyboard. If you are engaged in a PHO conversation, typing any key on the keyboard cancels the directory listing and returns you to your conversation. If you are not engaged in a PHO conversation, this action returns you to the PHO command input line.

The following information is included in the directory display:

- 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:

```
 > PHONE <RET>

                RSX-11M PLUS Phone Facility       15-AUG-84

%DIRECTORY YUKON <RET>
Press any key to cancel directory listing and continue.
------------------------------------------------------------
Process Name        User Name    Terminal      Phone Status

BRAUNSTEIN          BRAUNSTEIN   unusable      ---
Larry               AUGUSTUS     TT2           available
Joe Smith           SMITH        TT4           /nobroadcast
Bob Fraser          FRASER       TT13          available

4 persons listed
```

7.4.4 EXIT Command

The EXIT command exits you from PHO and returns control to the Command Language Interpreter (CLI). (When you enter the EXIT command, PHO automatically executes the HANGUP command; see Section 7.4.6.)

Typing <CTRL/Z>, when you are not engaged in a conversation, is equivalent to entering the EXIT command.

7.4.5 FACSIMILE Command

The FACSIMILE command includes the contents of a file in your phone conversation. The specified file appears in the viewport of every user involved in the PHO session. PHO will continue to display the contents of the file until it reaches the end of the file or until you type any key on your keyboard. The format for the FACSIMILE command is:

FACSIMILE *filespec*

7.4.6 HANGUP Command

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 you issue a HANGUP command, the other participant(s) in your conversation will receive the following message:

username `just hung up the phone`

where *username* is the name of the participant who has hung up his/her phone. (If you hang up your phone, then *username* would be your name.)

Typing <CTRL/Z> while engaged in a conversation is equivalent to entering the HANGUP command.

7.4.7 HELP Command

The HELP command provides you with information about the PHO utility. To obtain general information about PHO, type:

PHONE HELP

To obtain specific information about an individual command or topic, type HELP followed by the command or topic, as shown below:

PHONE HELP *topic*

The information you request is displayed on your screen until it is finished or until you type any key on your keyboard. If you are engaged in a PHO conversation, typing any key on your keyboard returns you to your conversation. If you are not engaged in a PHO conversation, this action returns you to the PHO command input line.

7.4.8 HOLD/UNHOLD Commands

The HOLD command places everyone who is currently on the phone with you on hold (including everyone who has you on hold). PHO informs each person involved in the conversation that he/she has been placed on hold. (PHO allows you to call other users while you have someone on hold.)

The UNHOLD command negates a previous HOLD command. Any user who has previously been placed on hold by means of the HOLD command is removed from hold by specifying the UNHOLD command and is returned to any prior conversation. (However, any user who has been placed on hold before the previous HOLD command remains on hold.)

7.4.9 MAIL Command

The MAIL command sends a short message (no longer than a single line) to another user. For example, if you phoned someone who was not in, you could leave a message to have them phone you back.

If the person to whom you want to send MAIL is on your node, you need only specify the MAIL command along with the user name and the message itself (enclosed in quotation marks). If the person to whom you want to send MAIL is on a remote node, you must specify the MAIL command along with that person's DECnet node name and user name, and the message itself (enclosed in quotation marks). The format for the MAIL command is:

MAIL [*nodename*::]*user-name* "*single line message*"

An alias can be specified for the *nodename*:: and/or *user-name* parameter(s).

7.4.10 REJECT Command

The REJECT command rejects a phone call from another user. The person calling you will receive a message at his terminal that the call has been rejected.

The REJECT command can be used with the optional parameter, EXIT. If the EXIT parameter is specified, PHO performs an EXIT command following the rejection of the call and returns control to the CLI. The format for the REJECT command is:

PHONE REJECT [EXIT]

You can also define REJECT in your login command file as the PHONE REJECT EXIT command, so that you need only specify the word REJECT in order to reject a call and return control to your CLI.

7.5 PHO Diagnostic Messages

PHO returns diagnostic messages which inform you of the status of the PHO utility and of any errors that may occur while you are using (or attempting to use) PHO. The following is a list of some of the most common PHO diagnostic messages. These messages have been categorized into two sections:

- **Informational Messages.** PHO returns informational messages which inform you of someone's attempt to phone you, of the current action and status of the utility while you are using it, and of actions taken by yourself (or other participants in your PHO conversation), such as setting up conference calls or terminating the conversation.

- **Error Messages.** PHO returns error messages which inform you of any error conditions that occur while you are using, or attempting to use, PHO.

7.5.1 Informational Messages

username **has set up a conference call with** *username.*

The person with whom you are conversing has included another individual in the conversation (see Section 7.2 on conference calls).

username **just hung up the phone.**

The person with whom you were conversing has terminated the conversation (see Section 7.4.6, HANGUP command).

nodename::*username* **is phoning you on** *nodename*::

A user identified by his/her DECnet *nodename*::*username* is attempting to establish a PHO conversation with you on your local node (specified by *nodename*::). You have the option of answering, rejecting, or ignoring the call (see Section 7.4.1, ANSWER command).

Attempting to answer...

PHO has received your command to answer an incoming call and is attempting to answer the call. This message is displayed from the time you attempt to answer a call until a connection is established with the caller.

Establishing DECnet link...

PHO has received your command to place a call to a remote user and is currently establishing a DECnet logical link to that user. This message is displayed from the time you place a call until a connection is established to the remote PHO utility.

Press any key to cancel the directory listing and continue.

The directory listing that you have requested with the DIRECTORY command can be terminated by typing any key on the keyboard (see Section 7.4.3, DIRECTORY command).

Press any key to cancel the help information and continue.

The help information that you have requested with the HELP command can be terminated by typing any key on the keyboard (see Section 7.4.7, HELP command).

Ringing *username*...**(Press any key to cancel call and continue.)**

PHO has established a connection with the specified *username*'s phone and is currently ringing that person. This message is displayed from the time a connection is established with a remote PHO utility until the person you are calling either 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 at this time, and has rejected your call (see Section 7.4.10, REJECT command).

That person just cancelled the call.

The person with whom you were conversing has hung up the phone (see Section 7.4.6, HANGUP command).

That person's phone is busy right now. Try again later.

The "phone" (or terminal), of the person whom you are attempting to call is in a state which prevents it from receiving an incoming call (for example, the terminal is attached to another task).

Sending mail...

PHO is currently sending a mail message to the user that you specified with the MAIL command (see Section 7.4.9, MAIL command).

7.5.2. Error Messages

Extraneous text *text* ignored on command line.

You have included text in a command which PHO is ignoring. (The extraneous text is specifed in the error message, and is shown here as the variable *text*.) PHO will process the command while ignoring the extraneous text.

That person's phone just went dead. Try again.

The logical connection to the remote node of the person you are phoning has been lost. You should attempt to phone that person again.

An error has occurred in remote communication.

PHO has received an error in issuing a network directive.

An error has occurred in transmitting to *nodename*

An error occurred when PHO issued a network transmit directive to the remote node of the person you are phoning (specified by *nodename*).

Help library is missing. Please inform your system manager.

HELP information (returned by specifying the HELP command, Section 7.4.7), is available only when a help library is installed on your system. If no help library exists, you should inform your system manager.

Invalid command entered. Try again.

You have specified a command which PHO cannot interpret. You should check the spelling of the command and/or check the list of valid PHO commands found in Table 7-2.

Invalid help keywords specified. Try HELP.

You have specified help information for a topic which PHO does not recognize. You should specify the HELP command (without a specific topic), to obtain a list of the topics for which help is available.

Invalid specification of node or person. Try again.

You have specified a node name or user name that is not correct. You should 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, or when a person who was previously attempting to phone you has hung up his phone.

No one is on hold now.

You have issued an UNHOLD command while you currently have no one on HOLD.

No one with that name is available at this time.

The person you are attempting to phone is not currently logged on. You should check that you have spelled the user's name correctly and that the person is currently a user on the remote node you specified. You can issue a DIRECTORY command to display the users who are currently available to phone on any given node (see Section 7.4.3, DIRECTORY command).

Only a nodename is allowed in a DIRECTORY command.

The correct syntax of the DIRECTORY command is as follows (see Section 7.4.3 for more information):

DIRECTORY [*nodename*[::]]

That person's phone is unplugged (/NOBROADCAST).

The person you are calling has set his/her terminal to /NOBROADCAST. /NOBROADCAST status prevents that person from receiving (or being notified of), any incoming calls.

That person's terminal cannot be used as a telephone.

The person you are calling is using a type of terminal that cannot be used for PHO conversations (for example, a hardcopy terminal).

There is no room on the screen for another person.

PHO can display up to six viewports at one time. You have therefore exceeded the maximum number of conversations which can be active in a conference call.

Try again, using the following format: MAIL addressee "message"

You have issued a MAIL command that is incorrect. This error message specifies the correct format for the MAIL command. (See Section 7.4.9, MAIL command, for more information).

You cannot place redundant calls.

You cannot phone someone whom you are already conversing with by means of PHO.

You did not specify a person to call. Try again.

You issued a DIAL command without specifying a user name. The format for the DIAL command is as follows (see Section 7.4.2 for more information):

PHONE DIAL [*nodename*::]*username*

Your phone is ringing; you cannot place an outgoing call.

You cannot attempt to phone someone while an incoming call to your terminal is pending.

Unrecognized object at *nodename*

PHO is not installed at the remote node (specified by *nodename*) or the object is not set up properly. You should inform that remote node's system manager of this error.

You cannot use this type of terminal as a telephone.

You cannot use PHO from certain types of terminals (for example, hardcopy terminals).

The network has not been properly loaded or started.

For you to be able to use PHO, the network must be loaded and active. You should consult your system manager if you receive this error, or wait and try again.

8
The Terminal Communications Utility (TLK)

The Terminal Communications utility (TLK) allows you to send single line messages or to engage in an interactive exchange of messages (a dialog) with other terminal users in the network. Using TLK, you can conduct "conversations" with terminal users on your local node or with users located on an accessible remote node (see Figure 8-1). In order to receive and execute TLK requests, a remote node must support the compatible server task for the TLK utility. The Listen utility (LSN) is the server task for the TLK program. 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).

8.1 TLK Communication Modes

TLK provides two communication modes for transmitting messages:

- **Single message mode** allows you to send a single line message to another terminal (see Section 8.2).

- **Dialog mode** allows you to conduct an interactive exchange of messages -- a dialog -- with another terminal (see Section 8.3).

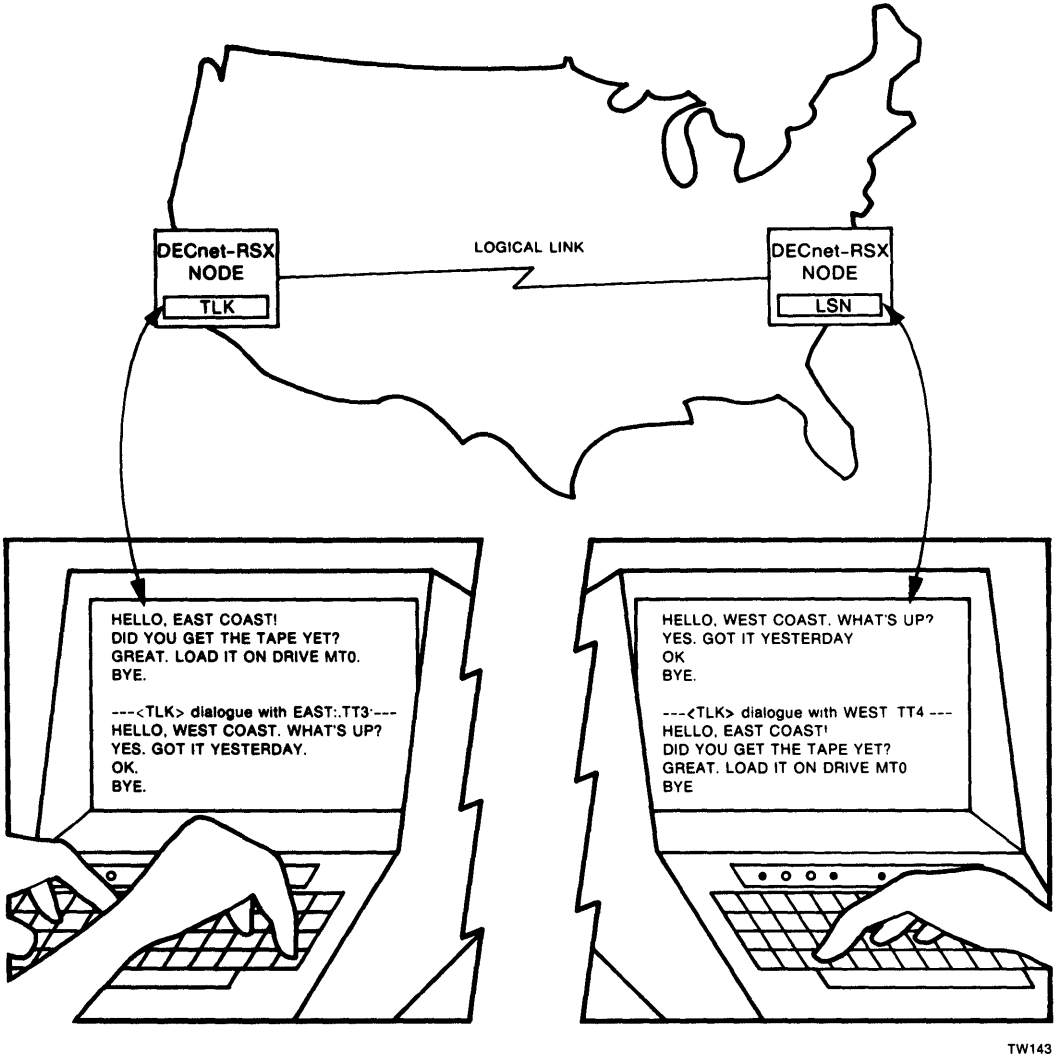


Figure 8-1: The TLK Utility: Communication between Terminals

8.2 Single Message Mode

Using single message mode, you can send a short message or notice (no longer than a single line) to any other terminal in the network that supports TLK.

8.2.1 Format for Sending Messages in Single Message Mode

The format for sending a single-line TLK message is:

>TLK [*targetnode*::][TTn:]'*message*

where

targetnode:: is the 1- to 6-character name of the receiving node to which you are sending the message. This node name must be followed by a double colon. If the receiving terminal is located on the local node (the default node), you can omit this argument.

TT*n*: is the 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:). (See the description in Section 8.3.1 on how to find the TT*n*:.)

message is any ASCII string that you send as your message. This string cannot be extended beyond the end of the command line and is preceded by an apostrophe. (The apostrophe is optional if the terminal device specification (TT*n*:) has been included in the command line.)

Example 1: Sending a Single-line Message to a Remote User

A local user sends a message from his terminal, TT12: on node YUKON, to remote user TT5: on node BASIN (Note that an apostrophe need not precede the message when the target terminal argument is included.)

```
>TLK BASIN: TT5 I AM READY FOR A FILE TRANSFER.<RET>
```

When the message is received, the target terminal displays it along with header information identifying the source node and TTn:, and the date and time that the message was received:

```
<TLK> YUKON::TT12: 16-DEC-83 10:11:30
I AM READY FOR A FILE TRANSFER.
```

As soon as the message has been sent, TLK either prompts you for another message or terminates the TLK session with a return to MCR or DCL, depending on the method you use to invoke TLK. If you are prompted for another message by TLK, but prefer to end the TLK session and return control to MCR or DCL, type <CTRL/Z>.

Example 2: Sending a Single-line Message to a Local User

```
>TLK TT2:'TODAY'S BUDGET MEETING HAS BEEN CANCELED.<RET>
```

Because the target node is omitted in this example, TLK sends the message to TT2: on the local node by default.

8.3 Dialog Mode

Using dialog mode, you can have a conversational exchange of messages with another network terminal user. Dialog mode is interactive. It allows a user at another terminal to respond to you without establishing a new connection. TLK messages can be sent or received by either terminal, regardless of which terminal initiated the dialog. If the user on the remote terminal sends a message while you are in the middle of typing a line, the line you are typing is interrupted, and the message received from the remote terminal is printed. TLK then redisplays your interrupted line on the terminal so that you can resume typing where you left off. See Section 8.3.5 for a sample dialog mode exchange.

If your terminal driver allows IO.RPR and IO.WBT terminal functions (see Appendix A, Table A-1), you will have hard-copy dialog support. With the addition of IO.ATA, you get video dialog support for TLK and LSN.

8.3.1 Format for Sending Messages in Dialog Mode

The format for invoking a TLK dialog is:

```
>TLK [targetnode::][TTn:]<RET>
<TLK> - START OF DIALOG
TLK>message
.
.
.
TLK> <CTRL/Z>
```

where

targetnode:: is the 1- to 6-character name of the receiving node to which you are sending the message. This node name must be followed by a double colon. If the receiving terminal is located on the local node (the default node), you can omit this argument.

TT*n*: is the 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 your message.

In order to send TLK messages, you must enter the target node name and the terminal identifier (TT*n*:) for the user with whom you want to communicate. One method of determining a user's TT*n*: is to use the PHO utility (see Chapter 7, Section 7.4.3) to perform a phone directory for the target node The PHO DIRECTORY command will list information regarding each user currently logged on to the target node, including each user's TT*n*:.

Another way for you to find a user's TT*n*: is to use the RMT utility (see Chapter 6) to log on to the target node and then enter the MCR DEVICE /LOG command or the DCL SHOW USERS command. On an RSX-11M-PLUS system, the command displays information regarding the TT*n*:, the UIC, the date and time of log in, the number of active tasks, and the name of each user presently logged on to that node. (On an RSX-11M system, the command displays only the TT*n*: and the UIC.)

Example:

```
>DEVICE /LOG<RET>

TT11:    [232,201] [350,363] 14-APR-83 09:58  1 B. SMITH
TT12:    [316,314] [350,362] 14-APR-83 10:14  0 J. JONES
TT13:    [350,346] [350,346] 14-APR-83 11:57  1 S. ANDERSON
TT22:    [300,210] [350,377] 14-APR-83 14:55  2 D. PAGE
```

8.3.2 Sending and Receiving Dialog Messages

In the following examples, you are a user on local node YUKON::TT12: who wants to initiate a dialog with a remote user on node BASIN::TT5:. You enter the following command:

```
>TLK BASIN::TT5:<RET>
```

The remote user then receives the following question, to which he must answer YES or NO:

```
<TLK> -- Do you want to talk to YUKON::TT12:?  [Y/N]:
```

NOTE

> If a user who has been solicited for TLK does not answer YES or NO within approximately 10 seconds, he will receive the following message indicating the exact time that he was disconnected from TLK:

```
<TLK> -- Timed out on response (10:15:02)
```

If the remote user answers YES, he receives the following display and prompt for messages:

```
<TLK> IN DIALOG WITH YUKON::TT12: 16-DEC-83 10:14:55
TLK>
```

And you, the initiator of the dialog, receive the start of dialog display and prompt for messages.

```
>TLK BASIN::TT5.<RET>
<TLK> - START OF DIALOG
TLK>
```

Whenever you (or the remote user) receive a TLK> prompt, you can enter a new message line, as shown below:

```
>TLK BASIN::TT5:<RET>
<TLK> - START OF DIALOG
TLK> HI RUSS.  THIS IS JERRY.<RET>
TLK> CAN YOU SEND ME A COPY OF<RET>
TLK> THE MOYNIHAN FILE BY FRIDAY?<RET>
```

8.3.3 Terminating Dialog Mode

When you are finished with your dialog, either you or the remote user can type <CTRL/Z> to terminate. <CTRL/Z> causes the following message to display on both terminals:

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

8.3.4 Dialog Mode Failure Messages

The following messages indicate reasons why an attempt to start a dialog has failed:

TLK -- Dialog mode rejected by remote user

The remote user answers NO to the question prompting him for a TLK dialog and, therefore, the TLK dialog does not occur.

TLK -- Terminal not accessible

The remote user has failed to answer the question prompting him for a TLK dialog within approximately 10 seconds.

TLK -- Terminal busy

The remote user's terminal has been attached by a task for exclusive use.

8.3.5 Sample Dialog Mode Exchange

In the following example, NODEZ initiates a TLK dialog to request files from NODEM. When NODEM sends messages to respond to NODEZ, those messages temporarily interrupt NODEZ's input.

As shown in this example, dialog mode terminates when either user types <CTRL/Z> (displayed as ^Z). (<CTRL/Z> causes the end of dialog message to display on both terminals.)

```
 NODEZ                                NODEM

>TLK NODEM::TT6:  <RET>

                                          <TLK> - DO YOU WANT TO TALK TO
                                               NODEZ::TT2:? [Y/N]:Y <RET>

<TLK> - START OF DIALOG                        <TLK> IN DIALOG WITH NODEZ::TT2:
                                               12-JAN-83 12:30:02
TLK> HELLO NODEM. THIS IS NODEZ <RET>          TLK>
                                               <TLK> HELLO NODEM. THIS IS NODEZ.
TLK> DO YOU READ ME?<RET>                      TLK>
                                               <TLK> DO YOU READ ME?
TLK> I NEED                                    TLK> HELLO NODEZ. YES, I READ YOU.<
<TLK> HELLO NODEZ. YES, I READ YOU.
^R
I NEED THE REAGAN FILE.<RET>                   TLK>
                                               <TLK> I NEED THE REAGAN FILE.
TLK> CAN                                       TLK> I DO NOT HAVE THAT FILE<RET>
<TLK> I DO NOT HAVE THAT FILE.
^R
CAN YOU SEND THE O'NEILL<RET>                  TLK>
                                               <TLK> CAN YOU SEND THE O'NEILL
TLK> FILE INSTEAD THEN?<RET>                   TLK>
                                               <TLK> FILE INSTEAD THEN?

TLK>                                           TLK> I'LL SEND IT AT 3:00 VIA NFT.<
<TLK> I'LL SEND IT AT 3:00 VIA NFT.
TLK> THANKS.<RET>                              TLK>
                                               <TLK>THANKS.
TLK>^Z
<TLK> - END OF DIALOG                          <TLK> - END OF DIALOG
```

8.3.6 Video Screen Dialog

If you use a VT52 or a VT100 display terminal in dialog mode, TLK can provide a split video screen display (that is, the screen is divided in half). The top half of the screen displays the messages that you type, and the bottom half displays the messages typed by the other user. Each half of the screen operates independently and can scroll messages when it is filled. (TLK wraps subsequent lines for terminals operating at transmission rates below 600 baud.) Figures 8-1 and 8-2 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 IO.WBT, IO.RPR, and IO.ATA terminal functions (see Table A-1) must all be supported by your target system. The video mode option need not be supported by the target node, in which case the messages will appear on the remote terminal in normal dialog mode format (see Section 8.3.1). 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:

```
>SET {/VT52  }= TI:<RET>
     {/VT100 }
```

For DCL:

```
>SET TERMINAL {VT52  }<RET>
              {VT100 }
```

Sending and Receiving Video Dialog Messages

To use video dialog mode, invoke TLK as you would for normal dialog mode (see Section 8.3). All of the procedures for initiating and accepting (or rejecting) a dialog (see Section 8.3.2) are the same for video dialog mode as they are for normal dialog mode When the dialog is accepted by the remote user, both terminals automatically enter video dialog mode (assuming that both terminals support the video mode option).

Example:

To initiate a dialog with terminal TT3: on node MISHA (assume here that both terminals are VT100s), enter

```
>TLK MISHA::TT3:<RET>
```

Just as with normal dialog mode, the remote terminal user will be asked if he or she wants to "talk" to you. If the remote user answers YES, then the TLK split screen display will appear on your screen, and you can begin typing your messages.

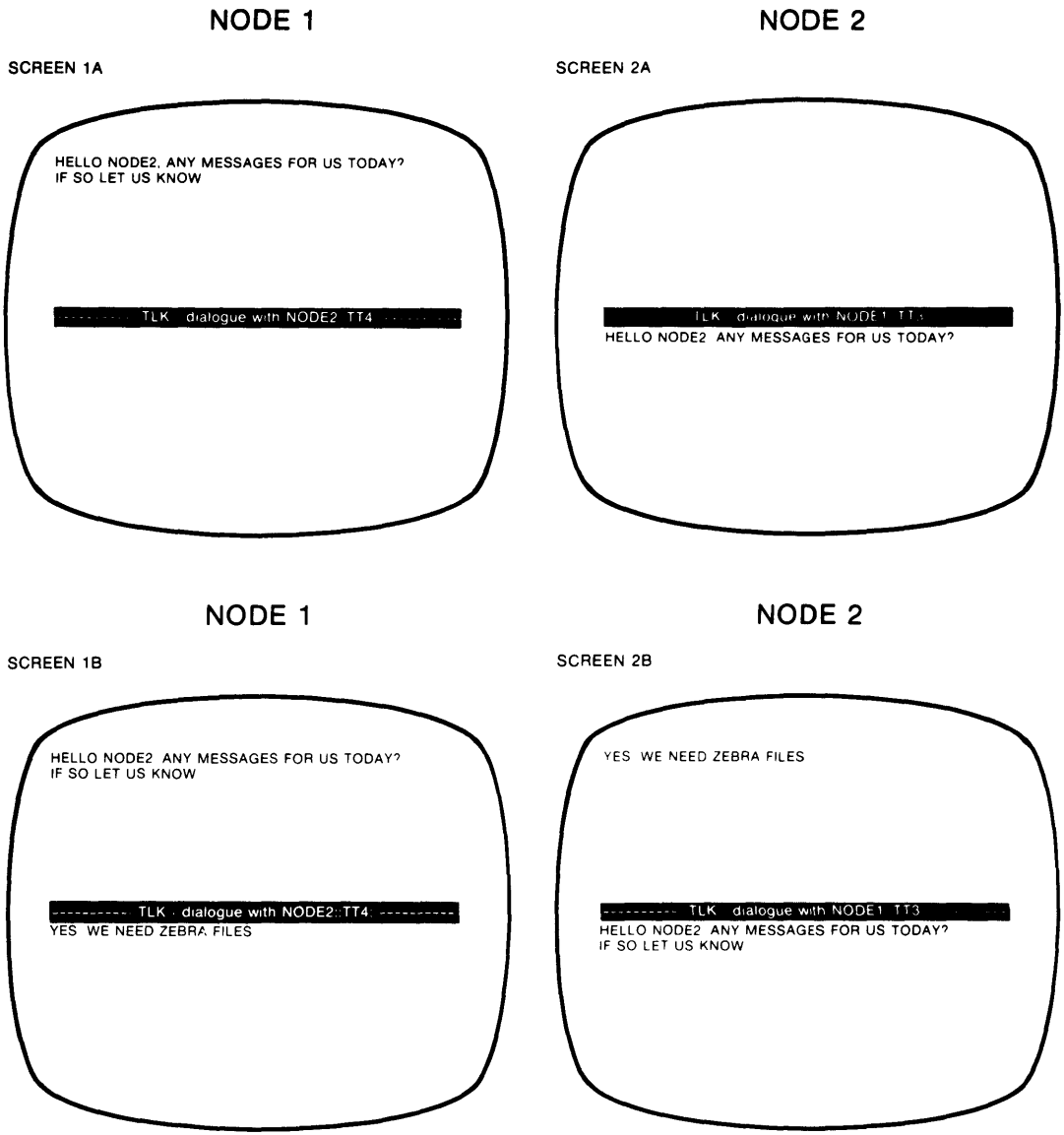


Figure 8-2: The TLK Utility: Sample Split Screen Video Display

As you enter a message, it appears on the upper half of your screen, as shown in Screen 1A of Figure 8-2. Your message line is sent when you press <RET> or when you reach the end of the current line (for example, the 80th character on a VT100). In the latter case, TLK automatically sends the message line. Additional characters that you type appear on the next line. Screen 2A of Figure 8-2 illustrates how your message appears on the receiving terminal of NODE2. Notice that the receiving terminal does not receive your second line until you press <RET>.

Messages from the remote user are displayed 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 8-2).

Terminating Video Dialog Mode

As with normal dialog mode, video dialog mode terminates when either terminal user types a <CTRL/C> or a <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 8-1 describes these special effects.

Table 8-1: Special Terminal Keys for TLK Video Dialog Mode

Key	Effects
<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/C> or <CTRL/Z>	Terminates the TLK session.
<CTRL/R> or <CTRL/W>	Clears your screen and updates the entire display. Useful when a system message or broadcast overwrites part of your screen and you wish to "refresh" the screen.

TLK displays all other control characters, including TAB, as a space.

8.4 Executing TLK Command Files

You can create command files that contain TLK single message mode or dialog mode messages and execute them using the TLK utility. The maximum number of levels of indirect command files is six. DECnet-11S does not support the use of indirect command files.

8.4.1 Executing Single Message Command Files

Single message mode command/message lines that you create inside a TLK command file must adhere to the following format:

[*targetnode*::][TT*n*:]'*message*

where the variable arguments have the same values as those listed in Section 8.2.1.

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.
```

The four separate messages in the example will be sent to the particular node and terminal destination that each message line specifies when you type:

>TLK *@filename*[/TR]<RET>

where

filename is the name of the command file containing the four messages. The default type for a command file is CMD.

/TR is an optional trace switch that causes the command/message lines in the command file to be displayed at your terminal as each line is executed.

Example:

```
>TLK @MONGO/TR<RET>
```

TLK executes the command/message lines in MONGO.CMD and displays them at your terminal.

8.4.2 Executing Dialog Command Files

TLK allows you to execute command files for dialog mode messages. A command file can contain the actual list of dialog messages that you want to send to a remote terminal. Messages in this kind of file should adhere to the following format:

targetnode::TT*n*:
message
message
message
...

To execute such a file, use the same command syntax specified in Section 8.2.1 for single message mode. However, if you do not include the target node name and terminal specification inside the command file, you must specify it in your command line:

>TLK *targetnode*::TT*n*:<RET>
<TLK> - START OF DIALOG
TLK> *@filename*<RET>

A command file can also execute other command files. Files in this category contain the address (that is, target node name and terminal specification) of the target user and a command to execute a secondary command file containing the actual dialog to be sent to the target user. 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.

Files of this type can be executed using the syntax specified in Section 8.2.1 for single message mode.

NOTE

If a command file causes you to enter dialog mode, it will be nonvideo dialog mode, regardless of the type of terminal you are using.

8.5 Exit with Status

TLK exits with a status code that can be useful in the following circumstances:

o In an indirect command file

o When spawned by a parent task or connected to a parent task

TLK can issue 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, an exit status code 2 (EX$ERR) is issued when TLK exits.

You can test the status code returned by TLK and, on the basis of the results, specify alternative procedures.

Example: Indirect command file

```
.IF <EXSTAT> = 1 .GOTO JUNO
```

Consult RSX system documentation for additional information about testing status in indirect command files and testing return status in a parent task.

8.6 TLK Error Messages

If TLK is unable to complete a message transmission, one of the following error messages will print on your terminal.

TLK -- Command line read error

An error was detected in reading the command line. Retype the line in the correct format.

TLK -- Connection aborted

The connection to the target terminal has been aborted by the network. This could be caused by one of three conditions: (1) there has been a line failure between the two nodes, (2) the remote node crashed, or (3) the network in either node has been turned off.

TLK -- Dialog mode not supported

A request for TLK dialog mode has been issued on a system that supports only single message mode.

TLK -- Dialog mode not supported by remote TLK

Dialog mode is not supported by the remote TLK server task. Use single message mode (see Section 8.2).

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

A connection to the remote TLK server task could not be established (for example, the remote node is unreachable).

TLK -- Invalid TLK device

TLK cannot be invoked from a non-TT: device (for example, HT., if you are using RMT).

TLK -- No more connections at remote TLK

The maximum number of connections has been made at the target node. Try again later.

TLK -- No such command file

The system could not find the command file that you specified in your command line.

TLK -- No such node

The target node name that you specified 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

There is a syntax error in the command line. For example, you might have omitted the apostrophe that precedes a single message after you defaulted the device specification, or the double colon that must follow the target node name could be missing.

TLK -- Terminal busy

The target terminal is busy. Either it is engaged with another terminal, or another task has attached the terminal.

TLK -- Terminal not accessible

The operating system at the target node is unable to access the terminal that you specified in your command line. The target terminal could be dedicated to another job, or there may not be a target terminal with the number you specified.

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 (for example, the network software is not installed or is not running).

A
QIO Functions Supported by the HT: Driver

Table A-1 lists the standard and device-specific functions of the QIO macro that can be executed over an RMT line with the HT: driver. Each function must also be supported by the local terminal driver. For detailed information on the parameters and usage of each function, refer to the *RSX-11M/M-PLUS I/O Driver's Reference Manual.* The last column of this table identifies any deviations from the operation as it is performed on the TT: driver.

A.1 HT: QIO Subfunctions

Many of the device-specific functions in Table A-1 can be selected by using subfunction bits. One or more subfunctions can be selected by ORing their relative bits in a QIO macro. Allowable function/subfunction combinations are outlined in the *RSX-11M/M-PLUS I/O Driver's Reference Manual* (this manual also lists additional subfunction codes). The HT: driver supports the following RSX-11M/M-PLUS subfunctions:

TF.AST	TF.NOT	TF.RNE	TF.WAL
TF.BIN	TF.RAL	TF.RST	TF.WBT
TF.CCO	TF.RCU	TF.TMO	TF.XCC

The HT: driver does not support TF.ESQ and TF.XOF.

Table A-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, so you must specify an event flag to ensure that the task is synchronous with the function. You should 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.

(continued on next page)

Table A-1 (cont.): HT: QIO Functions

Function	Description	Comments
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 A.2 and A.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.
IO.RTT	Read logical block ended by specified terminator table	Same as for TT: driver.
SF.SMC	Set multiple characteristics	See Sections A.2 and A.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 between pieces).

A.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 result also depends upon whether the SET command is issued as a QIO from a task or as an MCR SET */characteristic* command. Table A-2 demonstrates the results in all 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 A-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-11 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

A.3 HT: Status Returns

Table A-3 lists the error and completion status codes that can be returned by the HT: driver. 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 A-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 ready
IE.IFC	Illegal function
IE.NOD	Buffer allocation failure
IE.OFL	Device off line
IE.PRI	Privilege violation
IE.SPC	Illegal address space

Index

F

H

P

Q

R

DECnet-RSX
Guide to User Utilities
AA-H223D-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

SOFTWARE DOCUMENTATION
1925 ANDOVER STREET TWO/E07
TEWKSBURY, MASSACHUSETTS 01876

DO NOT CUT FOLD HERE