

Digital SNA Data Transfer Facility for OpenVMS

Use

Part Number: AA-JM75F-TE

May 1996

This manual provides the information necessary to use the Digital SNA Data Transfer Facility (DTF) product.

Revision/Update Information: This is a revised manual.

Operating System and Version: OpenVMS VAX Versions 6.1, 6.2, or 7.0
OpenVMS Alpha Versions 6.1, 6.2, or 7.0
RSX-11M/M-PLUS Version 4.2(M)/Version
4.0(M-PLUS) or later
MS-DOS Version 3.0 or later
OS/2 Version 1.2 or later
MVS/SP, MVS/XA, MVS/ESA
VM/SP, VM/XA
See the SPD for the latest operating
system version.

Software Version: Digital SNA Data Transfer Facility for
OpenVMS, Version 3.2

May 1996

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

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

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

Digital conducts its business in a manner that conserves the environment.

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

Copyright © 1988, 1996 Digital Equipment Corporation, EDS Defense Limited
All Rights Reserved.

The following are trademarks of Digital Equipment Corporation: Alpha, DEC, DEC/CMS, DEC/MSS, DECnet, DECsystem-10, DECSYSTEM-20, DECUS, DECwriter, DIBOL, EduSystem, IAS, MASSBUS, OpenVMS, PDP, PDT, RSTS, RSX, UNIBUS, VAX, VAXcluster, VMS, VT, and the Digital logo.

IBM, RACF, and OS/2 are registered trademarks of International Business Machines Corporation.

CA-ACF2 and CA-TOP SECRET are registered trademarks of Computer Associates International, Inc.

MS-DOS is a registered trademark of Microsoft Corporation.

UNIX is a trademark of American Telephone and Telegraph.

Contents

Preface	xv
1 Introducing the Data Transfer Facility (DTF) Product	
1.1 What Is the DTF Product?	1-1
1.2 What Are the Components of the DTF Product?	1-2
1.3 What Are the Capabilities of the DTF Product?	1-3
1.3.1 Multiple User Interfaces (All Clients)	1-4
1.3.1.1 DECnet User Interfaces	1-5
1.3.1.2 IBM User Interfaces	1-5
1.3.2 Data Translation (All Clients)	1-6
1.3.3 Multiple File Types (All Clients)	1-6
1.3.3.1 Supported Non-IBM Files	1-6
1.3.3.2 Supported IBM Files	1-6
1.3.4 Remote Job Submission and Post Processing (All Clients) ...	1-7
1.3.5 Directory Capability (OpenVMS, Digital UNIX, ULTRIX, RSX-11M/M-PLUS, MS-DOS, and OS/2 Clients)	1-7
1.3.6 Recoverable File Transfer (OpenVMS TRANSFER/DTF and IBM Clients Only)	1-7
1.3.7 RMS Programming Interface (OpenVMS Clients Only)	1-8
1.3.8 DCL Record Level Access and DATATRIEVE Access (OpenVMS Clients Only)	1-8
1.4 What Does the DTF Product Allow Me to Do?	1-8
1.4.1 OpenVMS DCL Commands	1-8
1.4.2 OpenVMS TRANSFER/DTF Commands	1-9
1.4.3 IBM DTF Panel Interface Operations	1-9
1.4.4 IBM DTF File Transfer Commands	1-9
1.4.5 IBM DTF Command Processor Commands	1-9
1.4.6 DECnet-ULTRIX and DECnet/OSI Network File Transfer Commands	1-10
1.4.7 RSX-11M/M-PLUS DCL Commands	1-10
1.4.8 DECnet-RSX NFT Switches	1-10
1.4.9 DECnet-RSX FTS Switches	1-10

1.4.10	PATHWORKS for DOS NFT Commands	1-11
1.4.11	PATHWORKS for OS/2 NFT Commands	1-11
1.4.12	OpenVMS RMS Programming Access	1-11

2 DTF File Specifications

2.1	Specifying Remote IBM Files for DTF File Access	2-1
2.2	Specifying Remote DECnet Files for IBM-Initiated DTF File Access	2-4
2.2.1	Specifying Remote OpenVMS Files	2-5
2.2.2	Specifying Remote RSX-11M/M-PLUS Files	2-6
2.2.3	Specifying Remote MS-DOS or OS/2 Files	2-7
2.2.4	Specifying Remote ULTRIX and Digital UNIX Files	2-8
2.3	IBM File Specification Qualifiers	2-9
2.3.1	Qualifier Defaults	2-9
2.3.2	Qualifier Restrictions When Accessing Remote VM and MVS Clients	2-10
2.3.3	Qualifier Descriptions	2-15
2.4	Setting Up DTF File Definition Files	2-33
2.4.1	Adding an Entry to the File Definition Database	2-34
2.4.2	Modifying an Entry in the File Definition Database	2-35
2.4.3	Displaying File Definitions from the File Definition Database	2-35
2.4.4	File Format and Syntax	2-36
2.4.5	BASE and ALTERNATE_INDEX Sections	2-39
2.4.6	INDEX, DATA, and PATH Subsections	2-39
2.4.7	File Attributes	2-40

3 DTF-Supported File Types

3.1	Supported Input File Types	3-1
3.1.1	Supported File Organizations	3-1
3.1.2	Supported Record Formats	3-2
3.1.3	Supported Record Lengths	3-3
3.1.4	Supported Record Attributes	3-3
3.2	Supported Output File Types	3-4
3.2.1	Supported File Organizations	3-4
3.2.2	Supported Record Formats	3-5
3.2.3	Supported Record Lengths	3-6
3.2.4	Supported Record Attributes	3-7
3.3	File Transfer Considerations	3-7
3.3.1	Transferring Files Between IBM Clients	3-8

3.3.2	Transferring Files Between an IBM System and an Digital UNIX, ULTRIX, RSX-11M/M-PLUS, MS-DOS, or OS/2 Client	3-8
3.3.3	VSAM File Transfer Restrictions	3-8
3.3.4	Printer File Transfer Restrictions	3-9
3.3.5	Transferring IBM Tape-Resident Files	3-9
3.3.6	Transferring Files with Horizontal Tab Characters	3-10
3.3.7	Keyed Access to Sequential Files	3-10

4 Data Security

4.1	Local Data Security	4-1
4.2	Remote Data Security Without Proxy	4-2
4.3	How to Specify an IBM User ID and Password	4-4
4.4	What Is Proxy?	4-4
4.4.1	Benefits of Using Proxy Access	4-4
4.5	DECnet-Initiated Proxy	4-5
4.6	IBM-Initiated Proxy	4-7
4.6.1	Example of IBM-Initiated Transfer Request	4-7

5 OpenVMS File Transfer Operations

5.1	Transferring Files Between OpenVMS DTF Clients and IBM DTF Clients	5-1
5.2	APPEND Command	5-2
5.3	BACKUP Command	5-5
5.4	CLOSE Command	5-8
5.5	CONVERT Command	5-9
5.6	COPY Command	5-11
5.7	CREATE Command	5-14
5.8	DELETE Command	5-16
5.9	DIFFERENCES Command	5-18
5.10	DIRECTORY Command	5-20
5.10.1	Date and Time Restrictions for Created, Revised, and Expires Fields	5-24
5.11	EXCHANGE/NETWORK Command	5-24
5.12	OPEN Command	5-27
5.13	READ Command	5-28
5.14	SEARCH Command	5-29
5.15	SUBMIT/REMOTE Command	5-31
5.16	TYPE Command	5-33
5.17	WRITE Command	5-35

6 OpenVMS TRANSFER/DTF File Transfer Operations

6.1	Invoking the TRANSFER/DTF Utility	6-2
6.2	Exiting the TRANSFER/DTF Utility	6-3
6.3	Copying Files with the TRANSFER/DTF Utility	6-3
6.4	Defining OpenVMS/DTF Queue Attributes	6-14
6.5	Displaying the Status of File Transfer Jobs	6-17
6.6	Displaying OpenVMS/DTF Queue Characteristics	6-19
6.7	Displaying the OpenVMS/DTF Software Version Number	6-20
6.8	Canceling File Transfer Jobs	6-20
6.9	Displaying OpenVMS/DTF Error Messages	6-21

7 Using the DTF for IBM Panel Interface

7.1	The DTF Panel Interface	7-1
7.2	The Operation Selection Panel	7-3
7.3	The Send to Remote Panel	7-5
7.4	The Receive from Remote Panel	7-10
7.5	The Resume Previously Failed Transfer Panel	7-15
7.6	The Transfer Execution Panel	7-18
7.7	The Remote Node Options Panel	7-22
7.8	The MVS Input File Options Panel	7-24
7.9	The VM Input File Options Panel	7-27
7.10	The MVS Output File Options Panel	7-30
7.11	The VM Output File Options Panel	7-34
7.12	The Network File Options Panel	7-38
7.13	The Data Translation Options Panel	7-39
7.14	The Post-Processing Options Panel	7-41
7.15	The Batch Options Panel	7-42

8 Using the DTF for IBM File Transfer Commands

8.1	DTFSEND	8-3
8.2	DTFRECV	8-8
8.3	DTFRESUM	8-13

9 Using the DTF for IBM Command Processor

9.1	Invoking the DTF Command Processor	9-1
9.2	The DTF Command	9-2
9.3	DTF Subcommands	9-2
9.3.1	CLEAR Subcommand	9-4
9.3.2	DEFINE Subcommand	9-5
9.3.2.1	Keywords and Keyvalues for Specifying the Input File	9-7
9.3.2.2	Keywords and Keyvalues for Specifying the Output File	9-12
9.3.2.3	Keywords and Keyvalues for Specifying Transfer Parameters	9-18
9.3.3	END Subcommand	9-22
9.3.4	TRANSFER Subcommand	9-23
9.3.5	HELP [MVS]	9-26
9.3.6	LIST Subcommand	9-28
9.3.7	RECALL Subcommand	9-29
9.3.8	RESUME Subcommand	9-30
9.3.9	SET Subcommand	9-36
9.3.10	SHOW Subcommand	9-38
9.4	DTF Command Processor Usage Considerations	9-38
9.4.1	Using Tapes (MVS Systems)	9-39
9.4.2	Using Tapes (VM Systems)	9-39
9.4.3	Overriding the Server and Server Account	9-39
9.5	Executing the DTF Command Processor in Batch Mode (VM Systems)	9-39
9.6	Executing the DTF Command Processor in Batch Mode (MVS Systems)	9-40
9.6.1	The SYSTSIN DD Card	9-40
9.6.2	Notes and Restrictions for DTF Batch Mode Operations	9-44
9.7	DTF Return Codes in DTF Batch Mode Operations	9-45
9.8	Initiating DTF Command Processor Operations from Command Procedures	9-45
9.8.1	Variables Set by DTF	9-46

10 ULTRIX and UNIX File Transfer Operations

10.1	Transferring Files Between ULTRIX, UNIX, and IBM DTF Clients	10-1
10.2	dcat Command	10-1
10.3	dcp Command	10-3
10.4	Remote Backup and Restore Operations	10-5
10.5	dls Command	10-6
10.6	drm Command	10-8
10.7	Error Reporting	10-9

11 RSX-11M/M-PLUS File Transfer Operations

11.1	Transferring Files Between RSX-11M/M-PLUS DTF Clients and IBM DTF Clients	11-1
11.2	APPEND Command	11-2
11.3	COPY Command	11-4
11.4	CREATE Command	11-7
11.5	DELETE Command	11-9
11.6	DIRECTORY Command	11-11
11.7	SUBMIT/REMOTE Command	11-14
11.8	TYPE Command	11-16
11.9	Setting Default File Information	11-17
11.10	Error Reporting	11-18

12 MS-DOS and OS/2 File Transfer Operations

12.1	Transferring Files Between MS-DOS or OS/2 DTF Clients and IBM DTF Clients	12-1
12.2	APPEND Command	12-2
12.3	COPY Command	12-3
12.4	DELETE Command	12-5
12.5	DIRECTORY Command	12-7
12.6	SUBMIT Command	12-9
12.7	TYPE Command	12-10
12.8	Error Reporting	12-11

A Using OpenVMS DATATRIEVE with DTF

A.1	Defining a DATATRIEVE Domain to Access an IBM File	A-1
A.1.1	Before Defining Your DATATRIEVE Domain	A-1
A.1.2	Using the DATATRIEVE DEFINE Command	A-2
A.2	Preparing a DATATRIEVE Domain for Access to an IBM File	A-2
A.3	Manipulating Records from a Domain on an IBM System	A-3
A.4	Ending a DATATRIEVE Session That Is Accessing an IBM File	A-3

B RMS File System and Programming Constraints

B.1	RMS File System Constraints	B-1
B.2	RMS Programming Restrictions	B-2
B.2.1	Record Level Access to VSAM Files	B-2
B.2.1.1	Unsupported OpenVMS RMS Service Calls	B-2
B.2.1.2	Unsupported RMS Extended Attribute Blocks	B-3
B.2.1.3	Record Locking	B-3
B.2.1.4	Programming Considerations for VSAM Indexed Files	B-4
B.2.1.5	Programming Considerations for VSAM Relative Files	B-4
B.2.1.6	Programming Considerations for VSAM Sequential Files	B-5
B.2.2	Record Level Access to IBM Non-VSAM Files	B-5
B.3	Supported RMS Service Calls	B-5
B.3.1	\$CLOSE	B-7
B.3.2	\$CONNECT	B-7
B.3.3	\$CREATE	B-8
B.3.4	\$DELETE	B-11
B.3.5	\$DISCONNECT	B-11
B.3.6	\$DISPLAY	B-12
B.3.7	\$ERASE	B-14
B.3.8	\$FIND	B-14
B.3.9	\$FREE	B-14
B.3.10	\$GET	B-15
B.3.11	\$OPEN	B-16
B.3.12	\$PARSE	B-19
B.3.13	\$PUT	B-19
B.3.14	\$RELEASE	B-19
B.3.15	\$REWIND	B-19
B.3.16	\$SEARCH	B-20
B.3.17	\$UPDATE	B-20

C Maintaining the OpenVMS/DTF File Definition Database

C.1	Adding Entries to the File Definition Database	C-1
C.2	Displaying File Definition Attributes	C-3
C.3	Modifying a File Definition	C-6
C.3.1	Modifying the File Definition BASE Section	C-6
C.3.2	Modifying File Definition ALTERNATE_INDEX Sections	C-7
C.4	Removing an Entry from the File Definition Database	C-8

D VSAM File Support

D.1	A VSAM Primer for Non-IBM Users	D-1
D.1.1	Sequential and Relative Files	D-1
D.1.2	Indexed Files	D-2
D.2	DTF for IBM and VSAM Naming Conventions	D-5
D.2.1	DTF for IBM Naming Conventions	D-6
D.2.2	VSAM Naming Conventions	D-6
D.2.3	Choosing Your Own Name for ALTERNATE_INDEX	D-7
D.3	VSAM Cluster Attribute Defaults	D-8

E FTAM File Support

E.1	Overview of FTAM	E-1
E.2	FTAM User Facilities	E-2
E.2.1	FTAM Components	E-3
E.2.2	File Transfers Using FTAM Software and a DAP-FTAM Gateway	E-3
E.3	FTAM Document Types	E-6
E.3.1	Restrictions	E-6
E.4	Examples of OSI-Initiated File Transfer Operations	E-8
E.4.1	Examples of File Transfers Using the DAP-FTAM Gateway	E-10

F DTF Data Translation

F.1	Controlling Data Translation on DECnet-Initiated File Transfers	F-1
F.2	Controlling Data Translation on IBM-Initiated File Transfers	F-2
F.3	DMCS/EBCDIC Translation Tables	F-3
F.4	Field-Level Translation	F-3
F.4.1	Creating a New Record Definition	F-3
F.4.2	Supported CDD Data Structures	F-4
F.4.3	Translation of Digital CDD and IBM Data Types	F-4

G Correspondence Between DTF for IBM Keywords and OpenVMS/DTF Qualifiers

Index

Examples

9-1	Sample REXX EXEC to Perform Post-Processing	9-42
9-2	Sample JCL to Perform a Post-Processing Function	9-42
9-3	Sample REXX EXEC to Automatically Recover a Transfer	9-43

Figures

1-1	Software Components in a DTF Environment	1-4
2-1	Using the /MDxPASSWORD Qualifier - Part 1	2-23
2-2	Using the /MDxPASSWORD Qualifier - Part 2	2-24
4-1	Request Verified by MVS Proxy	4-6
4-2	IBM-Initiated Access: DECnet Client Node Not Running DTF	4-8
7-1	Operation Selection Panel	7-3
7-2	Send to Remote Panel	7-5
7-3	Receive from Remote Panel	7-10
7-4	Resume Previously Failed Transfer Panel	7-15
7-5	Transfer Execution Panel (MVS Systems)	7-18
7-6	Transfer Execution Panel (VM Systems)	7-19
7-7	Remote Node Options Panel	7-22
7-8	MVS Input File Options Panel	7-24
7-9	VM Input File Options Panel	7-27
7-10	MVS Output File Options Panel	7-30
7-11	VM Output File Options Panel	7-34
7-12	Network File Options Panel	7-38
7-13	Data Translation Options Panel	7-39
7-14	Post-Processing Options Panel	7-41
7-15	Batch Options Panel (VM Systems)	7-42
7-16	Batch Options Panel (MVS Systems)	7-43
D-1	File Structure of VSAM Sequential or Relative Files	D-2

D-2	File Structure for VSAM Indexed Files	D-3
D-3	File Structure for VSAM Indexed File with Multiple Keys . .	D-5
E-1	File Transfers Using FTAM Software and a DAP-FTAM Gateway	E-4

Tables

2-1	IBM File Specification Qualifiers and Their Usage for Remote VM Files	2-10
2-2	IBM File Specification Qualifiers and Their Usage for Remote MVS Files	2-13
2-3	MVS File Attributes and Their Usage in the BASE and ALTERNATE_INDEX Sections	2-40
2-4	VM File Attributes and Their Usage in the BASE and ALTERNATE_INDEX Sections	2-41
4-1	User ID and Password Syntax Rules	4-2
4-2	Access to Remote Systems Without Proxy	4-3
5-1	DTF Level of Support for APPEND Command Qualifiers . . .	5-3
5-2	DTF Level of Support for BACKUP Command Qualifiers . . .	5-6
5-3	DTF Level of Support for CLOSE Command Qualifiers	5-8
5-4	DTF Level of Support for CONVERT Command Qualifiers	5-9
5-5	DTF Level of Support for COPY Command Qualifiers	5-12
5-6	DTF Level of Support for CREATE Command Qualifiers	5-15
5-7	DTF Level of Support for DELETE Command Qualifiers . . .	5-16
5-8	DTF Level of Support for DIFFERENCES Command Qualifiers	5-18
5-9	DTF Level of Support for DIRECTORY Command Qualifiers	5-20
5-10	DTF Level of Support for EXCHANGE/NETWORK Command Qualifiers	5-25
5-11	DTF Level of Support for OPEN Command Qualifiers	5-27
5-12	DTF Level of Support for READ Command Qualifiers	5-28
5-13	DTF Level of Support for SEARCH Command Qualifiers . . .	5-29
5-14	DTF Level of Support for SUBMIT/REMOTE Command Qualifiers	5-31
5-15	DTF Level of Support for TYPE Command Qualifiers	5-34
5-16	DTF Level of Support for WRITE Command Qualifiers	5-35

6-1	Recoverable File Operations	6-1
6-2	TRANSFER/DTF Support for DCL COPY Command Qualifiers	6-4
6-3	Unique TRANSFER/DTF COPY Command Qualifiers	6-5
10-1	DTF Level of Support for dcat Command Options	10-2
10-2	DTF Level of Support for dcp Command Options	10-4
10-3	DTF Level of Support for dls Command Options	10-6
11-1	Supported RSX-11M/M-PLUS File Transfer Commands	11-1
11-2	DTF Level of Support for APPEND Command Qualifying Switches	11-2
11-3	DTF Level of Support for COPY Command Qualifying Switches	11-5
11-4	DTF Level of Support for CREATE Command Qualifying Switches	11-7
11-5	DTF Level of Support for DELETE Command Qualifying Switches	11-10
11-6	DTF Level of Support for DIRECTORY Command Qualifying Switches	11-12
11-7	DTF Level of Support for SUBMIT/REMOTE Command Qualifying Switches	11-14
11-8	DTF Level of Support for TYPE Command Qualifiers	11-16
11-9	DAP Error Codes	11-18
12-1	DTF Level of Support for APPEND Command Switches	12-2
12-2	DTF Level of Support for COPY Command Switches	12-4
12-3	OpenVMS/DTF Level of Support for DELETE Command Switch	12-6
12-4	DTF Level of Support for DIRECTORY Command Switches	12-8
12-5	DTF Level of Support for SUBMIT Command Switch	12-9
12-6	DTF Level of Support for TYPE Command Switch	12-11
12-7	DAP Error Codes	12-12
B-1	\$CLOSE FAB and XAB Input Fields	B-7
B-2	\$CONNECT RAB Input Fields	B-7
B-3	\$CREATE FAB Input Fields	B-8
B-4	\$CREATE XAB Output Fields	B-9
B-5	\$DELETE RAB Input Fields	B-11
B-6	\$DISPLAY NAM Input Fields	B-12
B-7	\$DISPLAY FAB and XAB Output Fields	B-12

B-8	\$FIND RAB Input Fields	B-14
B-9	\$GET RAB Input Fields	B-15
B-10	\$OPEN FAB Input Fields	B-16
B-11	\$OPEN XAB Output Fields	B-17
B-12	\$PUT RAB Input Fields	B-19
B-13	\$RELEASE RAB Input Fields	B-19
B-14	\$UPDATE RAB Input and Output Fields	B-20
E-1	FTAM User Facilities Supported by DTF Software	E-2
E-2	Restrictions	E-6
E-3	DTF Support and Mapping Among FTAM, OpenVMS, and IBM (MVS and VM) File Attributes	E-7
F-1	Correspondence of CDD Data Types and IBM Data Types . . .	F-4
F-2	Limitations on Data Transferred from Digital to IBM	F-6
F-3	Limitations on Data Transferred from IBM to Digital	F-7
G-1	Correspondence Between OpenVMS/DTF Qualifiers, DTF for IBM Command Processor Keywords, and DTF ISPF Entry Fields	G-1

Preface

The Digital SNA Data Transfer Facility (DTF) software is a Digital Equipment Corporation product that allows you to transfer files between IBM MVS and VM systems in an SNA network and systems in a Digital DECnet network. To use the DTF access routine, you must first install the appropriate versions of the software and hardware that you plan to use. Refer to the Software Product Description (SPD) for this information.

The DTF access routine consists of two major parts: the IBM-resident software (DTF for IBM systems) and the Digital-resident software (OpenVMS/DTF). The IBM-resident software can reside on a VM or MVS system (DTF for VM or DTF for MVS).

Manual Objective

This manual explains how to use the DTF product to access and transfer files between OpenVMS, ULTRIX, RSX-11M/M-PLUS, MS-DOS, and OS/2 systems in a Digital DECnet network and MVS and VM systems in an IBM SNA network.

Intended Audience

This manual is intended for the user who has access to the DTF software for the purpose of transferring files between systems. DTF software can be accessed from OpenVMS, DECnet-ULTRIX, DECnet-RSX, PATHWORKS for DOS, PATHWORKS for OS/2, IBM MVS, and IBM VM systems.

Changes and New Features

The Digital SNA Data Transfer Facility for OpenVMS, Version 3.2 differs from the Version 3.1 product only in that it includes support for utilizing TCP/IP to exchange messages with a cooperating application on the IBM host and the Digital SNA Domain Gateway or Digital SNA Peer Server.

The information relevant to TCP/IP transport support include the following:

- SNA_TCP_PORT logical
- SNA_TRANSPORT_ORDER logical
- Specifying TCP/IP hostnames

SNA_TCP_PORT Logical

The SNA_TCP_PORT logical refers to the remote connection TCP/IP port. The default connection TCP/IP port number is 108. For example, if you want the remote connection TCP/IP port number to be 1234, you can enter the following command line:

```
$ define SNA_TCP_PORT 1234
```

If you want the remote connection TCP/IP port to be made to a service defined and enabled in the UCX database; for example *service_name*, you can enter the following command line:

```
$ define SNA_TCP_PORT service_name
```

SNA_TRANSPORT_ORDER Logical

The SNA_TRANSPORT_ORDER logical refers to a transport list, which is used in automatic selection of transports. Connections are attempted once for each transport in the list until either a successful connection is made, or an error is returned when all transports in the list fail to connect.

For example, if you want the software to try the DECnet transport and if this fails then to try the TCP/IP transport, you can enter the following command line:

```
$ define SNA_TRANSPORT_ORDER "decnet, tcp"
```

If you want the software to try the TCP/IP transport and if this fails then to try the DECnet transport, you can enter the following command line:

```
$ define SNA_TRANSPORT_ORDER "tcp, decnet"
```

If you want the software to never try the DECnet transport and to try only the TCP/IP transport, you can enter the following command line:

```
$ define SNA_TRANSPORT_ORDER "nodecnet, tcp"
```

If you want the software to never try the TCP/IP transport and to try only the DECnet transport, you can enter the following command line:


```
$ define SNA_TRANSPORT_ORDER "decnet, notcp"
```

Note

If the SNA_TRANSPORT_ORDER logical is not defined, the default transport order for OpenVMS Alpha will be decnet, tcp; and the default transport order for OpenVMS VAX will be local, decnet, tcp.

Specifying TCP/IP Hostnames

If you want to specify a full path hostname, the hostname must be enclosed in a pair of double-quotes; for example, "foo.bar.company.com".

If you want the TCP/IP transport to be used as the preferred transport, without specifying a TCP/IP full path hostname, then define the SNA_TRANSPORT_ORDER with "tcp" as the first element in the transport list.

If the hostname ends with a single full-colon (":"), then the TCP/IP transport will be used; for example, "foo:" or foo:.

Note

If you specify a double full-colon (":::"), you force the DECnet transport to be used; for example, "foo:::" or foo::.

Manual Structure

This manual is divided into the following twelve chapters and seven appendixes:

Chapter 1	Introduces the DTF software and describes the OpenVMS /DTF software components.
Chapter 2	Describes the DTF file specifications and IBM file definition attributes.
Chapter 3	Describes the file types supported by DTF and restrictions DTF places on file types and file transfers.
Chapter 4	Describes data security with or without proxy.
Chapter 5	Describes how to use the OpenVMS DCL command interface to perform tasks using the DTF software.

Chapter 6	Describes how to use the OpenVMS TRANSFER/DTF utility to initiate recoverable file transfers. It also describes commands for monitoring and controlling recoverable file transfers.
Chapter 7	Describes how to use the menu-driven MVS and VM DTF panel interface.
Chapter 8	Describes how to use the basic MVS and VM DTF file transfer commands.
Chapter 9	Describes how to use the robust MVS and VM DTF command processor.
Chapter 10	Describes how to use DECnet-ULTRIX and DECNet/OSI commands to perform tasks using DTF software.
Chapter 11	Describes how to use the DECnet-RSX commands to perform tasks using the DTF software.
Chapter 12	Describes how to use PATHWORKS for DOS and PATHWORKS for OS/2 NFT commands to perform tasks using DTF software.
Appendix A	Describes how to use OpenVMS DATATRIEVE to access IBM file types using DTF.
Appendix B	Discusses the OpenVMS Record Management Services (RMS) programming interface and limitations to be aware of when you use DTF.
Appendix C	Gives an overview of the file definition commands of the OpenVMS SNADTFCFG utility.
Appendix D	Describes VSAM file structures and provides additional information about VSAM files.
Appendix F	Discusses IBM and Digital data types and the way DTF translates data between Digital and IBM systems.
Appendix E	Discusses the FTAM file types supported by DTF-FTAM.
Appendix G	Discusses the correspondence between OpenVMS/DTF qualifiers and DTF for IBM command processor keywords and ISPF panel entry fields.

The postage-prepaid Reader's Comments form on the last page of this manual is for your critical evaluation to assist us in preparing future documentation.

Associated Documents

The Digital SNA Data Transfer Facility software documentation consists of the following manuals:

- *Digital SNA Data Transfer Facility for OpenVMS Installation*
- *Digital SNA Data Transfer Facility for OpenVMS Management*
- *Digital SNA Data Transfer Facility for OpenVMS Problem Solving and Messages*
- *Digital SNA Data Transfer Facility for OpenVMS Use*

You should have the following Digital documents available for reference when you use the Digital SNA Data Transfer Facility software:

- *Common Data Dictionary Summary Description*
- *Common Data Dictionary User's Guide*
- *Common Data Dictionary Utilities Reference*
- *Common Data Dictionary Data Definition Language Reference*
- *Digital SNA Domain Gateway Installation*
- *Digital SNA Domain Gateway Management*
- *Digital SNA Domain Gateway Guide to IBM Resource Definition*
- *DECnet SNA Gateway Problem Determination Guide*
- *DECnet SNA Gateway-CT Installation*
- *DECnet SNA Gateway-CT Problem Solving (OpenVMS & ULTRIX)*
- *DECnet SNA Gateway-CT Management (OpenVMS)*
- *DECnet SNA Gateway-CT Guide to IBM Parameters*
- *DECnet SNA Gateway-ST Installation*
- *DECnet SNA Gateway-ST Problem Solving (OpenVMS)*
- *DECnet SNA Gateway-ST Guide to IBM Parameters*
- *DECnet SNA Gateway Management for OpenVMS*
- *Digital Peer Server Installation and Configuration*
- *Digital Peer Server Management*
- *Digital Peer Server Network Control Language Reference*
- *Digital Peer Server Guide to IBM Resource Definition*

- *OpenVMS SNA Installation*
- *OpenVMS SNA Problem Solving*
- *OpenVMS SNA Guide to IBM Parameters*
- *OpenVMS SNA Management*
- *OpenVMS SNA Problem Determination Guide*

OpenVMS client users may also find the following Digital OpenVMS manuals to be helpful:

- *OpenVMS User's Manual*
- *OpenVMS Record Management Utilities Reference Manual*
- *DEC DATATRIEVE User's Guide*
- *OpenVMS DCL Dictionary*
- *VMS DCL Concepts Manual*
- *Guide to OpenVMS File Applications*

ULTRIX client users may also find the following Digital DECnet-ULTRIX manual to be helpful:

- *DECnet-ULTRIX Use*

Digital UNIX client users can also find the following Digital DECnet/OSI manual helpful.

- *DECnet/OSI for Digital UNIX Introductory User's Guide*

RSX-11M/M-PLUS client users may also find the following Digital DECnet-RSX manual to be helpful:

- *DECnet-RSX Guide to User Utilities*

DOS client users may also find the following PATHWORKS for DOS manual to be helpful:

- *PATHWORKS for DOS DECnet User's Guide*

OS/2 client users may also find the following PATHWORKS for OS/2 manual to be helpful:

- *PATHWORKS for OS/2 Utilities Guide*

IBM client users may also find the following IBM manuals to be helpful:

- *JCL Reference*
- *CMS Primer*

- *IBM Access Method Services*

The following IBM manuals provide additional reference information that could be helpful in problem determination:

- *DFP: Access Method Services Reference*
- *IBM VTAM Customization*
- *MVS JCL Reference Manual*
- *MVS Service Aids Manual*
- *MVS System Commands*
- *MVS System Messages*
- *VTAM Operator Commands*
- *VM Planning and Reference*
- *VM/SP6 Connectivity, Planning, and Administration*
- *VM/SP5 TSAF*

Acronyms

The following acronyms are used throughout this manual:

ASCII	American Standard Code for Information Interchange
DCL	Digital Command Language
DMCS	Digital Multinational Character Set
DTF	Digital SNA Data Transfer Facility software
EBCDIC	Extended Binary Coded Decimal Interchange Code
ESDS	Entry Sequenced Data Set
GDG	Generation Data Group
HSM	IBM's Hierarchical Storage Manager
IBM MVS	IBM's Multiple Virtual Storage system
IBM SNA	IBM's Systems Network Architecture
IBM VM	IBM's Virtual Machine system
ISPF	Interactive System Productivity Facility
KSDS	Key Sequenced Data Set
MS-DOS	Microsoft Corporation's DOS operating system
NFT	Network File Transfer utility

PDS	Partitioned Data Set
REXX	Restructured Extended Executor
RRDS	Relative Record Data Set
SMS	IBM's Storage Management Subsystem
SNADTF CFG	SNA DTF Configuration utility
TRANSFER/DTF	SNA DTF Transfer utility
TSO	IBM's Time Sharing Option
ULTRIX	Digital's UNIX-based operating system
OpenVMS/DTF	Digital-resident Data Transfer Facility software
OpenVMS	The OpenVMS operating system
VSAM	IBM's Virtual Storage Access Method
VTAM	IBM's Virtual Telecommunications Access Method

Graphic Conventions

The following conventions are used throughout this manual:

Special type	This special type in examples indicates system output or user input.
UPPERCASE	Uppercase letters in command lines indicate keywords that must be entered. You can enter keywords in either uppercase or lowercase. You can abbreviate command keywords to the first three characters or their minimum unique abbreviation.
lowercase bold	Lowercase bolding is used for ULTRIX and Digital UNIX commands, command options, and arguments. Lowercase bolded words must be entered in lowercase.
<i>lowercase italics</i>	Lowercase italics in command syntax or examples indicate variables for which either the user or the system supplies a value.
CTRL <i>x</i>	Hold down the CTRL key and press the key specified by <i>x</i> .
[]	Square brackets in command syntax statements indicate that the enclosed values are optional. You can enter none or one. Default values apply for unspecified options. (Do not type the brackets.)
Return	Press the RETURN key. The RETURN key, which you must press to execute commands, is assumed in command examples and therefore is not shown in command displays.

Enter

Press the ENTER key. The ENTER key is used to enter IBM commands.

VM

Instructions or comments that apply to IBM's VM operating system only. Also used to indicate instructions or comments that apply only when accessing files on an IBM VM system.

MVS

Instructions or comments that apply to IBM's MVS operating system only. Also used to indicate instructions or comments that apply only when accessing files on an IBM MVS system.

OpenVMS

Indicates a Digital-to-Digital (OpenVMS only) process.

Introducing the Data Transfer Facility (DTF) Product

This chapter answers the following major questions about the Digital SNA Data Transfer Facility (DTF) product:

- What is the DTF product?
- What are the components of the DTF product?
- What are the capabilities of the DTF product?
- What does the DTF product allow me to do?

1.1 What Is the DTF Product?

The Digital SNA Data Transfer Facility software is a DECnet SNA access routine that connects a OpenVMS/DTF server node and its clients on a DECnet network. IBM MVS and VM client systems on an SNA network. The DTF software supports the following DECnet clients:

- OpenVMS
- ULTRIX
- Digital UNIX
- RSX-11M/M-PLUS
- MS-DOS
- OS/2

Note

You can use TCP/IP between the OpenVMS/DTF server node and the Digital SNA Domain Gateway or Digital SNA Peer Server.

The purpose of DTF is to extend the DECnet file transfer facilities to the IBM MVS and VM environments; therefore, the DTF software does not support file transfers between two DECnet clients. You should use the DECnet file transfer utilities for this purpose.

On OpenVMS clients, DTF supports the DCL interface, the TRANSFER/DTF utility supplied with OpenVMS/DTF, the RMS programming interface, and the VAX DATATRIEVE product. On RSX-11M/M-PLUS clients, DTF supports the DCL interface and the DECnet file transfer commands. On ULTRIX, Digital UNIX, MS-DOS, and OS/2 clients, DTF supports the DECnet file transfer commands. On IBM clients, DTF supports a menu-driven panel interface, a basic set of file transfer commands, and the more robust DTF command processor.

1.2 What Are the Components of the DTF Product?

The DTF access routine requires two major pieces of software: the Digital-resident software (OpenVMS/DTF) and the IBM-resident software (DTF for IBM).

The OpenVMS/DTF software is packaged into the following two software installation kits:

- **OpenVMS/DTF server software.** The OpenVMS/DTF server software kit is installed on the OpenVMS node chosen to be the OpenVMS/DTF server node. The server node performs data translation (if required) and transfers the data between DTF clients. The OpenVMS/DTF server software includes the OpenVMS/DTF utilities software, so you must not install both kits on the server node. An OpenVMS/DTF server can serve a large number of clients.
- **OpenVMS/DTF utilities software.** The OpenVMS/DTF utilities software kit is optional software that can be installed on additional OpenVMS nodes chosen to be DTF client nodes. The client nodes communicate with the OpenVMS/DTF server node. The OpenVMS/DTF utilities software includes the following two utilities:
 - **The TRANSFER/DTF utility:** Allows you to recover and continue transferring files after a communications failure.
 - **The SNADTFCFG utility:** Allows a server manager to configure the OpenVMS/DTF server and create file definitions that are used to create IBM files. This utility also lets a server manager monitor and control several databases on the OpenVMS/DTF server node.

All IBM clients are required to have DTF for IBM software installed. The IBM-resident software is packaged in two software installation kits: one kit for MVS systems and one kit for VM systems. The IBM software takes requests from IBM users and forwards those requests to the OpenVMS/DTF server. The software also handles transfer requests from the OpenVMS/DTF server. The IBM-resident software includes four logical components:

- The DTF command processor accepts requests from IBM users and passes these requests to the local Network Manager.
- The Network Manager establishes network connections. For IBM-initiated requests, the connections are between local File Managers and remote nodes. For DECnet-initiated requests, the connections are between remote nodes and the local Node Manager.
- The Node Manager receives incoming requests and selects File Managers to handle the requests.
- File Managers control the actual file transfers.

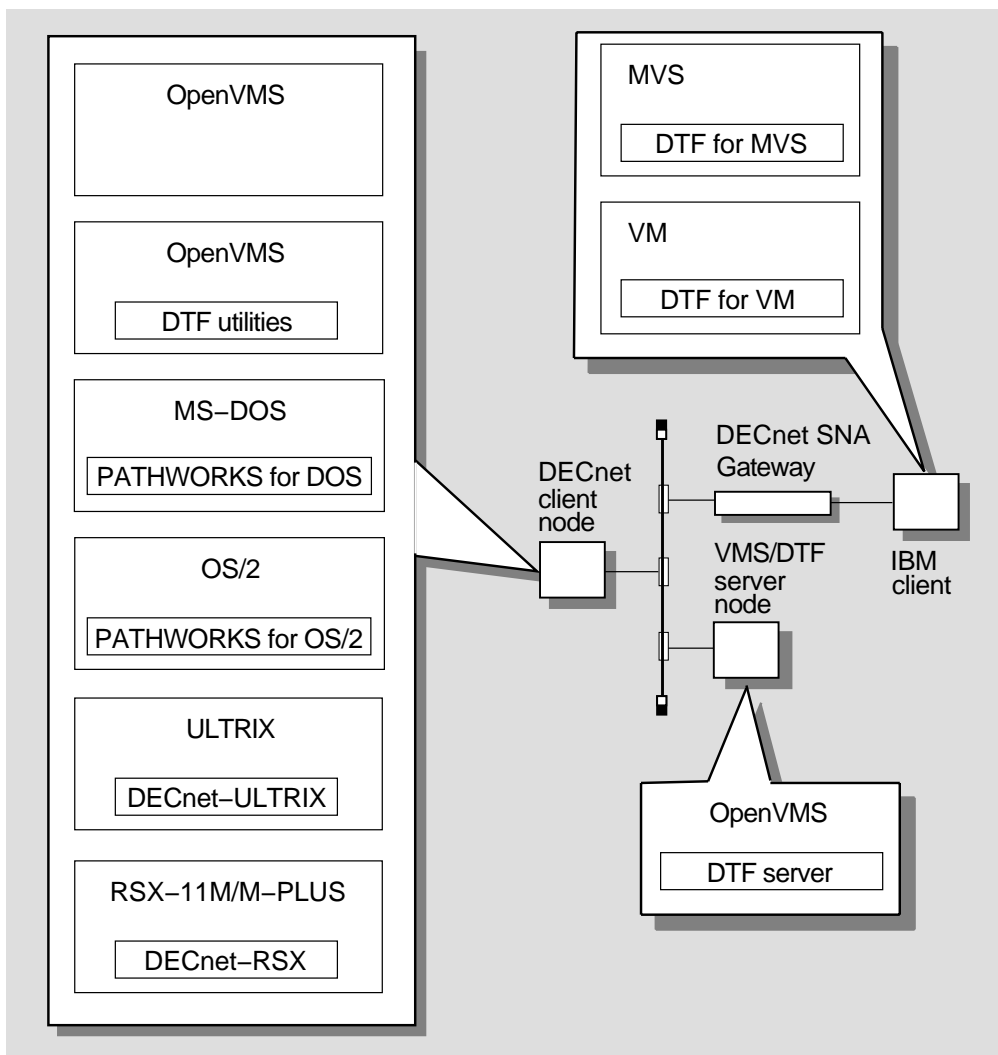
Figure 1-1 shows the relationship of the software components in a DTF environment containing a DECnet SNA Gateway. Note that the DECnet SNA Gateway can also be TCP/IP node.

1.3 What Are the Capabilities of the DTF Product?

The DTF software has the following capabilities:

- Multiple user interfaces (all clients)
- Data translation (all clients)
- Multiple file types (all clients)
- Remote job submission and post processing (all clients)
- Directory capability (OpenVMS, RSX-11M/M-PLUS, Digital Unix, ULTRIX, MS-DOS, and OS/2 clients)
- Recoverable file transfer (OpenVMS TRANSFER/DTF and IBM clients only)
- RMS programming interface (OpenVMS clients only)
- Record level access (OpenVMS clients only)

Figure 1-1 Software Components in a DTF Environment



LKG-4629-93R

1.3.1 Multiple User Interfaces (All Clients)

The DTF product provides both Digital and IBM users with a familiar user interface. This interface allows both Digital and IBM users to transfer data using an interface to which they are already accustomed.

1.3.1.1 DECnet User Interfaces

Most of the network file transfer commands work in much the same way with DTF as they do when they are used to access a file on a DECnet network. The only difference is that one of the file specification parameters is replaced with an IBM file specification. This makes accessing a file on an IBM client as easy as accessing a file on a DECnet network. Refer to Chapter 2 for more information on the IBM file specification parameters and IBM file specification qualifiers. Refer to Chapter 5 and Chapter 6 for information on using the OpenVMS client interfaces. Refer to Chapter 10 for information on using the ULTRIX and Digital UNIX client interfaces. Refer to Chapter 11 for information on using the RSX-11M/M-PLUS client interface. Refer to Chapter 12 for information on using the MS-DOS and OS/2 client interfaces.

1.3.1.2 IBM User Interfaces

DTF provides three user interfaces for MVS clients and three user interfaces for VM clients that you can use to copy files. The interfaces for MVS clients are as follows:

- Panel interfaces for TSO ISPF interactive users
- Interactive and batch TSO commands
- TSO command processor

The interfaces for VM clients are as follows:

- Panel interfaces for CMS ISPF and non-ISPF users
- Interactive and batch CMS commands
- CMS command processor

The DTF panel interface provides an easy-to-use method of transferring files. The panels prompt you for the information required to copy a file. The panel interface is recommended if you are a new DTF user. The panel interface allows you to submit a copy request to the batch subsystem. Refer to Chapter 7 for more information about the DTF panel interface.

The set of DTF file transfer commands is an alternative to the ISPF interface. These are single-line commands that you enter from either the TSO or CMS READY prompt on either MVS or VM. Refer to Chapter 8 for more information about the DTF file transfer command interface.

The final way to copy files is by using the DTF command processor. The command processor offers the most flexibility and functionality for copying files across a network. You should use the command processor if you are an experienced DTF user and you need its robust interface. Refer to Chapter 9 for more information about the DTF command processor interface.

1.3.2 Data Translation (All Clients)

The DTF product allows you to translate text between DMCS (Digital Multi-national Character Set), a superset of ASCII (American Standard Code for Information Interchange), and EBCDIC (Extended Binary Coded Decimal Interchange Code) on a record-by-record basis. Data translation is performed by default in all cases, except when a file is transferred between two MVS clients, two VM clients, or an MVS client and a VM client.

You can change the data translation default in the following ways:

- Turn data translation on or off to override the default.
- Specify a different translation table to control data translation. This table resides on the OpenVMS server node.
- Specify a VAX CDD record definition to provide field-level data translation.

For more information on selecting data translation, see the `/TRANSLATE` qualifier in Section 2.3.3.

The OpenVMS/DTF server manager can control whether data translation occurs on the OpenVMS/DTF server or on the DTF for IBM client. For more information on controlling where translation is performed, see the *Digital SNA Data Transfer Facility for OpenVMS Management* manual.

For more information about field-level data translation, refer to the *Common Data Dictionary Data Definition Language Reference* manual and Appendix F of this manual.

1.3.3 Multiple File Types (All Clients)

You can use the DTF software to transfer a variety of non-IBM and IBM file types.

1.3.3.1 Supported Non-IBM Files

On OpenVMS clients, DTF software can transfer and create all local RMS file types. On Digital UNIX, ULTRIX, RSX-11M/M-PLUS, MS-DOS, and OS/2 clients, DTF software can transfer and create the sequential file type.

1.3.3.2 Supported IBM Files

On IBM clients, the DTF software can transfer and create VSAM and non-VSAM file types.

VSAM file types supported in transfers between IBM clients and OpenVMS clients are entry sequenced data sets (ESDS), relative record data sets (RRDS), and key sequenced data sets (KSDS). IBM disk-resident files are supported and can be transferred to and from OpenVMS RMS files.

Non-VSAM file types supported in transfers between IBM clients and all supported DECnet clients have a sequential organization. This includes partitioned data set (PDS) members on MVS clients and CMS maclib members on VM clients. Both IBM disk- and tape-resident files are supported and can be transferred to and from files on the DECnet clients.

For more specific information about file types supported and any restrictions on transfer operations, see Chapter 3.

1.3.4 Remote Job Submission and Post Processing (All Clients)

DECnet clients can use the DTF product to submit jobs to IBM batch subsystems. Some DECnet clients allow you to copy and submit in one operation while some clients require two separate operations.

In addition, IBM and OpenVMS clients have the capability of submitting a batch job to perform post processing after a successful file transfer.

1.3.5 Directory Capability (OpenVMS, Digital UNIX, ULTRIX, RSX-11M/M-PLUS, MS-DOS, and OS/2 Clients)

You can use the DTF software to obtain listings of files on IBM clients. All clients supporting this directory capability allow you to specify wildcard characters that you can use to display groups of files.

1.3.6 Recoverable File Transfer (OpenVMS TRANSFER/DTF and IBM Clients Only)

The OpenVMS/DTF utilities and DTF for IBM software allow you to initiate a recoverable file transfer between a Digital OpenVMS client and an IBM MVS or VM client for non-VSAM files. DTF for IBM software also allows you to initiate non-VSAM recoverable copy operations between an IBM client and another IBM client. If a recoverable file transfer operation fails because of a system or network error, the transfer can be resumed from the last checkpoint after the system or network recovers. The checkpoint and recovery feature is a valuable time saver, especially for transferring large amounts of data.

An OpenVMS client does not support this recovery capability unless it has the TRANSFER/DTF utility installed. OpenVMS users should refer to Chapter 6 for a complete description of the TRANSFER/DTF utility. IBM users should refer to the chapter describing their preferred interface.

1.3.7 RMS Programming Interface (OpenVMS Clients Only)

You can use the standard Record Management Services (RMS) programming interface transparently with DTF to access data sets on an IBM client from an OpenVMS client. Refer to Appendix B for a detailed description of the RMS programming interface and restrictions associated with DTF.

1.3.8 DCL Record Level Access and DATATRIEVE Access (OpenVMS Clients Only)

The DTF software allows you to perform DCL-initiated record level access to VSAM and non-VSAM files. VSAM file types supported are sequential files, relative files, and indexed files (including multikey VSAM files). Refer to Chapter 5 for more information.

DTF supports the use of DATATRIEVE operations on IBM files.

1.4 What Does the DTF Product Allow Me to Do?

DTF allows you to transfer and manipulate files on Digital and IBM clients using the commands and programming calls listed in the sections that follow.

1.4.1 OpenVMS DCL Commands

- APPEND - append one file to another
- BACKUP - archive and restore a file or set of files
- CLOSE - close a file
- CONVERT - convert and merge files
- COPY - copy a file to another file
- CREATE - create a new file
- DELETE - delete an existing file
- DIFFERENCES - show the differences between two files
- DIRECTORY - show a listing of files
- EXCHANGE/NETWORK - copy a file using block- or record-mode transfers
- OPEN - open a file for record access
- READ - read a record from a file
- SEARCH - search a file for a character string
- SUBMIT/REMOTE - submit a file to a batch subsystem

- TYPE - display a file on the terminal
- WRITE - write a record to a file

1.4.2 OpenVMS TRANSFER/DTF Commands

- COPY - copy a file to another file
- SET QUEUE - set TRANSFER/DTF queue characteristics
- SHOW JOBS - show active TRANSFER/DTF requests
- SHOW QUEUE - show TRANSFER/DTF queue characteristics
- SHOW VERSION - show TRANSFER/DTF version number
- CANCEL JOB - cancel a pending TRANSFER/DTF request
- SHOW ERROR - convert error codes to error messages

1.4.3 IBM DTF Panel Interface Operations

- Send a local file to a remote system
- Receive a file from a remote system
- Resume a failed transfer request

1.4.4 IBM DTF File Transfer Commands

- DTFSEND - copy a local file to a remote file
- DTFRECV - copy a remote file to a local file
- DTFRESUM - resume a failed transfer request

1.4.5 IBM DTF Command Processor Commands

- DEFINE - define transfer parameters
- LIST - show defined transfer parameters
- TRANSFER - transfer a file using defined parameters
- RECALL - recall defined parameters
- RESUME - resume a failed transfer request
- SET - set interface characteristics
- SHOW - show interface characteristics
- CLEAR - clear transfer parameters and reset interface characteristics

1.4.6 DECnet-ULTRIX and DECnet/OSI Network File Transfer Commands

- **dcat** - display a file
- **dcp** - copy, append, or submit a file
- **dls** - display a file listing
- **drm** - delete an existing file

1.4.7 RSX-11M/M-PLUS DCL Commands

- APPEND - append one file to another
- COPY - copy a file to another file
- CREATE - create a new file
- DELETE - delete an existing file
- DIRECTORY - show a listing of files
- SUBMIT - submit a file to a batch subsystem
- TYPE - display a file on the terminal

1.4.8 DECnet-RSX NFT Switches

- (no switch) - copy a file to another file
- /AP - append one file to another
- /DE - delete an existing file
- /LI, /FU, /AT, or /BR - show a listing of files
- /SB - submit a file to a batch subsystem

1.4.9 DECnet-RSX FTS Switches

- (no switch) - copy a file to another file
- /AP - append one file to another
- /DE - delete an existing file
- /SB - submit a file to a batch subsystem

1.4.10 PATHWORKS for DOS NFT Commands

- APPEND - append a file to another file
- COPY - copy a file to another file
- DELETE - delete an existing file
- DIRECTORY - display a file listing
- SUBMIT - submit a file to a batch subsystem
- TYPE - display a file on the terminal

1.4.11 PATHWORKS for OS/2 NFT Commands

- APPEND - append a file to another file
- COPY - copy a file to another file
- DELETE - delete an existing file
- DIRECTORY - display a file listing
- SUBMIT - submit a file to a batch subsystem
- TYPE - display a file on the terminal

1.4.12 OpenVMS RMS Programming Access

- \$CLOSE
- \$CONNECT
- \$CREATE
- \$DELETE
- \$DISCONNECT
- \$DISPLAY
- \$ERASE
- \$FIND
- \$FREE
- \$GET
- \$OPEN
- \$PARSE
- \$PUT

- \$RELEASE
- \$REWIND
- \$SEARCH
- \$UPDATE

2

DTF File Specifications

This chapter covers, in detail, the DTF file specification syntax for remote file names.

On DECnet clients, the commands used to invoke DTF are the commands usually used for network file transfer. To use DTF you substitute an IBM VM or MVS file specification for one or more of the file names in any DTF-supported file transfer command.

On IBM clients, remote file specifications are entered using mechanisms available in each interface. File specifications are entered directly only when you wish to specify an option not available in the interface.

2.1 Specifying Remote IBM Files for DTF File Access

To specify a remote IBM file to DTF, use the following syntax:

For OpenVMS, RSX-11M/M-PLUS, MS-DOS, and OS/2 clients:

node" access-control"::"ibm-filename ibm-file-spec-qualifiers"

For ULTRIX and Digital UNIX clients:

node/access-control/::"ibm-filename ibm-file-spec-qualifiers"

For IBM clients:

'ibm-filename ibm-file-spec-qualifiers'

The surrounding single quotation marks are optional if no file qualifiers are specified.

where

node specifies the node on which the OpenVMS/DTF server software resides.

access-control

specifies the name of the OpenVMS/DTF server account assigned by your OpenVMS/DTF server manager. The account name is usually SNADTF. For more information about the correct value for the *access-control* parameter, contact your OpenVMS/DTF server manager.

Digital UNIX, ULTRIX, MS-DOS, and OS/2 Clients Only: If you do not enter a password in the access control string, the DECnet-ULTRIX, DECnet/OSI, or PATHWORKS for DOS software will prompt you for the password. If you enter a double slash (/) in the access control string, no password prompt appears.

VM *ibm-filename*

specifies an IBM VM CMS file name or a VSAM file name. VM VSAM file names follow the same syntax rules as MVS file names. VM CMS file names have the following format:

name file-type [file-mode][(maclib-member)]

where

- | | |
|----------------------|---|
| <i>name</i> | specifies an unqualified file name that must be 1 to 8 characters long. |
| <i>file-type</i> | specifies an IBM file type 1 to 8 characters long. If the file is a maclib, then the <i>file-type</i> must be MACLIB. |
| <i>file-mode</i> | specifies an optional two-character IBM file mode. The file mode is made up of an optional file mode letter followed by an optional file mode number. The file mode letter specifies the minidisk on which the file resides. The file mode number specifies the access mode (0-6) to be used. DTF for IBM ignores the file mode letter and defaults the file mode to 1 if the file mode parameter is not specified. |
| <i>maclib-member</i> | specifies an optional maclib member name that must be enclosed in parentheses and must be 1 to 8 characters long. Use the name stored in the maclib directory. No checking is done for MACRO, MEND, or COPY statements. If this option is specified, then the file type must be specified as MACLIB. |

For example:

```
PROG EXEC A1
```

references the file PROG of type EXEC using access mode 1. Note that the mode letter A is ignored.

MVS *ibm-filename* specifies an IBM MVS sequential file name or a VSAM file name. All MVS file names have the following syntax:

name [(*pds-member* or *gdg-number*)]

where

name specifies either a qualified or unqualified file name.

An unqualified file name must be 1 to 8 characters long and must begin with an alphabetic character.

A qualified file name consists of a series of unqualified segments separated by periods (.). Each unqualified segment must follow the syntax rules for an unqualified file name. The entire qualified file name can be up to 44 characters long. Tape-resident file names are further limited; see Section 3.3.5.

Many IBM sites require you to use your user ID as the first segment in a qualified file name. This is not required by DTF, but if this convention is required by your site, use of an invalid qualified file name results in a privilege violation error.

pds-member or *gdg-number* specifies an optional partitioned data set (PDS) member name or generation data group (GDG) number. VSAM files do not use PDS members or GDG numbers.

A PDS member name is enclosed in parentheses and must be from 1 to 8 alphabetic characters long.

A GDG number is enclosed in parentheses and must be one of the following:

- *+n* (to indicate a new version)
- 0 (to indicate the current version)
- *-n* (to indicate a previous version)

When using a GDG number with qualified file names, the qualified portion of the file name cannot be more than 35 characters long.

For example:

```
JONES.TEST.FILE(DATA)
```

references the file DATA in the partitioned data set JONES.TEST.FILE.

ibm-file-spec-qualifiers

can be appended after a file name when using DTF. Most IBM file specification qualifiers can be associated with defaults that are set up during installation of the DTF for IBM software as described in the *Digital SNA Data Transfer Facility Installation (MVS)* and *Digital SNA Data Transfer Facility Installation (VM)* manuals. These defaults are often modified, so you may need to contact an IBM system programmer for the most current IBM file specification qualifier defaults.

You can specify a group of IBM file specification qualifiers with the /FILE_DEFINITION qualifier. The /FILE_DEFINITION qualifier allows you to specify the name of a file definition contained in the OpenVMS/DTF server's file definition database. Any file specification qualifiers included with the /FILE_DEFINITION qualifier add to or override the qualifiers set up in the file definition.

Note

You must use the file definition method described in Section 2.4 to specify certain VSAM file qualifiers.

Refer to the IBM *JCL Reference* manual for a complete description of MVS file names or the *CMS Primer* manual for a complete description of VM file names.

2.2 Specifying Remote DECnet Files for IBM-Initiated DTF File Access

Remote DECnet file specifications must contain a client node or TCP/IP host name and any associated access control string, followed by a double colon (::), at the beginning of the file specification. If a file specification does not begin with a client node specification, the DTF command processor assumes that the file is a remote IBM file name.

Note

DTF does not support DECnet node to DECnet node file transfers. Use the standard DECnet file transfer utilities for this purpose.

A remote DECnet file specification has the following general format:

node" access-control"::operating-system-specific-filename

where

<i>node</i>	is the name of the DECnet client node. The client node name can be either the DECnet node name or the DECnet cluster alias. The <i>node</i> , along with any optional access control information, is always terminated by a double colon (::); the DTF command processor uses this information to differentiate between IBM and DECnet file specifications.
<i>access-control</i>	is optional information that can include a user ID and password. Although this information can be included as part of the file specification, this practice is not recommended. When the user ID and password are entered this way, they remain visible on the screen. It is better to use the PASSWORD entry fields on the DTF panel interface or use the prompting feature of the DTF command interface to enter this information.
<i>operating-system-specific-filename</i>	The file name portion of the file specification will vary based on the operating system running on the client node. The formats for the file names are described in the sections that follow.

2.2.1 Specifying Remote OpenVMS Files

Enter the remote OpenVMS file specification as shown. The *node* and *access control* portion of the file specification were previously described in Section 2.2.

node" access-control"::device:<directory>filename.type;version

where

<i>node</i>	is the 1- to 6-character name of the DECnet client node.
<i>access-control</i>	is optional information that can include a user ID and password.
<i>device</i>	is a device on the OpenVMS system, such as DRV1: or USER\$:. A device name can be up to 15 characters. The login procedure on the client node can include a default for the device specification. The device name must be terminated with a colon (:).
<i>directory</i>	is a directory name on the OpenVMS system. A directory name can be up to 39 characters; acceptable characters are alphanumeric characters, the underscore (_), and the dollar sign (\$). Subdirectories can be specified by entering a chain of up to 8 directory names, each separated by a period (.). The login procedure on the client node can include a default for the directory specification. In OpenVMS, the entire directory name is enclosed with square brackets ([]). This symbol is not available in the EBCDIC character set; therefore, angle brackets (< >) are substituted.

<i>filename</i>	is a character string that names the file. This string can be up to 39 characters; acceptable characters include alphanumeric characters, the underscore (_), and the dollar sign (\$). The file name is terminated with a period (.).
<i>type</i>	is a character string that identifies the file type. This string can be up to 39 characters; acceptable characters include alphanumeric characters, the underscore (_), and the dollar sign (\$).
<i>version</i>	is an optional file version number. The version number is a decimal number between 1 and 32767. Multiple versions can exist; the latest version of a file is the one with the highest version number. If the file specification includes a version number, that number is separated from the file type by a semicolon (;).

For example:

```
USER$656:<ROBERTSON.PROJECTS.NETWORK>DOC_PLAN.FINAL;22
```

refers to the file DOC_PLAN of type FINAL and version 22 that is stored in the subdirectory NETWORK in the subdirectory PROJECTS in directory ROBERTSON on device USER\$656.

2.2.2 Specifying Remote RSX-11M/M-PLUS Files

Enter the remote RSX-11M/M-PLUS file specification as shown. The node and access control portion of the file specification were previously described in Section 2.2.

```
node" access-control"::device:<ufd>filename.type;version
```

where

<i>node</i>	is the 1- to 6-character name of the DECnet client node.
<i>access-control</i>	is optional information that can include a user ID and password.
<i>device</i>	is a device on the RSX-11M/M-PLUS system, such as DL1: or DM:. A device name can be up to 4 characters. The first two characters (which must be alphabetic) specify the device type; the second two characters (which must form an octal number) specify the device unit. The login procedure on the client node can include a default for the device specification. The device name must be terminated with a colon (:).

<i>directory</i>	<p>is a directory name on the RSX-11M/M-PLUS system. A directory name can be up to 9 characters; acceptable characters are A - Z and 0 - 9.</p> <p>You can also use UIC-based directory names which use the syntax <<i>g</i>, <i>m</i>>. The value <i>g</i> is the group and the value <i>m</i> is the group member. Both values must be between 0 and 377 octal.</p> <p>The login procedure on the client node can include a default for the directory specification. In RSX-11M/M-PLUS, the entire directory name is enclosed with square brackets ([]). This symbol is not available in the EBCDIC character set; therefore, angle brackets (<>) can be substituted.</p>
<i>filename</i>	is a character string that names the file. This string can be up to 9 characters; acceptable characters are A - Z and 0 - 9. The file name is terminated with a period (.).
<i>type</i>	is a character string that identifies the file type. This string can be up to 9 characters; acceptable characters are A - Z and 0 - 9.
<i>version</i>	is an optional file version number. The version number is a decimal number between 1 and 32767. Multiple versions can exist; the latest version of a file is the one with the highest version number. If the file specification includes a version number, that number is separated from the file type by a semicolon (;).

For example:

```
DL2:<PHIL>DOCPLAN.FINAL;22
```

refers to the file DOCPLAN of type FINAL and version 22 that is stored in the directory PHIL on device DL2:.

2.2.3 Specifying Remote MS-DOS or OS/2 Files

Enter the remote MS-DOS or OS/2 file specification as shown. The node and optional access control portions of the file specification were previously described in Section 2.2. The file name portion of the file specification must be enclosed in double quotation marks (") as shown.

```
node"access-control": "drive:\directory\ filename.type"
```

where

<i>node</i>	is the 1- to 6-character name of the DECnet client.
<i>access-control</i>	is optional information that can include a user ID and password.
<i>drive</i>	is an optional drive name. The drive name is terminated with the colon (:) and backslash (\) characters, such as A:\ and C:\.
<i>directory</i>	is an optional path name to a directory. The path name is terminated by the backslash (\).

filename is an alphanumeric file name of up to 8 characters. The file name must be terminated by a period (.).

type is an alphabetic file extension of up to 3 characters.

For example:

```
C:\ROGERS\REPORT.TXT
```

refers to the file REPORT.TXT that is stored in the directory ROGERS on device C:.

2.2.4 Specifying Remote ULTRIX and Digital UNIX Files

Enter the remote ULTRIX or Digital UNIX file specification as shown. The node and optional access control portions of the file specification were previously described in Section 2.2. The file name portion of the file specification must be enclosed in slashes (/) as shown.

Note

ULTRIX file names are case sensitive. That is, file abc is not the same file as ABC.

```
node/access-control::"path /filename"
```

where

node is the 1- to 6-character name of the DECnet client node.

access-control is optional information that can include a user ID and password.

path is the absolute or relative path name. If you do not specify an absolute path name, the absolute path used is based on the access control information provided. An absolute path name is prefixed by a slash (/).

filename is an alphanumeric string of up to 256 characters that identifies a file. A file name cannot contain any colon (:) characters or any slashes (/) before a colon.

Example of an absolute path:

```
/usr/rogers/report.txt
```

refers to the file **report.txt** that is accessed using the path **/usr/rogers**.

Example of a relative path:

```
dnnode/jones passwd/::"bin/chess"
```

refers to the file **chess** that is accessed using the path **~jones/bin**.

2.3 IBM File Specification Qualifiers

You can use the IBM file specification qualifiers described in this section to supply attributes that are unique to IBM files. When you specify IBM file specification qualifiers with a value, you must use a colon (:) to separate the qualifier from the value. If you use an equals sign (=) as a separator instead of a colon (:), your request is rejected and the following message is displayed:

```
SYSTEM-F-NOSUCHOBJ, Network object is unknown at remote node
```

A user entering a DIRECTORY command, and incorrectly entering an equals sign in the IBM file specification, will observe the following:

```
$dir 0"snadtff"::"file.*/user=owner"  
Directory 0"snadtff"::  
"file.*/user=owner"  
Total of 1 file
```

For ULTRIX and Digital UNIX client nodes only: All qualifiers in this chapter are shown in uppercase letters. Unlike typical Digital UNIX or ULTRIX commands, ULTRIX client users may use either upper or lower case in the entire IBM file specification.

2.3.1 Qualifier Defaults

Most IBM file specification qualifiers have a default value. You select the default value by omitting the IBM file specification qualifier. The default value is always controlled by the DTF for IBM software on the remote client. In many cases, the default value can be defined when installing the DTF for IBM software. Refer to the *Digital SNA Data Transfer Facility Installation (MVS)* and *Digital SNA Data Transfer Facility Installation (VM)* manuals for more information on assigning default values at DTF for IBM installation time.

Depending on whether the default value can be changed, the default value for an IBM file specification qualifier is determined by one of the following two rules:

- **The default value of the qualifier cannot be changed:** The qualifier's default is documented in the description of the qualifier.
- **The default value of the qualifier can be changed when DTF for IBM is installed:** You should contact your IBM site for the most current value of the qualifier's default.

2.3.2 Qualifier Restrictions When Accessing Remote VM and MVS Clients

Table 2–1 lists the DTF for IBM qualifiers and the types of VM files each qualifier can be used for (CMS files, tape-resident files, and VSAM files). Each qualifier is described in greater detail in Section 2.3.3.

Table 2–2 lists the DTF for IBM qualifiers and the types of MVS files each qualifier can be used for (non-VSAM disk-resident files, tape-resident files, and VSAM files). Each qualifier is described in greater detail in Section 2.3.3.

Table 2–1 IBM File Specification Qualifiers and Their Usage for Remote VM Files

Qualifier	CMS Files	Tape-Resident Files	VSAM Files
/ALIGNMENT	ignored	ignored	ignored
/ALLOCATION	supported	ignored	supported
/BATCHID ⁷	supported	ignored	ignored
/BLOCK_SIZE ^{3,6}	ignored	supported	ignored
/CASE	supported	supported	rejected
/CATALOG	ignored	ignored	ignored
/CLASS ⁷	supported	ignored	ignored
/DEFAULT ^{1,2,3,4,5}	supported	supported	supported
/DENSITY ^{1,2,3,6}	ignored	supported	ignored
/DIRECTORY_BLOCKS	ignored	ignored	ignored
/FILE_DEFINITION ^{1,2,3,4,5}	supported	supported	supported
/HSMRECALL	ignored	ignored	ignored
/IGNORE ^{1,2,3}	supported	supported	supported
/KEY ¹	ignored	ignored	supported
/LABEL ^{1,2,3,6}	ignored	supported	ignored

¹This qualifier is used when reading IBM input files for copy, search, compare, and display operations.

²This qualifier is used when writing over existing IBM output files.

³This qualifier is used when creating a new IBM file.

⁴This qualifier is used when deleting an IBM file.

⁵This qualifier is used when listing an IBM directory.

⁶This qualifier is used only for IBM tape-resident files.

⁷This qualifier is used only for SUBMIT/REMOTE operations.

(continued on next page)

Table 2–1 (Cont.) IBM File Specification Qualifiers and Their Usage for Remote VM Files

Qualifier	CMS Files	Tape-Resident Files	VSAM Files
/MDADDRESS ^{1,2,3,4,5}	supported	ignored	supported
/MDMPASSWORD ^{1,2,3,4,5}	supported	ignored	supported
/MDRPASSWORD ^{1,2,3,4,5}	supported	ignored	supported
/MDWPASSWORD ^{1,2,3,4,5}	supported	ignored	supported
/MRS	ignored	ignored	ignored
/NULL ^{2,3}	supported	ignored	ignored
/OWNERID ^{1,2,3,4,5}	supported	ignored	supported
/PASSWORD ^{1,2,3,4,5}	supported	supported	supported
/RECORD_DEFINITION ^{1,2,3}	supported	supported	supported
/RELEASE	ignored	ignored	ignored
/RETENTION_PERIOD ³	ignored	ignored	supported
/RFA ^{1,2,3}	ignored	ignored	supported
/SECONDARY_ALLOCATION	supported	ignored	supported
/SECURITY_DATA ^{1,2,3,4,5}	supported	supported	supported
/SEQUENCE_NUMBER ^{1,2,3,6}	ignored	supported	ignored
/SINGLE ^{3,6}	ignored	supported	ignored
/SMSDCLASS	ignored	ignored	ignored
/SMSMCLASS	ignored	ignored	ignored
/SMSSCLASS	ignored	ignored	ignored
/SPANNED ³	ignored	supported	supported
/SUPERSEDE ^{2,3}	supported	ignored	rejected
/TRANSLATE ^{1,2,3}	supported	supported	supported
/UNIT ^{1,2,3,6}	ignored	supported	ignored

¹This qualifier is used when reading IBM input files for copy, search, compare, and display operations.

²This qualifier is used when writing over existing IBM output files.

³This qualifier is used when creating a new IBM file.

⁴This qualifier is used when deleting an IBM file.

⁵This qualifier is used when listing an IBM directory.

⁶This qualifier is used only for IBM tape-resident files.

(continued on next page)

Table 2–1 (Cont.) IBM File Specification Qualifiers and Their Usage for Remote VM Files

Qualifier	CMS Files	Tape-Resident Files	VSAM Files
/USERID ^{1,2,3,4,5}	supported	supported	supported
/VOLUME ^{1,2,3,6}	ignored	supported	ignored
/VSAM ^{1,2,3,4,5}	ignored	ignored	supported

¹This qualifier is used when reading IBM input files for copy, search, compare, and display operations.

²This qualifier is used when writing over existing IBM output files.

³This qualifier is used when creating a new IBM file.

⁴This qualifier is used when deleting an IBM file.

⁵This qualifier is used when listing an IBM directory.

⁶This qualifier is used only for IBM tape-resident files.

Table 2–2 IBM File Specification Qualifiers and Their Usage for Remote MVS Files

Qualifier	Non-VSAM Disk-Resident Files	Tape-Resident Files	VSAM Files
/ALIGNMENT ³	supported	ignored	ignored
/ALLOCATION	supported	ignored	supported
/BATCHID	ignored	ignored	ignored
/BLOCK_SIZE ³	supported	supported	ignored
/CASE	supported	supported	rejected
/CATALOG ³	supported	supported	ignored
/CLASS	ignored	ignored	ignored
/DEFAULT ^{1,2,3,4,5}	supported	supported	supported
/DENSITY ^{1,2,3,6}	ignored	supported	ignored
/DIRECTORY_BLOCKS ³	supported	ignored	ignored
/FILE_DEFINITION ^{1,2,3,4,5}	supported	supported	supported
/HSMRECALL ^{1,2,4,5}	supported	supported	supported
/IGNORE ^{1,2,3}	supported	supported	supported
/KEY ¹	ignored	ignored	supported
/LABEL ^{1,2,3,6}	ignored	supported	ignored
/MDADDRESS	ignored	ignored	ignored
/MDMPASSWORD	ignored	ignored	ignored
/MDRPASSWORD	ignored	ignored	ignored
/MDWPASSWORD	ignored	ignored	ignored
/MRS	supported	supported	supported
/NULL	supported	supported	ignored
/OWNERID	ignored	ignored	ignored
/PASSWORD ^{1,2,3,4,5}	supported	supported	supported

¹This qualifier is used when reading IBM input files for copy, search, compare, and display operations.

²This qualifier is used when writing over existing IBM output files.

³This qualifier is used when creating a new IBM file.

⁴This qualifier is used when deleting an IBM file.

⁵This qualifier is used when listing an IBM directory.

⁶This qualifier is used only for IBM tape-resident files.

(continued on next page)

Table 2–2 (Cont.) IBM File Specification Qualifiers and Their Usage for Remote MVS Files

Qualifier	Non-VSAM Disk-Resident Files	Tape-Resident Files	VSAM Files
/RECORD_DEFINITION ^{1,2,3}	supported	supported	supported
/RELEASE ³	supported	ignored	ignored
/RETENTION_PERIOD ³	supported	supported	supported
/RFA ^{1,2,3}	ignored	ignored	supported
/SECONDARY_ALLOCATION	supported	ignored	supported
/SECURITY_DATA ^{1,2,3,4,5}	supported	supported	supported
/SEQUENCE_NUMBER ^{1,2,3,6}	ignored	supported	ignored
/SINGLE ³	supported	supported	ignored
/SMSDCLASS ³	supported	ignored	supported
/SMSMCLASS ³	supported	ignored	supported
/SMSSCLASS ³	supported	ignored	supported
/SPANNED ³	supported	supported	supported
/SUPERSEDE ^{2,3}	supported	ignored	rejected
/TRANSLATE ^{1,2,3}	supported	supported	supported
/UNIT ^{1,2,3,4,5}	supported	supported	ignored
/USERID ^{1,2,3,4,5}	supported	supported	supported
/VOLUME ^{1,2,3,4,5}	supported	supported	supported
/VSAM ³	ignored	ignored	sequential files only

¹This qualifier is used when reading IBM input files for copy, search, compare, and display operations.

²This qualifier is used when writing over existing IBM output files.

³This qualifier is used when creating a new IBM file.

⁴This qualifier is used when deleting an IBM file.

⁵This qualifier is used when listing an IBM directory.

⁶This qualifier is used only for IBM tape-resident files.

2.3.3 Qualifier Descriptions

This section describes the IBM file specification qualifiers used when accessing files on IBM DTF clients. Qualifiers supported only when accessing files on IBM VM clients are followed by a `[VM]`. Qualifiers supported only when accessing files on IBM MVS clients are followed by a `[MVS]`. If `[VM]` or `[MVS]` is not present, the qualifier is supported for both VM and MVS file access. The qualifiers are described in alphabetical order. All qualifiers can be abbreviated as long as the abbreviation remains unique.

/ALIGNMENT:option `[MVS]`

Specifies allocating data sets on one of the following boundaries when creating a data set:

- TRACK
- CYLINDER

Usage Notes:

- This qualifier is valid only for creating a file.
- The default specifies that DTF for IBM should supply a default. The DTF for IBM default is specified at the IBM site when DTF for IBM is installed. The default is recommended for this qualifier.
- The CYLINDER option allocates the smallest number of full cylinders to hold the data set. This option can result in inefficiently used disk space.
- The TRACK option allocates the smallest number of full tracks to hold the data set. This option is generally more appropriate because it results in more efficiently used disk space.

/ALLOCATION:number-of-512-byte-blocks

Specifies or overrides the file size for the file being created. The number of blocks can be up to %x7FFFFFFF.

Usage Notes:

- This parameter should be used only for those Digital accessors (for example, DCL BACKUP) that do not send the allocation quantity to the IBM system.
- Use of this qualifier with other accessors, including TRANS/DTF, could cause the file to be overallocated or underallocated. This results in a space error (file exceeded allocation).
- Use of this qualifier allows the installation to back up files on IBM system disks and specify a file size.

/BATCHID: *name* VM

Specifies the name of the batch machine to use when submitting a batch job. The name must be 1 to 8 characters long.

Usage Note:

- If you do not specify a value, DTF for IBM supplies the default specified at the IBM site.

/BLOCK_SIZE: *blocksize*

Specifies a number from 1 to 32760 to define the block size for creating a data set.

Usage Notes:

- This qualifier is valid only when creating a file.
- By default, DTF for IBM creates a blocked file with the block size set to the default value specified by DTF for IBM.
- The DTF for IBM default is recommended for this qualifier.
- VM This qualifier is valid only when creating tape-resident files.
- The /SINGLE qualifier overrides the /BLOCK_SIZE qualifier and creates an unblocked file.
- If you specify a block size, you must use the following rules:
 1. For files with fixed-length records, you must specify a block size that is an even multiple of the record length.
 2. For files with variable-length records, you must specify a block size that is at least 4 more than the maximum record length.
 3. For files with variable spanned records, you must specify a block size that is at least 8.

/CASE

Specifies whether the IBM file specification contains lowercase characters.

Usage Notes:

- If the IBM file type is VSAM, and a lowercase character is given in the file specification, the user will get a message that the file is not found. VSAM files cannot contain lowercase characters.
- Wildcard directories that return file specifications with lowercase characters will have a /CASE qualifier returned with those file specifications.

- The DTF for IBM system installer must specify that NODEPARMS LOWERCASECREATE is a supported feature.
- If the IBM file is SMS (system manage storage) managed, then lowercase file specifications are unsupported.

Examples:

VM Using file specification with mixed upper- and lowercase characters.

```
$ DIR ARBOR"VMXADEV"::"* * 0/CASE/OWNER:IBMOWNER"
```

```
Directory ARBOR"VMXADEV"::
  "LIONS NAMES 0"
  "OFS $remind 0/CASE"
  "OFSMAIL OFSLOGf1 0/CASE"
```

Total of 3 files.

MVS Using file specification with a wildcard.

```
$ DIR 0"SNADTF"::"IBMID.*/CASE"
```

```
Directory 0"SNADTF"::
  "IBMVOL:IBMID.login/CASE"
  "IBMVOL:IBMID.AAA"
```

In both examples, the /CASE qualifier is added to the file specifications which contain upper- and lowercase characters.

/CATALOG:option **MVS**

Specifies whether the data set you create is cataloged. The possible options and their meanings are as follows:

- YES—Catalog the data set.
- NO—Do not catalog the data set.

Usage Notes:

- This qualifier is valid only for creating a file.
- The default specifies that DTF for IBM should supply a default. The DTF for IBM default is specified at the IBM site when DTF for IBM is installed. The default is recommended for this qualifier.
- The IBM site can require or prevent cataloging of all new files. If the IBM site does not have a global cataloging default specified, it may have another default level specified. In this case, the IBM site selects a default setting for requests that do not specify the /CATALOG qualifier.

- If you use tape volumes for output, you may want to use the NO option. Check your IBM site for the recommended tape option.
- For SMS-managed data sets this qualifier is ignored and the data sets are always cataloged.

/CLASS: *class-name* VM

Specifies the class name to use when submitting a batch job. The class name is a single character (A - Z or 0 - 9) and must have already been defined by the system programmer.

Usage Note:

- If you do not specify a value, DTF for IBM supplies the default specified at the IBM site.

/DEFAULT

Used to change your default entry in the OpenVMS/DTF server's proxy database (see Section 4.2). When the OpenVMS/DTF server manager has created more than one entry for you in the OpenVMS/DTF proxy database, the manager designates one entry as your default entry. This entry is used when you do not specify the /USERID qualifier. The /DEFAULT qualifier is used to designate a new default entry in the proxy database different from the original default entry.

Usage Note:

- If you do not have an entry in the server's proxy database, then this qualifier is ignored.

/DENSITY: *density*

Specifies one of the following tape densities when reading or writing a file on tape:

- 800
- 1600
- 6250

Usage Notes:

- The default is to select the highest density allowed by the device for reading or writing files.
- This qualifier is ignored with nontape devices.

/DIRECTORY_BLOCKS:n MVS

Specifies a number from 1 to 999999 to define how many 256-byte directory blocks are allocated to a partitioned data set (PDS) directory for creating the first member of a PDS.

Usage Notes:

- This qualifier is valid only for creating a PDS.
- If you do not specify a value, DTF for IBM supplies the default specified at the IBM site.
- This qualifier is ignored if a new member is added to an existing PDS.
- If the IBM data set name does not specify a PDS member name, this qualifier is ignored and a non-PDS data set is created.
- You should overallocate the value for this qualifier because the value cannot be expanded.
- The value you specify can contain 20 member names in each directory block. However, if other utilities, such as the ISPF editor, write member names to the data set, you should plan on only five member names for each directory block.

/FILE_DEFINITION:file-definition

Specifies the use of an entry in the file definition database. This entry contains a list of default attributes for file creation and file access.

Usage Notes:

- The default is not to use a file definition.
- Any additional qualifiers included with the /FILE_DEFINITION qualifier override the defaults set up in the file definition database.
- For more information about setting up file definitions and the file definition database, see Section 2.4.

/[NO]HSMRECALL MVS

The /HSMRECALL qualifier is used to indicate whether the IBM system should recall an archived file if the file is not in the system's first-level storage.

Usage Notes:

- The default is /NOHSMRECALL.
- If you use /HSMRECALL, your request stays pending until the secondary-level storage is made available. This can cause the system to hang.

- Because access to HSM files can be disabled when DTF is installed on an MVS system, you should check with the MVS system programmer if you receive indications that HSM files are not supported at the IBM client.

/IGNORE:condition

Specifies an error condition that should be suppressed during a file operation. The condition value may be set to the following:

PRECISION_ LOSS	causes DTF to ignore the error CVTPRELOS. This error occurs when DTF is unable to convert a floating point field between IBM and Digital formats because precision would be lost. See <i>Digital SNA Data Transfer Facility for OpenVMS Problem Solving and Messages</i> for additional information about this error condition.
--------------------	---

Usage Note:

- The default is to detect and report all errors.

/KEY:(position,length[,position,length,...])

The */KEY* qualifier is used only for VSAM indexed files. It lets you dynamically manipulate key of reference (KRF) numbers.

If a VSAM indexed file has multiple keys, the keys are assigned KRF values based on file name collating sequences. Refer to Appendix D for a description of VSAM indexed file structures. If the assigned KRF sequence is not satisfactory to you, then use this qualifier to assign the desired KRF sequence.

The values for this qualifier are given in position/length pairs. The first pair identifies the position and length of the primary key (key of reference number 0), the second pair identifies the first alternate index (key of reference number 1), and so forth.

Usage Notes:

- The default is to use key of reference number 0 (the primary key).
- You are altering the KRF sequence only for the life of this access. Use this qualifier only if you have an existing application that requires a different KRF sequence than the default.
- If you supply a position/length pair that does not match any key, then the access will fail with an error.
- The number of position/length pairs that you specify can be less than the actual number of keys in the file.
- You can specify a maximum of 255 key pairs in one */KEY* qualifier.

/LABEL:label

Specifies one of the following directory label formats for reading or writing data sets to tape:

- AL—ANSI Version 1 labels or ISO/ANSI/FIPS Version 3 labels
- BLP—Bypass label processing
- NL—Non-labelled
- SL—Standard label

Usage Notes:

- This qualifier is ignored with nontape devices.
- The default for this qualifier is /LABEL:SL.
- **[MVS]** Only SL or AL can be specified with an MVS tape input file.
- **[VM]** Only SL can be specified with a VM tape input file.
- **[MVS]** You can specify AL, BLP, NL, or SL with an MVS tape output file. You cannot specify BLP or NL if you are using the OpenVMS TRANSFER/DTF COPY command with the /RECOVER qualifier.
- **[VM]** You can specify BLP, NL, or SL with a VM tape output file. You cannot specify BLP or NL if you are using the OpenVMS TRANSFER/DTF COPY command with the /RECOVER qualifier.

/MDADDRESS:disk-address **[VM]**

Specifies the address of the minidisk. On VM/SP systems, the address consists of three hexadecimal digits. On VM/XA systems, the address consists of four hexadecimal digits.

Usage Note:

- If you do not specify a value, DTF for IBM supplies the default specified at the IBM site.

/MDMPASSWORD:password **[VM]**

/MDRPASSWORD:password **[VM]**

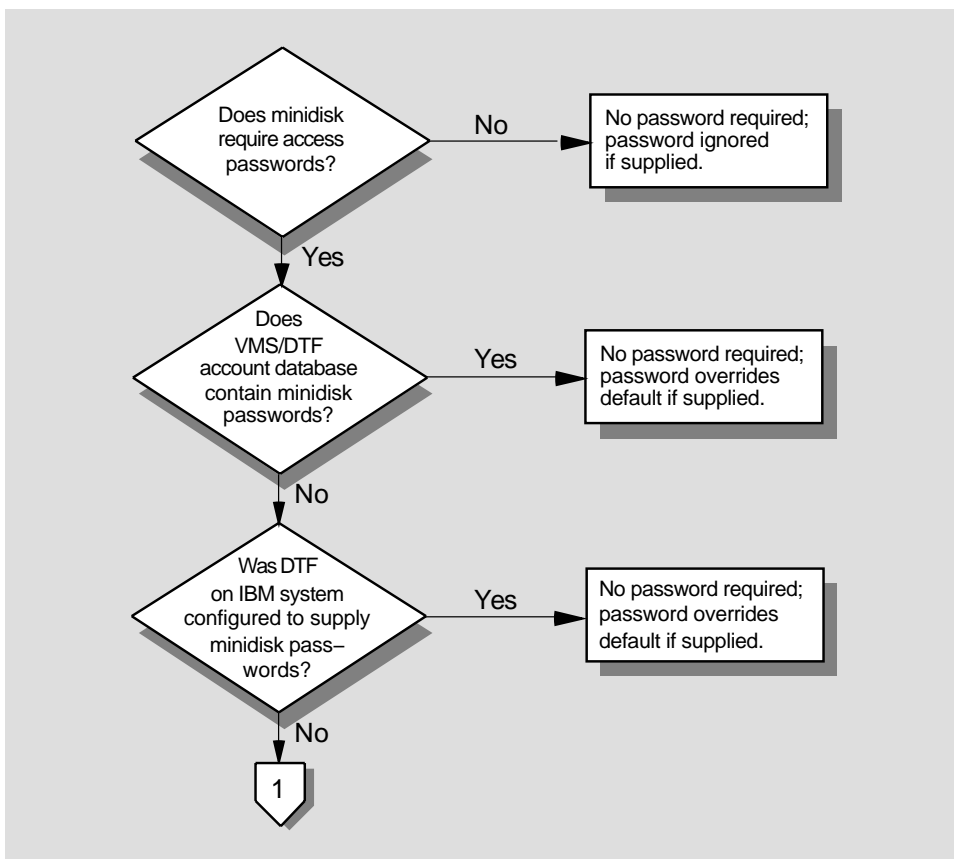
/MDWPASSWORD:password **[VM]**

Specifies the multiwrite (/MDMPASSWORD), read (/MDRPASSWORD), or write (/MDWPASSWORD) minidisk password for minidisk access. Minidisk passwords are 1 to 8 characters long.

Usage Notes:

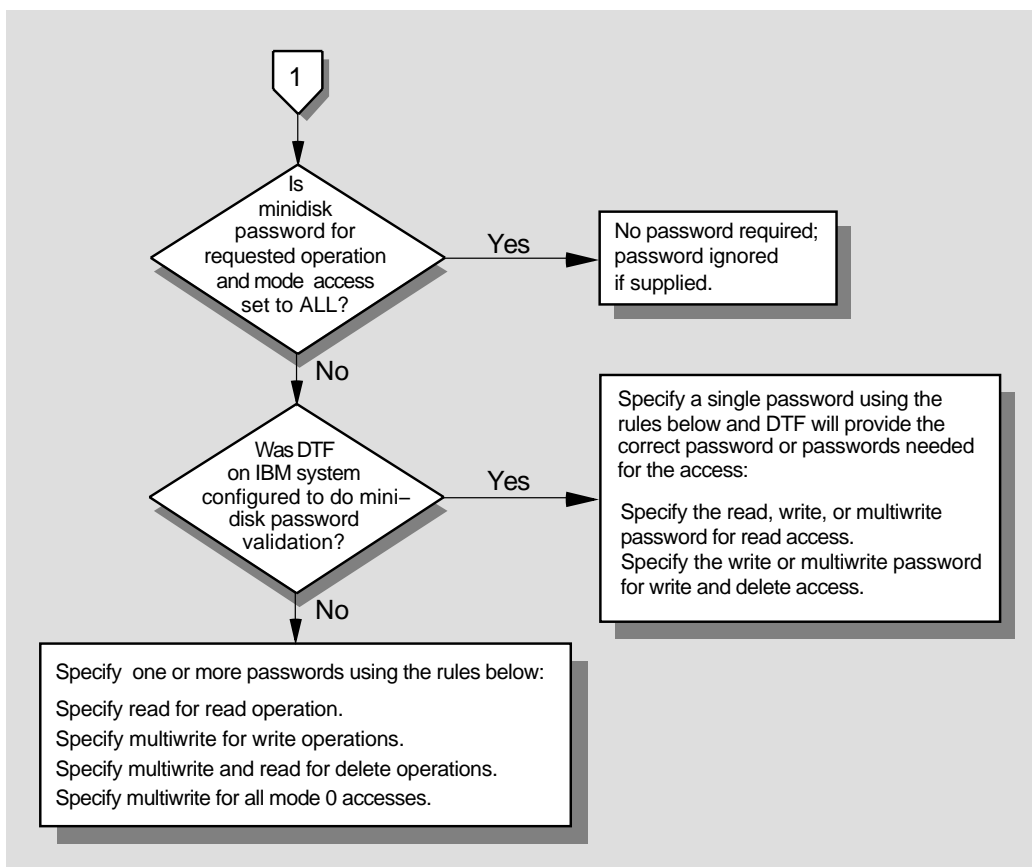
- The DTF software follows standard CMS minidisk linking rules to allow access to files. If you have a minidisk linked for read only, you can use any of the files on the minidisk as the input file to a transfer. If the minidisk is linked for write mode, you can use any files on that disk as input or output files for a transfer. If you submit a transfer to a CMS batch machine, you must provide the minidisk address and a link password in the transfer parameters so that the batch virtual machine can link to the minidisk.
If the minidisk is not linked and accessed, you must supply a minidisk address, link password, and an optional owner ID to link and access the minidisk. If you are copying a file from an unlinked and unaccessed minidisk, use the read password, unless the file is a file mode 0 file, in which case follow the WRITE LINK rules. When copying to an unlinked, unaccessed minidisk, always supply the minidisk multiwrite password.
- The preceding information and other usage notes are shown in Figure 2-1 and Figure 2-2.
- For mode 0 read access IBM users must specify the multiwrite password using the /MDMPASSWORD qualifier within the quoted file name. You cannot simply specify the multiwrite password using the panel fields or the PASSWORD options.

Figure 2-1 Using the /MDxPASSWORD Qualifier - Part 1



LKG-4416A-93R

Figure 2-2 Using the /MDxPASSWORD Qualifier - Part 2



LKG-4416B-93R

/MRS:record-length

Specifies the maximum record size (MRS) for a VB file to be created on the IBM MVS system where the record lengths exceed 255 bytes and the MRS is set to 0 for the file definition.

Usage Notes:

- This qualifier is valid only for creating a file.
- The default is to supply no maximum records size.
- **[MVS]** This qualifier is for MVS files only.
- This qualifier is not recommended unless you specifically need it.
- This value is useful for transfers in which a VB file is to be created on the IBM MVS system and record lengths exceed 255 bytes with the MRS defaults or is set to 0.
- The maximum record size defaults to 255 plus header if the file does not have an MRS value. If the file has records more than 255 bytes, then the transfer will fail with invalid record size. To correct this problem, convert the file and insert MRS in to the file using the CONVERT/FDL command.
- The maximum record size (MRS) value (or the Longest Record Length (LRL) value if MRS is not sent) is used to determine what record size the file should be created with when a OpenVMS file is sent to the IBM system. If this value is a good indication of the average record size in the file then the allocation quantity that DTF for MVS calculates will be quite good. This is because it is a close approximation to how many records are in the file based on the number of 512 byte blocks RMS has told DTF for MVS to allocate and the record size chosen for the output file. If the MRS or LRL is a bad indication of the average record size, consider using the /ALLOCATION qualifier with the correct value.

Examples:

[MVS] Specifying maximum record size for file with records longer than 255
The file X.X is variable length with the longest record equal to 350 bytes and the MRS set to 0.

```
$ COPY X.X 0"SNADTF"::"file.x/MRS:350"
```

/NULL:option

Specifies the action to take when a null record is to be written to an IBM file. You can choose from the following options:

<i>value</i>	specifies a number or character that should be stored in place of the null record. The value must be specified as one of the following:
<i>%Cchar</i>	specifies that the indicated character should be stored in place of the null record.
<i>%Dnumber</i>	specifies that the indicated decimal number should be stored in place of the null record. The number must be between -128 and +127 inclusive.
<i>%Xnumber</i>	specifies that the indicated hexadecimal number should be stored in place of the null record. The number must be between 0 and FF inclusive.
<i>%Onumber</i>	specifies that the indicated octal number should be stored in place of the null record. The number must be between 0 and 277 inclusive.
SPACE	specifies that a single space should be stored in place of the null record.
IGNORE	specifies that the null record should be ignored (that is, not copied).
NONE <input type="checkbox"/> VM	specifies no special processing will occur. An error will be returned if a null record is encountered.
NONE <input type="checkbox"/> MVS	specifies no special processing will occur. The record will be accepted as is.
REJECT	specifies that an error should be returned if a null record is encountered.

Usage Notes:

- If you do not specify a value, the default specified in the server account database will be used.
- Applies only to non-VSAM files.
- This qualifier is valid only for Digital to IBM transfers. It will be ignored when copying files from IBM to Digital.
- VM Null records cannot be written to CMS files. If a null record is encountered, an error will be returned.

/OWNERID:owner VM

Specifies the 1- to 8-character minidisk owner as specified in the VM directory.

Usage Note:

- The default is to use the IBM /USERID qualifier provided or the IBM user ID provided by a proxy mechanism. If the /USERID qualifier is not specified, then the /OWNERID qualifier is required.

/PASSWORD:password

Specifies a 1- to 8-character password for verifying access for a particular user ID through the appropriate IBM security system.

Usage Notes:

- The default is to supply no password.
- Do not specify this qualifier if the IBM site does not use a security system.
- The use of this qualifier depends on whether a proxy database is set up. See Chapter 4 for more information.

/RECORD_DEFINITION:record-definition

Specifies a 1- to 32-character full dictionary path name of a record definition in the VAX Common Data Dictionary (CDD) that controls field-level data translation during a DTF file transfer. For more information about using the CDD product, refer to the *VAX Common Data Dictionary User's Guide*. For more information about data translation in OpenVMS/DTF, see Appendix F.

Usage Notes:

- The default is not to use a record definition.
- If the /TRANSLATE qualifier is also specified, the table you specify with the /TRANSLATE qualifier determines how text fields will be translated.

/[NO]RELEASE MVS

Specifies the release of any unused tracks after creating an IBM data set. The unused space is released to the smallest integral allocation used to create the file.

Usage Notes:

- This qualifier is valid only for creating or modifying a file.
- The default qualifier is /NORELEASE.

/RETENTION_PERIOD:n

Specifies the retention period in days before which a data set cannot be deleted.

Usage Notes:

- This qualifier is valid only for creating a file.

- The default is not to specify a retention period.
- **VM** This qualifier is valid only for VSAM files.
- This qualifier is not recommended unless you specifically need it.
- If you attempt to overwrite a tape-resident file before its retention period expires, DTF for IBM waits until an operator grants permission to delete the file. On MVS systems, this operator is the MVS console; on VM systems, this operator is the Node Manager machine. You can abort the operation locally by terminating the operation using a command or control character appropriate to your local system.

/[NO]RFA (OpenVMS clients only)

Tells DTF for IBM to maintain a table of VSAM relative byte addresses (RBAs) by primary key value. RBAs are similar to the relative file addresses (RFAs) used in RMS.

Usage Notes:

- This qualifier is used only for indexed files requiring RFA access. DATATRIEVE uses RFA access.
- The default is /NORFA.
- The table of VSAM RBAs is maintained only for the life of the access.
- The amount of table space you are allowed is determined at DTF for IBM installation time. If you exceed this table space size, you will receive a status code indicating that RFA table space has been exceeded and the requested operation (for example, \$GET or \$PUT) will not be performed.
- To free your table space, the indexed file must be closed and opened. If you are a DATATRIEVE user, you must issue a FINISH command followed by a READY command to close and open a file.
- If a record is added to a PDS member, DTF rewrites the member. Therefore, RFA values for records in the member will no longer be valid. If the application that added the record to the PDS member requires valid RFA values (DATATRIEVE requires this), the RFA values need to be reestablished. In DATATRIEVE this means reestablishing the collection.

/SECONDARY_ALLOCATION: number-of-512-byte-blocks

Specifies or overrides the defined default extension quantity for the input file. The number of blocks can be up to %x7FFFFFFF.

Usage Notes:

- This qualifier should be used only if the default extension quantity defined for the input file is insufficient for future growth.
- Note that the DCL BACKUP utility will not send this value to the IBM system. If you are writing savesets to IBM disks, this qualifier may be necessary if the /ALLOCATION value has been underestimated.

/SECURITY_DATA: *data*

The /SECURITY_DATA qualifier is an IBM site-dependent qualifier. The qualifier allows you to pass from 1 to 255 characters to the IBM security system. The qualifier's use depends on the installation options chosen for DTF for IBM. Check with the IBM system programmer for more information on how to use this qualifier.

Usage Note:

- The default is to supply no security data.

/SEQUENCE_NUMBER: *n*

Specifies a sequential file number from 1 to 9999 for reading or writing files on tape.

Usage Notes:

- This qualifier is ignored with nontape devices.
- The default qualifier is /SEQUENCE_NUMBER:1.

/[NO]SINGLE

Specifies whether data is written as a separate block or multiple records are written to each block. The /SINGLE qualifier specifies an unblocked file (that is, every record is written as a separate block).

If the /NOSINGLE qualifier is specified, a blocked file is created (that is, multiple records are included in a single block).

Usage Notes:

- This qualifier is valid only for creating a file.
- The default qualifier is /NOSINGLE.
- This qualifier is valid only for tape-resident files.
- The /SINGLE qualifier is not recommended because in most cases it results in inefficiently used disk space.

- If you specify the /SINGLE qualifier, the /BLOCK_SIZE qualifier is ignored. Refer to the /BLOCK_SIZE qualifier for additional information.

/SMSDCLASS: *data-class* MVS

/SMSMCLASS: *management-class* MVS

/SMSSCLASS: *storage-class* MVS

Specifies the SMS data class (/SMSDCLASS), management class (/SMSMCLASS), or storage class (/SMSSCLASS) used during storage allocation at file creation time. SMS class names are 1 to 8 characters long.

Usage Notes:

- These qualifiers are valid only for creating a file.
- If you do not specify values for these qualifiers, DTF for IBM supplies the defaults specified at the IBM site.
- Prior to allocating storage DTF makes a call to the security exit to determine if you have access to the indicated storage class and management class. A security violation will occur if you do not have access to the classes you specified.
- IBM does not recommend that users code these parameters. Therefore, the IBM site's system programmer may have restricted the use of these parameters.
- The IBM system programmer can specify values that will override any values you may specify. Check with the IBM system programmer for more information.
- Access to SMS files can be disabled when DTF is installed on an MVS system. Check with the MVS system programmer if you receive indications that SMS files are not supported at the IBM client.

/[NO]SPANNED

Specifies whether variable-length records are spanned across a physical block or whether files with fixed-length records are assigned the standard attribute.

Usage Notes:

- This qualifier is valid only for creating a file.
- The default qualifier is /NOSPANNED.
- You cannot append a file to fixed-block standard files (DSORG=FBS) with DTF or any IBM utility.
- The /SPANNED qualifier is not recommended with either fixed- or variable-length records unless it is specifically required.

/[NO]SUPERSEDE

Specifies creation of a new output file or, if the output file already exists, overlaying the existing file. If the /SUPERSEDE qualifier is specified and the file transfer fails, the original IBM data set is unaffected.

Usage Notes:

- The default qualifier is /NOSUPERSEDE.
- The /SUPERSEDE qualifier creates a temporary file during the file transfer. When the copy operation is successful, it renames the temporary file (using a DELETE/RENAME operation).
- The /SUPERSEDE qualifier is not supported for VSAM files.
- The /SUPERSEDE qualifier is not supported for elements of an SMS-managed generation data group.

/[NO]TRANSLATE:file-spec

Specifies whether the data is to be translated. If you specify the /TRANSLATE qualifier, data is translated using a default translation table. If you specify a file specification with the /TRANSLATE qualifier, you can include your own translation table. For information about translation tables, refer to the *Digital SNA Gateway for Synchronous Transport, Digital Gateway for Channel Transport* and *OpenVMS SNA Management Guides*).

If you specify the /NOTRANSLATE qualifier, data translation does not occur.

Usage Notes:

- The default qualifier is /TRANSLATE.
- If the /RECORD_DEFINITION qualifier is also specified, the /TRANSLATE qualifier determines how data is translated in text fields.
- The translation table must be stored on the OpenVMS/DTF server node and must be accessible to all DTF server accounts. Digital recommends storing the translation table in SYSS\$LIBRARY.
- If you specify /TRANSLATE, translation is performed by either OpenVMS/DTF or by DTF for IBM. The translation location depends on information in your OpenVMS/DTF server account.
- The file name used with the /TRANSLATE qualifier must be a fully-specified file name.

/UNIT:unit-spec MVS

Specifies a unit from 1 to 8 alphanumeric characters long to classify devices for creating a data set or creating an uncataloged data set.

Usage Note:

- If you do not specify a value, DTF for IBM supplies the default specified at the IBM site.

/UNIT:unit-spec VM

Specifies one of the following unit types for tape-resident files:

- TAPE - allows the operator to select the appropriate tape unit.
- T3420 - specifies that a 3420-type open-reel tape unit be used.
- T3480 - specifies that a 3480-type cartridge tape unit be used.

Usage Notes:

- This parameter is required for all tape accesses and must not be used for any other accesses.
- An IBM site can specify additional or different unit names during DTF for IBM installation. Check with the IBM system programmer to get the unit names in use at the site you will be accessing.

/USERID:userid

Specifies a user ID of 1 to 8 alphanumeric characters to verify access through the appropriate IBM security system.

Usage Notes:

- The default is not to supply a user ID.
- Do not specify this qualifier if the IBM site does not use a security system.
- The use of this qualifier depends on whether a proxy database is set up. See Chapter 4 for more information.

/VOLUME:(vol-name[,...])

Specifies a user-defined volume serial name of 1 to 6 alphanumeric characters for creating an IBM data set on MVS and VM clients or reading an uncataloged IBM data set on MVS clients. You can include a total of 252 alphanumeric characters. The parentheses are required if more than one volume serial name is supplied.

Usage Notes:

- The default is not to supply a volume serial name.

- **[MVS]** A volume name is required for reading uncataloged data sets. If a volume name is not specified for MVS output files, the MVS operating system determines where the file will be located. A total of 42 volumes can be specified with the `/VOLUME` qualifier.
- **[VM]** A volume name is required for accessing specific tape-resident data sets. Omit the qualifier if the IBM operator is mounting a scratch tape for the requested operation. The qualifier is ignored in all other cases. Only 1 volume can be specified with the `/VOLUME` qualifier.

`/[NO]VSAM`

Specifies whether an IBM file is a VSAM file.

Usage Notes:

- **[VM]** This qualifier is required for **all** VSAM accesses.
- **[MVS]** This qualifier is valid only for creating a file and is ignored otherwise. The `/VSAM` qualifier is used to specify that a VSAM entry sequenced data set (ESDS) should be created instead of a non-VSAM sequential data set.
- The default is `/NOVSAM`.
- **[MVS]** This qualifier is supported only if it is indicated during DTF for IBM installation time that VSAM file creation is supported.
- **[MVS]** When transferring a VSAM ESDS to another MVS client, you must specify this parameter.

2.4 Setting Up DTF File Definition Files

This section discusses creating file definition files. File definition files are used in conjunction with the `/FILE_DEFINITION` qualifier to establish default values for remote IBM file creation and file access. File definition files can be used to significantly decrease the number of qualifiers that you have to enter on the command line. File definition files are a convenience option and are not required to use DTF. You may wish to skip this section and its subsections and proceed to Chapter 3 until you become more familiar with DTF.

You can use the IBM file specification qualifiers to specify file creation parameters, but this may lead to typing errors and is more time consuming. It is much easier to create a DTF file definition file containing this information. You can then specify many parameters using the `/FILE_DEFINITION` qualifier. Once created, the DTF file definition file is stored in the file definition database on the OpenVMS/DTF server.

Note

You must use this method to specify certain VSAM file parameters that cannot be specified using the IBM file specification qualifiers.

To set up a DTF file definition file, use the editor of your choice to create a sequential file that contains the appropriate specifications. The values specified in this file allow you to override the default values that are assigned by the IBM system programmer at installation time or used by VSAM when creating VSAM files.

Notes

Attributes for non-VSAM files can be specified directly with the ADD FILE_DEFINITION command using the qualifiers described in Section 2.3.3. See Appendix C for more information about this technique.

The OpenVMS/DTF server manager can associate a default file definition for each server account. Check with your OpenVMS/DTF server manager.

2.4.1 Adding an Entry to the File Definition Database

To add an entry to the file definition database, you must use the SNADTFCFG ADD FILE_DEFINITION command. OpenVMS client users need the OpenVMS/DTF utilities software installed on their node and may have to contact the OpenVMS/DTF server manager to obtain the necessary privileges to perform this operation. All other client users must contact the OpenVMS/DTF server manager to have the manager place the file definition in the file definition database for them.

The command to add the file definition file MYINDEX.FDL to the server's file definition database as the file definition INDEXED is:

```
DTFCFG> ADD FILE_DEFINITION INDEXED MYINDEX.FDL
```

Refer to Appendix C for more information about the ADD FILE_DEFINITION command.

2.4.2 Modifying an Entry in the File Definition Database

To modify an entry in the file definition database, you must use three SNADTFCFG commands. You should perform the following steps:

1. Use the `SHOW FILE_DEFINITION` command with the `/OUTPUT` qualifier to produce a copy of the file definition that you want to change.
2. Edit this copy of the file to make your changes.
3. Delete the file definition that you want modified using the `REMOVE FILE_DEFINITION` command.
4. Replace the file definition with your modified file using the `ADD FILE_DEFINITION` command.

Your commands would resemble the following:

```
DTFCFG> SHOW FILE_DEFINITION/OUTPUT=temp-file file-def-name
$ EDIT temp-file
DTFCFG> REMOVE FILE_DEFINITION file-def-name
DTFCFG> ADD FILE_DEFINITION file-def-name temp-file
```

Appendix C contains more detailed information about the commands used for modifying file definitions.

2.4.3 Displaying File Definitions from the File Definition Database

To display your file definitions, use the `SHOW FILE_DEFINITION` command. This command allows you to display the contents of a particular file definition or to list the names of file definitions.

The command to show the file definition `SPANNED` is:

```
DTFCFG> SHOW FILE_DEFINITION SPANNED
BASE
  HSMRECALL      NO
  RELEASE        NO
  SINGLE         NO
  SPANNED        YES
  TRANSLATE      YES
```

Refer to Appendix C for more detailed information on the `SHOW FILE_DEFINITION` command.

2.4.4 File Format and Syntax

The file definition format is a grouping of IBM file attributes into related sections and subsections. BASE and ALTERNATE_INDEX are the section headings. INDEX, DATA, and PATH are the subsection headings. Attributes can be defined within each section or subsection.

The syntax rules for this file are:

- Except for the BASE section, all sections are optional.
- Only the BASE section can be specified for non-VSAM files.
- Comments are preceded by an exclamation point (!).
- Blank lines may be included.
- There must be at least one space between an attribute and the value associated with it.
- Only one attribute is allowed on each line.

The format of the syntax is the following:

```
BASE
  [attribute_1      value_1]
  .
  [attribute_n      value_n]
  [INDEX]
  [attribute_1      value_1]
  .
  [attribute_n      value_n]
  [DATA]
  [attribute_1      value_1]
  .
  [attribute_n      value_n]
[ALTERNATE_INDEX]
  [attribute_1      value_1]
  .
  [attribute_n      value_n]
  [INDEX]
  [attribute_1      value_1]
  .
  [attribute_n      value_n]
  [DATA]
  [attribute_1      value_1]
  .
  [attribute_n      value_n]
  [PATH]
  [NAME            value]
.
.
[ALTERNATE_INDEX]
```



```

[attribute_1      value_1]
.
[attribute_n      value_n]
[INDEX]
    [attribute_1   value_1]
    .
    [attribute_n   value_n]
[DATA]
    [attribute_1   value_1]
    .
    [attribute_n   value_n]
[PATH]
    [NAME          value]

```

For example, the following file definition describes an MVS file:

```

BASE
  HSMRECALL          NO
  RECORD_DEFINITION  CDD$TOP.TRANSBIN
  RELEASE            NO
  SINGLE             NO
  SPANNED            NO
  SUPERSEDE         NO
  TRANSLATE          YES
  VSAM               YES

```

The following file definition describes a VM file:

```

BASE
  BATCHID            CMSBATCH
  CLASS              A
  MDADDRESS          197
  OWNERID            BIRD
  RELEASE            NO
  SINGLE             NO
  SPANNED            NO
  SUPERSEDE         NO
  TRANSLATE          YES
  VSAM               NO

```

The following example of a file definition entry defines the attributes for an indexed file.

```

BASE
  SPANNED            NO
  TRANSLATE          YES
ALTERNATE_INDEX
  NAME               DEPTA.AIX1
  PATH
    NAME             DEPTA.PATH1

```

The following example of a file definition entry specifies attributes for creating a multikey VSAM file (containing four keys). Each component of the file is located on a different volume.

```

BASE
    RETENTION_PERIOD    1000
    TRANSLATE           YES
    USERID              PAYROLL
    INDEX
        NAME            PAYROLL.MAIN.BASE.IDX
        VOLUME          TSOVL1
    DATA
        NAME            PAYROLL.MAIN.BASE.DAT
        VOLUME          TSOVL2
ALTERNATE_INDEX
    NAME                PAYROLL.MAIN.AIX1
    INDEX
        NAME            PAYROLL.MAIN.AIX1.IDX
        VOLUME          TSOVL3
    DATA
        NAME            PAYROLL.MAIN.AIX1.DAT
        VOLUME          TSOVL4
    PATH
        NAME            PAYROLL.NAME
ALTERNATE_INDEX
    NAME                PAYROLL.MAIN.AIX2
    INDEX
        NAME            PAYROLL.MAIN.AIX2.IDX
        VOLUME          TSOVL5
    DATA
        NAME            PAYROLL.MAIN.AIX2.DAT
        VOLUME          TSOVL6
    PATH
        NAME            PAYROLL.EMPNO
ALTERNATE_INDEX
    NAME                PAYROLL.MAIN.AIX3
    INDEX
        NAME            PAYROLL.MAIN.AIX3.IDX
        VOLUME          TSOVL7
    DATA
        NAME            PAYROLL.MAIN.AIX3.DAT
        VOLUME          TSOVL8
    PATH
        NAME            PAYROLL.JOBCODE

```

The following DTF file definition entry might be used to define attributes when creating a VSAM sequential file.

```
BASE
  RECORD_DEFINITION  CDD.SALES.MONTHLY
  VSAM               YES
```

2.4.5 BASE and ALTERNATE_INDEX Sections

The BASE section is used to specify the attributes for creating non-VSAM files and VSAM clusters. You must specify the BASE section.

BASE specifies the attributes of:

- non-VSAM files (including CMS minidisk)
- VSAM sequential files
- VSAM relative record files
- the VSAM base cluster for VSAM indexed files

The ALTERNATE_INDEX section is used to specify the attributes for creating the alternate index clusters for VSAM multikey indexed files.

ALTERNATE_INDEX is used by IBM to create additional keys of reference for an indexed file. For more information about VSAM file structure, refer to Appendix D.

2.4.6 INDEX, DATA, and PATH Subsections

The INDEX and DATA subsections are components of the BASE and ALTERNATE_INDEX sections. The PATH subsection is a component of the ALTERNATE_INDEX section.

INDEX

VSAM indexed files contain data records and an index that lets you access these records by way of the key field. The INDEX subsection allows you to assign particular VSAM attributes to the index of a base or alternate index cluster. The INDEX subsection can be specified only when defining a VSAM indexed file.

DATA

The DATA subsection describes the attributes of the data records in the VSAM file. The DATA subsection can be specified for all VSAM file types.

PATH

PATH associates the alternate index with the base cluster. PATH contains information about the key of reference assigned to a VSAM file. Refer to Appendix D for more information about VSAM files.

2.4.7 File Attributes

All IBM file specification qualifiers, with the exception of the password qualifiers, can be used as file attributes in the file definition BASE section subject to the same restrictions described in Section 2.3.

Note

Attributes in the file definition file do not begin with a slash (/) and do not use the colon (:) used in qualifiers. Qualifiers with the /NO option use YES and NO as argument values in the file definition file.

For VSAM files, Table 2–3 lists the file attributes used for MVS VSAM files and shows in which sections and subsections the attributes can be used. Table 2–4 lists the file attributes used for VM VSAM files and shows in which sections and subsections the attributes can be used. Both tables contain six VSAM file attributes that can only be defined using a file definition file. These attributes are described following the tables. For a description of the other attributes, see the description of the corresponding qualifier in Section 2.3.3.

Table 2–3 MVS File Attributes and Their Usage in the BASE and ALTERNATE_INDEX Sections

File Attribute	VSAM BASE Section	ALTERNATE_INDEX Section	VSAM INDEX and DATA Subsections	VSAM PATH Subsection
BUFFERSPACE	supported	supported	not allowed	not allowed
COMPONENT_ALLOCATION	supported	supported	supported ¹	not allowed
HSMRECALL	supported	supported	not allowed	not allowed
IGNORE	supported	supported	not allowed	not allowed
MRS	supported	not allowed	not allowed	not allowed
NAME	not allowed	supported	supported ¹	supported
RECORD_DEFINITION	supported	not allowed	not allowed	not allowed
REPLICATE	supported	supported	supported ¹	not allowed
RETENTION_PERIOD	supported	not allowed	not allowed	not allowed
REUSE	supported	supported	supported ¹	not allowed

¹If specified in a subsection, overrides the value specified in the section.

(continued on next page)

Table 2–3 (Cont.) MVS File Attributes and Their Usage in the BASE and ALTERNATE_INDEX Sections

File Attribute	VSAM BASE Section	ALTERNATE_INDEX Section	VSAM INDEX and DATA Subsections	VSAM PATH Subsection
RFA	supported	not allowed	not allowed	not allowed
SECURITY_DATA	supported	not allowed	not allowed	not allowed
SHARE_OPTIONS	supported	supported	supported ¹	not allowed
SMSDCCLASS	supported	supported	not allowed	not allowed
SMSMCLASS	supported	supported	not allowed	not allowed
SMSSCLASS	supported	supported	not allowed	not allowed
SPANNED	supported	not allowed	not allowed	not allowed
TRANSLATE	supported	not allowed	not allowed	not allowed
USERID	supported	not allowed	not allowed	not allowed
VOLUME	supported	supported	supported ¹	not allowed
VSAM	supported	not allowed	not allowed	not allowed

¹If specified in a subsection, overrides the value specified in the section.

Table 2–4 VM File Attributes and Their Usage in the BASE and ALTERNATE_INDEX Sections

File Attribute	VSAM BASE Section	ALTERNATE_INDEX Section	VSAM INDEX and DATA Subsections	VSAM PATH Subsection
BATCHID	supported	not allowed	not allowed	not allowed
BUFFERSPACE	supported	supported	not allowed	not allowed
CLASS	supported	not allowed	not allowed	not allowed
COMPONENT_ALLOCATION	supported	supported	supported ¹	not allowed
IGNORE	supported	supported	not allowed	not allowed
MDADDRESS	supported	ignored	not allowed	not allowed
NAME	not allowed	supported	supported ¹	supported
OWNERID	supported	not allowed	not allowed	not allowed

¹If specified in a subsection, overrides the value specified in the section.

(continued on next page)

Table 2–4 (Cont.) VM File Attributes and Their Usage in the BASE and ALTERNATE_ INDEX Sections

File Attribute	VSAM BASE Section	ALTERNATE_ INDEX Section	VSAM INDEX and DATA Subsections	VSAM PATH Subsection
RECORD_DEFINITION	supported	not allowed	not allowed	not allowed
REPLICATE	supported	supported	supported ¹	not allowed
RETENTION_PERIOD	supported	not allowed	not allowed	not allowed
REUSE	supported	supported	supported ¹	not allowed
RFA	supported	not allowed	not allowed	not allowed
SECURITY_DATA	supported	not allowed	not allowed	not allowed
SHARE_OPTIONS	supported	supported	supported ¹	not allowed
TRANSLATE	supported	not allowed	not allowed	not allowed
USERID	supported	not allowed	not allowed	not allowed
VOLUME	supported	supported	supported ¹	not allowed
VSAM	supported	not allowed	not allowed	not allowed

¹If specified in a subsection, overrides the value specified in the section.

Attributes Unique to File Definition Files

BUFFERSPACE [n]

Specifies the minimum space required for buffers as a decimal number that is less than 16776704 (decimal).

Usage Notes:

- If you do not specify BUFFERSPACE, VSAM determines the size by using default values for the control intervals.
- The default value is recommended for this attribute.

COMPONENT_ALLOCATION *option*

Specifies how the space for a component is allocated:

- UNIQUE—VSAM data space is created and assigned exclusively to each component
- SUBALLOCATION—space from one of the VSAM data spaces on the volume is assigned to the component

Usage Notes:

- For VSAM files, SUBALLOCATION is the default. The default value is recommended.

ICF Users

Note that UNIQUE is the default for ICF catalogs. SUBALLOCATION is ignored.

- You cannot specify the REUSE attribute if you specify UNIQUE.
- If you do not specify UNIQUE as the value, VSAM data space must already be available on the volume that will be used by the component.

NAME *entry-name*

Specifies the name of a component so that you can assign each component a name.

Usage Notes:

- If you do not specify a name, a name will be generated.
- Names must conform to the MVS file name standards described in Section 2.1.

REPLICATE *option*

Specifies whether each index record in a component is repeatedly written on a direct access device track in the following manner:

- YES—Each index record is written as many times as possible.
- NO—Each index record is written only once.

Usage Notes:

- The default value for this attribute is NO.
- This attribute interacts with the default for VSAM's IMBED attribute to determine the physical attributes of the index record.
- This attribute relates only to a key-sequenced data set (KSDS).

REUSE *option*

Specifies whether a component can be used repeatedly to load new records. If REUSE is specified, then the file is treated as if it were empty each time the file is opened with write access.

Usage Note:

- If you specify YES, you cannot specify UNIQUE as the value for the COMPONENT_ALLOCATION attribute.

SHARE_OPTIONS *option-1* [*option-2*]

Specifies 1 or 2 single-digit numbers that control how a component can be shared among users:

- The value *option-1* specifies the kind of sharing allowed among users across multiple regions of a single CPU:
 - 1** specifies that any number of users can have read access or that a single user can have read and write access to a data set. This is the default setting for VSAM files.
 - 2** specifies that any number of users can have read access and that a single user can have read and write access to a data set.
 - 3** specifies that any number of users can have read and write access to a data set.
 - 4** specifies that any number of users can have read and write access to a data set and that buffers used for direct processing are refreshed for each request.
- The optional value *option-2* specifies the amount of sharing allowed among users across systems:
 - 3** specifies that any number of users can have read and write access to a data set. This is the default setting for VSAM files.
 - 4** specifies that any number of users can have read and write access to a data set and that buffers used for direct processing are refreshed for each request.

Usage Notes:

- For data integrity purposes, you should make sure that the values supplied for this attribute are the same for the DATA and INDEX subsections.
- The IBM system programmer can set a default for SHARE_OPTIONS at installation time. This value will be used at creation time.

DTF-Supported File Types

This chapter describes the file types supported by DTF and restrictions on the types of file transfer operations you can perform.

3.1 Supported Input File Types

This section describes considerations that you should be aware of before you use the interfaces to define input files to be transferred. These considerations include the following areas:

- File organizations
- Record formats
- Record lengths
- Record attributes

3.1.1 Supported File Organizations

DTF supports the following MVS file organizations as input:

- **VSAM**—This group of files includes Key Sequenced Data Set (KSDS), Relative Record Data Set (RRDS), and Entry Sequenced Data Set (ESDS) file organizations. There are restrictions on VSAM file transfers. Section 3.3.3 explains these restrictions.
- **Physical Sequential**—This group of files includes sequential files, members of a partitioned data set (PDS), generation data group (GDG) files, and tape-resident files.

DTF supports the following VM file organizations as input:

- **VSAM**—This group of files includes Key Sequenced Data Set (KSDS), Relative Record Data Set (RRDS), and Entry Sequenced Data Set (ESDS) file organizations. There are restrictions on VSAM file transfers. Section 3.3.3 explains these restrictions.

- **Physical Sequential**—This group of files includes CMS files, MACLIB members, and tape-resident files.

DTF supports the following OpenVMS file types as input:

- **Sequential**—No restrictions.
- **Relative**—There are some restrictions for this file organization.
- **Indexed**—There are some restrictions for this file organization.

DTF supports only sequential files as input from RSX-11M/M-PLUS systems.

The Digital UNIX, ULTRIX, MS-DOS, and OS/2 operating systems can support only files that have a sequential organization. DTF supports sequential files from these systems as input.

3.1.2 Supported Record Formats

DTF supports MVS and VM input files with fixed, variable, or variable spanned record formats.

DTF supports OpenVMS input files that have the following record formats:

- **Fixed**—No restrictions.

Note

OpenVMS fixed format records are converted to variable format records when transferred to VSAM sequential files.

- **Variable**—No restrictions.
- **Variable with fixed control**—There are some restrictions. Section 3.2.4 describes restrictions for output file record attributes.
- **Stream**—DTF accepts input files with the OpenVMS stream, stream-LF, or stream-CR record formats. However, the output file will be created with variable format records with the carriage control attribute.

DTF supports RSX files that have the following record formats:

- **Fixed**—No restrictions.

Note

RSX fixed format records are converted to variable format records when transferred to VSAM sequential files.

- **Variable**—No restrictions.

DTF supports Digital UNIX, ULTRIX, MS-DOS, and OS/2 input files in stream record format only; this is the only format allowed by these operating systems. Although DTF supports stream record formats as input, the stream record formats will be converted into variable format records and assigned the carriage control attribute.

3.1.3 Supported Record Lengths

The input file record length determines the output file record length. DTF supports input files with a record length of up to 32760 bytes. On OpenVMS systems, you should set the `RMS_NETWORK_BLOCK_COUNT` to the record length plus 5 divided by 512 rounded to the next highest integer. For example, to transfer 512-byte records you should set `RMS_NETWORK_BLOCK_COUNT` to 2:

```
RMS_NETWORK_BLOCK_COUNT = INTEGER((512 + 5)/512) = 2
```

3.1.4 Supported Record Attributes

DTF supports the following combinations of record attributes for MVS input files and VM tape-resident input files:

- Any combination of the supported record formats with or without blocking, including fixed blocked standard (FBS) and variable blocked spanned (VBS).
- Any combination of the supported record format and blocking used with ANSI printer control. Section 3.3.4 explains ANSI printer control restrictions.

DTF supports OpenVMS input files with the following record attributes:

- Records with the FORTRAN printer control. Section 3.3.4 explains FORTRAN printer control restrictions.
- Records with implied carriage return. The access methods used by DTF for MVS will not retain the record attribute. See Section 3.2.4 for additional information about the record attributes of the output file.
- Records with no record attributes if the record format is fixed.
- Records with printer control if the record format is variable with fixed control (VFC). See Section 3.2 for additional information about printer control.

DTF does not support any RSX record attributes.

Digital UNIX, ULTRIX, MS-DOS, OS/2, and VM disk-resident files do not have record attributes.

3.2 Supported Output File Types

This section describes considerations that you should be aware of before you use the interfaces to define output files as targets of DTF transfers. These considerations include the following areas:

- File organizations
- Record formats
- Record lengths
- Record attributes

If a transfer operation results in the creation of an output file, the attributes of the output file are assigned using the attributes of the input file.

3.2.1 Supported File Organizations

DTF supports the following MVS file organizations as output:

- **VSAM**—This includes KSDS, RRDS and ESDS organizations. Section 3.3.3 describes the VSAM restrictions.

A VSAM KSDS will be created if the input file was an RMS indexed file or a VSAM KSDS. Section D.3 explains the VSAM cluster attribute defaults.

A VSAM RRDS will be created if the input file was an RMS relative file or a VSAM RRDS. Section D.3 explains the VSAM cluster attribute defaults.

A VSAM ESDS will be created if the input file was an RMS sequential file or an IBM sequential file and the /VSAM qualifier is added to the output MVS file name. Section D.3 explains the VSAM cluster attribute defaults.

- **Physical sequential**—This includes sequential files, members of a partitioned data set, generation data group files, and tape-resident data sets.

If the PDS already exists, the member will be added; otherwise a new PDS file will be created.

If the input file is either a sequential file or a VSAM ESDS file and the /VSAM qualifier is not specified, a physical sequential output file will be created.

DTF supports the following VM file organizations as output:

- **VSAM**—This includes KSDS, RRDS and ESDS organizations. Section 3.3.3 describes the VSAM restrictions.

A VSAM KSDS will be created if the input file was an RMS indexed file or a VSAM KSDS. Section D.3 explains the VSAM cluster attribute defaults.

A VSAM RRDS will be created if the input file was an RMS relative file or a VSAM RRDS. Section D.3 explains the VSAM cluster attribute defaults.

A VSAM ESDS will be created if the input file was an RMS sequential file or an IBM sequential file and the /VSAM qualifier is added to the output VM file name. Section D.3 explains the VSAM cluster attribute defaults.

- **Physical sequential**—This includes CMS files, MACLIB members, and tape-resident files.

If the input file is either a sequential file or a VSAM ESDS file and the /VSAM qualifier is not specified, a physical sequential output file will be created.

DTF supports the following OpenVMS file types as output:

- **Sequential**—This file format will be created if the input file has either an IBM sequential or VSAM ESDS organization.
- **Relative**—This file format will be created if the input file has a VSAM RRDS file organization. This file organization is not supported in recoverable mode.
- **Indexed**—This file format will be created if the input file has a VSAM KSDS file organization. This file organization is not supported in recoverable mode.

DTF supports only RSX-11M/M-PLUS sequential files as output.

The Digital UNIX, ULTRIX, OS/2, and MS-DOS operating systems can create only files that have a sequential organization. DTF supports sequential files as output on these systems.

3.2.2 Supported Record Formats

An MVS or VM file created during a transfer will have the record format of the input file. DTF supports the following record formats for output:

- **Fixed**—No restrictions.

Note

OpenVMS fixed format records are converted to variable format records when transferred to VSAM sequential files.

- **Variable**—A variable length record file will be created if the input OpenVMS file has a record format of V or VFC or if the file is transferred from an Digital UNIX, ULTRIX, MS-DOS, or OS/2 system. Any fixed printer control characters in a record will not be written to the file.
- **Variable spanned**—A variable spanned record format will be created if the output IBM file name is specified with the /SPANNED qualifier. Section 2.3.3 explains how to use the /SPANNED qualifier.

OpenVMS output files will take the record formats from the IBM input files. DTF supports the following record formats on output:

- Fixed
- Variable

RSX-11M/M-PLUS output files will take the record formats from the IBM input files. DTF supports the following record formats on output:

- Fixed
- Variable

The Digital UNIX, ULTRIX, OS/2, and MS-DOS operating systems can support only stream format records. On these systems DTF ignores the format of the input record and creates an output file with stream format records.

3.2.3 Supported Record Lengths

The output file record length is taken from the input file. DTF supports output files with a record length of up to 32760 bytes.

[MVS] MVS output files with variable record lengths will have record lengths 4 bytes greater than that of OpenVMS input files. This difference is due to the MVS access method (BSAM/QSAM) requirements. Conversely, output files on DECnet nodes running OpenVMS, RSX-11M/M-PLUS, MS-DOS, OS/2, Digital UNIX, or ULTRIX will have record lengths 4 bytes less than that of MVS input files.

[MVS] When an output MVS file is created, a record length may have defaults set by DTF for MVS. This is the case if the input OpenVMS file does not have a maximum record size (MRS). Quite often this is the default for OpenVMS, RSX-11M/M-PLUS, MS-DOS, OS/2, Digital UNIX, and ULTRIX files with stream or variable format records. The MVS access methods (BSAM/QSAM /VSAM) require a maximum record length, so DTF for MVS must provide a default for this when it creates an output file with variable length records. DTF for MVS does this by following these steps:

1. Round the longest record in file (LRL) attribute up to the next highest multiple of 256.
2. Subtract 1.
3. Add the 4 byte requirement of BSAM/QSAM, if pertinent.

3.2.4 Supported Record Attributes

MVS output files and VM tape-resident output files will be created with the record attribute of the input file. DTF supports the following combinations of record attributes for MVS output files:

- Any combination of the supported record formats with or without blocking, including fixed blocked standard (FBS) and variable blocked spanned (VBS). See the /BLOCK_SIZE and /SINGLE qualifiers in Section 2.3.3 for their effect on blocking.
- Any combination of the supported record format and blocking used with ANSI printer control. The input OpenVMS file must have the FORTRAN printer control character. Section 3.3.4 explains FORTRAN printer control restrictions.

DTF supports OpenVMS output files with the following record attributes:

- Records with the FORTRAN printer control. The input MVS file must have the ANSI printer control character. Section 3.3.4 explains ANSI printer control restrictions.
- All OpenVMS files are created with the implied carriage control attribute, unless the ANSI printer control attribute is assigned instead.

DTF does not support any RSX record attributes for output files.

Digital UNIX, ULTRIX, MS-DOS, OS/2, and VM disk-resident files have no record attributes.

3.3 File Transfer Considerations

This section describes various additional considerations that you should be aware of when transferring files using DTF.

3.3.1 Transferring Files Between IBM Clients

Files transferred between IBM clients always pass through an OpenVMS server node; the IBM clients are not directly connected. Be aware of the following when transferring files between two IBM clients:

- IBM output files will use the default operating system attributes.
- **[MVS]** Copy only one partitioned data set (PDS) member at a time.
- No ISPF-style statistics will be kept for the output file.
- VSAM cluster component names will be generated for the output file.

3.3.2 Transferring Files Between an IBM System and an Digital UNIX, ULTRIX, RSX-11M/M-PLUS, MS-DOS, or OS/2 Client

Be aware of the following when transferring files between an IBM system (MVS or VM) and an Digital UNIX, ULTRIX, RSX-11M/M-PLUS, MS-DOS, or OS/2 client:

- Do not copy VSAM Keyed Sequence Data Set (KSDS) or Relative Record Data Set (RRDS) files between an IBM system and one of these DECnet clients. These DECnet clients have operating systems that do not support keyed or relative file organizations.
- If a file is copied from one of these DECnet clients to an IBM system, it will be stored as a variable format record.
- If an IBM file is copied to one of these DECnet clients, it will be stored as stream format (variable format in RSX-11M/M-PLUS). The IBM operating systems do not support stream format records; these DECnet clients have operating systems that support stream format records only.

3.3.3 VSAM File Transfer Restrictions

The following restrictions apply to VSAM file transfers:

- If an OpenVMS TRANSFER/DTF transfer request involves VSAM files, the transfer must execute in nonrecoverable mode only.
- If a transfer request involves VSAM RRDS and ESDS files, you can not overlay an existing file.
- VSAM RRDS or KSDS files cannot be copied to client nodes running MS-DOS, OS/2, RSX-11M/M-PLUS, Digital UNIX, or ULTRIX.
- **[VM]** DTF for VM supports only one VSAM access at a time. If a request arrives that requires access to a DOS disk and a VSAM access is already in progress, then "VSAM is already active" is returned.

3.3.4 Printer File Transfer Restrictions

The DTF product cannot transfer IBM files in machine print format. However, DTF can be used to transfer both IBM FORTRAN ANSI print format files and OpenVMS RMS FORTRAN carriage control format files. The following restrictions apply to IBM ANSI and OpenVMS FORTRAN printer control characters:

- IBM ANSI printer control characters are not fully supported by the OpenVMS FORTRAN printer control statements.
- The IBM ANSI printer triple line feed character (-) is not recognized by OpenVMS FORTRAN printer control; it is interpreted as a single line feed.

3.3.5 Transferring IBM Tape-Resident Files

DTF does not support IBM input tapes that have a label type of NL. In addition, DTF for VM does not support tapes that have a label type of AL.

OpenVMS utilities that use the LIB\$FILESCAN system service cannot access IBM tape-resident files. This service causes an extra open and close sequence before accessing the file for the requested operation. Utilities that use LIB\$FILESCAN include the COPY, TYPE, and DELETE commands. Therefore, you must use the TRANSFER/DTF utility to access IBM tape-resident input files.

The following restrictions apply if you use the TRANSFER/DTF utility's /RECOVER qualifier (or the RECOVERY option in the IBM interfaces) when reading and writing IBM tape-resident files:

- The label type must be SL or AL (AL not supported by DTF for VM).
- The MVS operating system will call for the tape to be mounted several times during the transfer. To avoid having to remount the tape, use the PUBLIC attribute when the tape is first mounted. The VM operating system will call for the tape to be mounted once at the beginning of the transfer.

DTF for MVS supports tapes that have a label type of AL for both input and output. DTF assumes that files on these tapes are in ASCII format; therefore, do not specify field-level data translation.

Data set names for IBM tape-resident files are effectively limited to 17 characters. DTF interfaces allow longer data set names; however, MVS and VM limit data set names on tapes to 17 characters. If you specify more than 17 characters when accessing a tape-resident file, the name is truncated on the LEFT, resulting in the 17 RIGHTMOST characters being used as the file name.

Be sure to specify the /UNIT qualifier when accessing tapes. If you fail to specify the /UNIT qualifier the default unit is used. This is usually set to point to a disk device. This can cause problems when you work with unmounted tapes. DTF for MVS will issue a mount requested message on the MVS console indicating a request to mount a disk not a tape. DTF for VM will send a similar message to the virtual machine defined during DTF for VM installation (as indicated by the TAPEOPER keyword).

3.3.6 Transferring Files with Horizontal Tab Characters

If an OpenVMS file with DMCS horizontal tab characters is transferred to an IBM file, the tab characters are translated to EBCDIC tab characters. Most IBM utilities do not recognize the horizontal tab character; therefore, the file will appear to contain no horizontal tabs. When the file is transferred back to the OpenVMS system, the EBCDIC tab characters will be translated to DMCS tab characters and the tabs will be recognized by many OpenVMS utilities.

If you wish to have the file appear the same on both systems, you should run the OpenVMS file through a utility that converts horizontal tab characters to the required number of space characters. If you do not mind losing horizontal tab information, you can design a translation table that translates DMCS horizontal tabs to EBCDIC space characters.

3.3.7 Keyed Access to Sequential Files

RMS allows application programs to directly access records in sequential files by using the record's relative file address (RFA) or by using the record's key (or relative record number).

DTF supports direct access to IBM sequential files using RFA access.

DTF simulates keyed access to IBM sequential files. The IBM operating systems allow blocks that are not filled to capacity; therefore, DTF cannot simply use the key value to compute the record's address. To simulate keyed access, DTF reads from the beginning of the file until it finds the correct record. This simulation results in greater performance degradation as the size of the file increases or as the requested record nears the end of the file.

4

Data Security

This chapter covers the data security measures provided by the operating systems that are supported by DTF software. Data security can be described in terms of data security for files accessed on the local system and data security for files accessed on a remote system. Data security is further enhanced through proxy methods. What proxy is and the benefits of proxy are covered in this chapter.

4.1 Local Data Security

The DTF software uses the rules established by the local security system to allow access to files.

- On MVS clients, the DTF software internally verifies the right to access a file. This is done using calls to the security system by a security exit that can be customized during installation.
- On VM clients, the DTF software follows standard CMS minidisk linking rules to allow access to files. If the minidisk is not linked and accessed, you must supply a minidisk address, link password, and an optional owner ID to link and access the minidisk.
- On OpenVMS and RSX-11M/M-PLUS clients, your file access is determined by the access privileges associated with the account you use to log in.
- On Digital UNIX clients, ULTRIX clients, MS-DOS clients, and OS/2 clients, your file access is determined in a manner consistent with DECnet file access.

4.2 Remote Data Security Without Proxy

The DTF software sends the user ID and password you type in to the remote system to allow the remote system to check if access is allowed. The rules for the user ID and password depend on the remote operating system.

Table 4-1 shows the user ID and password syntax rules for each supported operating system.

Table 4-1 User ID and Password Syntax Rules

Operating System	Length of User ID	Length of Password	Case Sensitivity
OpenVMS	1-12 characters	1-39 characters	Not case sensitive
MS-DOS and OS/2	1-40 characters	1-10 characters	Not case sensitive
Digital UNIX and ULTRIX	1-8 characters	1-8 characters	Case sensitive
RSX-11M/M-PLUS	1-14 characters	1-39 characters	Not case sensitive
MVS	1-8 characters	1-8 characters	Not case sensitive
VM	1-8 characters	1-8 characters	Not case sensitive

Table 4–2 shows the information the user supplies in a transfer request to a remote system for a specific file type.

Table 4–2 Access to Remote Systems Without Proxy

Remote System	What You Supply	Security System
VM system	User ID, password, and minidisk password	Validates that the access information matches and links to the minidisk using the password you supplied in the transfer request.
MVS system	User ID and password	Validates access before performing the transfer.
DECnet node	ID and password	Validates access before performing the transfer.

4.3 How to Specify an IBM User ID and Password

IBM users can specify a user ID and password on the command line for IBM-to-IBM or DECnet-to-IBM requests. The /USERID and /PASSWORD qualifiers are used to verify access for a particular user ID through the appropriate IBM security system. If you do not know the password, or if you do not have privileges for a particular user ID and password, access will be denied. If a site does not want to issue user IDs and passwords, access through a proxy is the recommended alternative.

4.4 What Is Proxy?

Proxy, in general terms, means that one person has authorization to act for another. In the context of system security, proxy means that one user has authorization to access another user's data without specifying a password. User IDs and passwords are used when proxy accounts are not defined. In addition, if default proxy user IDs are implemented, you do not need to specify the remote user IDs. Refer to Chapter 2 for more information on the use of the /USERID and /PASSWORD qualifiers. For information on recommended use, see your system administrator or the person in charge of the proxy database.

4.4.1 Benefits of Using Proxy Access

Proxy allows a user on a DECnet, TCP/IP host, or IBM system to send a file transfer request to a remote system without using a password. DECnet, and IBM users can specify files just by specifying accounts. The benefits of using proxy access are:

- No need for users to remember multiple passwords.
- Reduction in user input on the command line.
- More security, since passwords are not displayed on the terminal, in a command procedure, or on the network.
- Elimination of the administrative task of distributing passwords, and
 - Control of passwords by users.
 - File access through the network but with no logon capabilities.

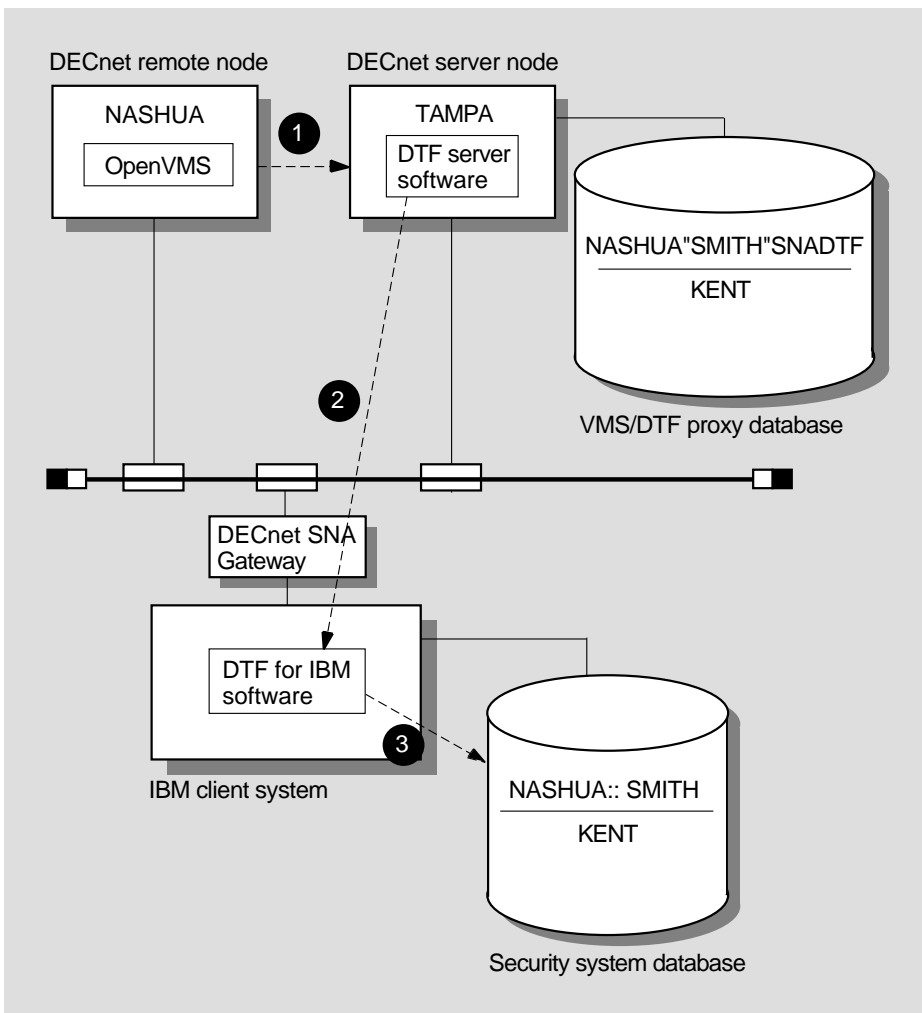
4.5 DECnet-Initiated Proxy

The IBM proxy mechanism passes the user's DECnet node or TCP/IP host name and account directly to one of the IBM security systems (such as ACF2). The IBM security system verifies that a proxy entry has been made for that user at that node and directly grants access based on rules set up by the IBM system programmer.

The IBM system programmer or security administrator implements the security system based on predetermined rules. Refer to *Digital SNA Data Transfer Facility Installation (MVS)* or *Digital SNA Data Transfer Facility Installation (VM)* for information on DECnet-initiated proxy through MVS proxy or VM proxy.

Figure 4-1 illustrates a verification to access data by MVS proxy. Note that this DECnet environment could also be TCP/IP.

Figure 4-1 Request Verified by MVS Proxy



LKG-5577-93R

1. DECnet user NASHUA::SMITH requests access to files owned by IBM user KENT.
2. The DTF for IBM software passes the DECnet user's DECnet node name (NASHUA) and account (SMITH) directly to the IBM security system. The IBM security system verifies that a proxy entry has been made for NASHUA::SMITH to access KENT's account.
3. The IBM security system grants access based on rules set up by the IBM system programmer.

4.6 IBM-Initiated Proxy

Users on IBM client systems can submit DECnet file specifications without specifying the DECnet password. This type of transfer is called "IBM-initiated access." DTF initiates the access on the Digital side with a user ID based on the IBM user ID. This allows the remote OpenVMS system manager to grant access through DECnet proxy to the DECnet accounts that the IBM users want to access. The IBM account is allowed to access only specific DECnet accounts.

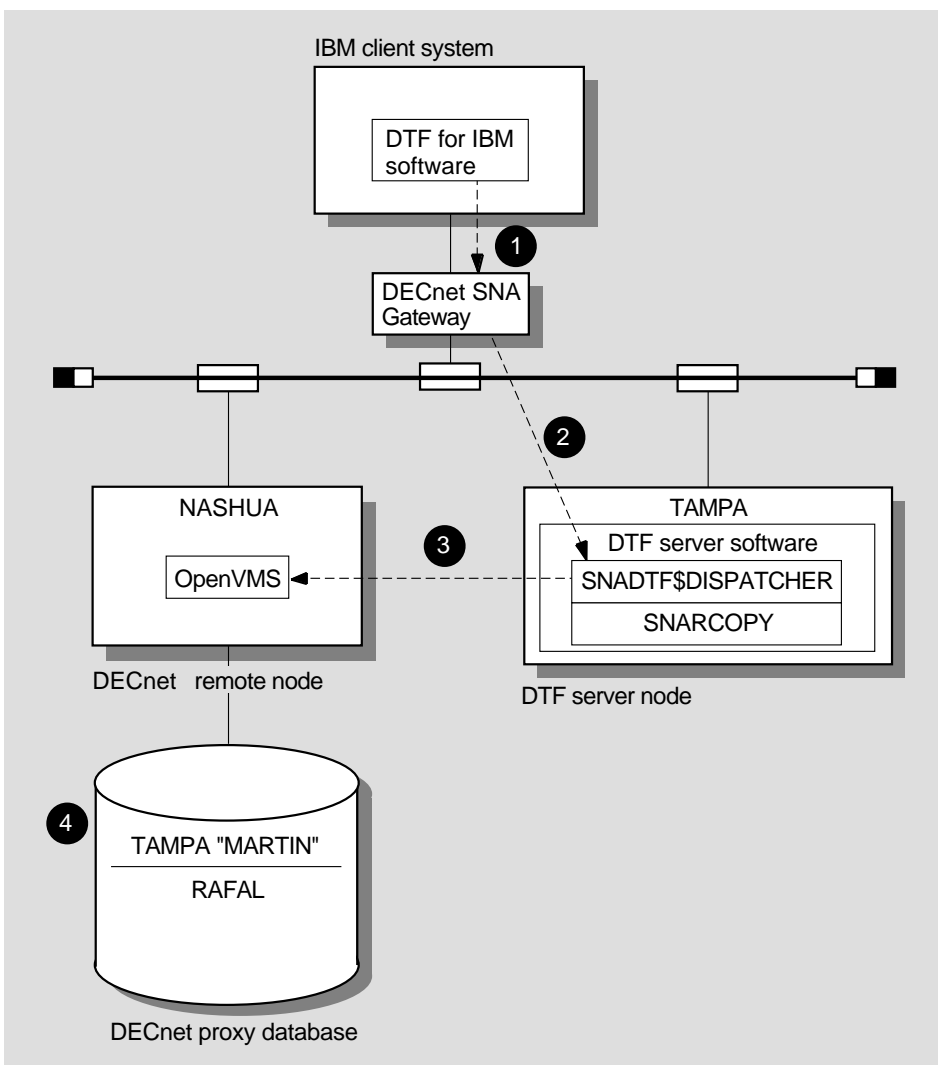
4.6.1 Example of IBM-Initiated Transfer Request

Figure 4-2 illustrates a request made from an IBM user for access to DECnet files. The remote DECnet node is not running DTF.

Note

You can use TCP/IP as a transport between the DTF Server software and the SNA Gateway instead of DECnet, when the SNA Gateway is either a Digital SNA Domain Gateway or a Digital SNA Peer Server.

Figure 4-2 IBM-Initiated Access: DECnet Client Node Not Running DTF



LKG-5494-93R

1. IBM user MARTIN issues a DTF command to access files owned by NASHUA"RAFAL."
2. The DTF server software on node TAMPA receives the request.
3. DTF changes its name to the IBM user name and a SNARCOPY request is issued.
4. NASHUA sees that TAMPA::MARTIN is trying to access a file in the RAFAL account. DECnet proxy checks if user TAMPA::MARTIN is set up for access to RAFAL. If the proxy entry is defined, the copy continues.

Note

For more detail on how proxy works, refer to the *Digital SNA Data Transfer Facility for OpenVMS Management*.

OpenVMS File Transfer Operations

This chapter describes how to use the OpenVMS DCL command interface with the DTF software to transfer files. You can use the DCL commands supported by DTF at any OpenVMS VAX, or OpenVMS Alpha node in a DECnet network so long as that node can access the OpenVMS/DTF server. You simply include the OpenVMS/DTF server node as part of the file specification. No special software is required at the client node.

You should use the DCL command interface when you need to copy small amounts of data quickly and interactively. The DCL command interface does not provide you with a checkpoint and recovery feature usually desired for large file transfers.

You should use the TRANSFER/DTF utility to transfer large amounts of data or when you desire transfer operations to occur as batch jobs. The TRANSFER/DTF utility provides you with a checkpoint and recovery feature that restarts a file transfer from the last checkpoint after a network or system failure. The TRANSFER/DTF utility does not provide checkpoint and recovery for VSAM files.

Refer to Chapter 6 for more information about the TRANSFER/DTF utility.

5.1 Transferring Files Between OpenVMS DTF Clients and IBM DTF Clients

You can use the following DTF-supported DCL commands with an IBM file specification.

- APPEND
- BACKUP
- CLOSE
- CONVERT
- COPY

- CREATE
- DELETE
- DIFFERENCES
- DIRECTORY
- EXCHANGE/NETWORK
- OPEN
- READ
- SEARCH
- SUBMIT/REMOTE
- TYPE
- WRITE

Note

DTF does not support wildcards for output file names.

5.2 APPEND Command

To add the contents of one or more specified input files to the end of a specified output file, use the DCL APPEND command:

APPEND input-file-spec[,...] output-file-spec

Table 5-1 shows the level of support DTF provides for the DCL APPEND command qualifiers. For more detailed information about the APPEND command, refer to the *DCL Dictionary*.

Notes

When DTF for MVS creates an MVS file, it calculates a maximum record length that is greater than the maximum record length of the input file (record length is the next greater multiple of 256 minus 1). Any attempt to append a file with records larger than the calculated maximum record length of the existing file will result in an error.

DTF does not support append operations to files with RECFM=FBS. The request will be rejected with the message:

-RMS-F-FOP, invalid record options

Table 5–1 DTF Level of Support for APPEND Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/ALLOCATION= <i>n</i>	supported	For IBM files, enough tracks or cylinders are allocated to hold <i>n</i> 512-byte blocks. If /ALIGNMENT:TRACKS is specified, the allocation is in tracks. If /ALIGNMENT:CYLINDERS is specified, the allocation is in cylinders.
/BACKUP	not supported ³	IBM files do not have a BACKUP date
/BEFORE[= <i>time</i>]	supported ⁴	–
/BY_OWNER[= <i>uic</i>]	supported	All IBM-resident files have a UIC of [0,0]. If you specify the /BY_OWNER qualifier with IBM input file specifications with a UIC other than [0,0], this qualifier is ignored.
/[NO]CONFIRM	supported	–
/[NO]CONTIGUOUS	ignored ¹	–
/CREATED	supported ⁴	–
/EXCLUDE= <i>filespec</i> [,...]	ignored ¹	–
/EXPIRED	supported ⁴	–

¹If specified an IBM file specification, the qualifier is ignored.

³If specified for an IBM input file specification, no file will be selected.

⁴See Section 5.10.1 for restrictions.

(continued on next page)

Table 5–1 (Cont.) DTF Level of Support for APPEND Command Qualifiers

Qualifier	For IBM File Specifications	Comments
<code>[MVS] /EXTENSION=<i>n</i></code>	supported	For IBM files, each extension allocates enough tracks or cylinders to hold <i>n</i> 512-byte blocks. If <code>/ALIGNMENT:TRACKS</code> is specified, the extension is in tracks. If <code>/ALIGNMENT:CYLINDERS</code> is specified, the extension is in cylinders.
<code>/[NO]LOG</code>	supported	–
<code>/MODIFIED</code>	supported ⁴	–
<code>/[NO]NEW_VERSION</code>	not supported	–
<code>/[NO]OVERLAY</code>	supported	–
<code>/PROTECTION=(<i>code</i>)</code>	ignored	–
<code>/[NO]READ_CHECK</code>	not supported ²	–
<code>/SINCE[=<i>time</i>]</code>	supported ⁴	–
<code>/[NO]WRITE_CHECK</code>	not supported ²	–

²Command fails with the following error message:
 -RMS-F-SUPPORT, Network operation not supported
 -FAL-F-FOP1, File processing options field rejected

⁴See Section 5.10.1 for restrictions.

Examples

`[MVS]`

```
$ APPEND TEST.LOG DRAGON"SNADTF"::"LOU.DTF(TEST) "
```

This command appends the latest version of the file TEST.LOG in the current OpenVMS default directory to the PDS member TEST of the LOU.DTF data set. The server account is SNADTF on the OpenVMS/DTF server node DRAGON. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the `/USERID` and `/PASSWORD` qualifiers.

`[MVS]`

```
$ APPEND INDU"IBMSYS1"::"ROCKY.STOCK(AMT)/USER:ROCKY/PASS:SECRET"-
_$ STOCK.QUOTE
```


This command appends the PDS member AMT of the ROCKY.STOCK data set located on the IBM DTF client to the OpenVMS file STOCK.QUOTE located on the current default directory. The server account is IBMSYS1 on OpenVMS/DTF server node INDU and the MVS user ID is ROCKY. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

VM

```
$ APPEND MASS"DTFVM"::"PROFILE EXEC A1/USER:RICHARDS" -  
_$_ DISK$1:[RICHARDS]LOGIN.COM
```

This command appends the file PROFILE EXEC to the OpenVMS file LOGIN.COM located in directory [RICHARDS]. File access on the VM system is done using mode 1. The server account is DTFVM, the OpenVMS /DTF server node is MASS, and the VM user ID is RICHARDS. Account verification on the VM system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers..

5.3 BACKUP Command

To create backup images of selected files, use the DCL BACKUP command:

`BACKUP input-file-spec[,...] output-file-spec`

Important

DTF supports only OpenVMS to IBM backup requests and only IBM to OpenVMS restore requests.

Table 5–2 shows the level of support DTF provides for the DCL BACKUP command qualifiers. For more detailed information about the BACKUP command, refer to the *OpenVMS Backup Utility Manual*.

Note

Always specify the /NOTRANSLATE IBM file specification qualifier when using the BACKUP command.

Table 5–2 DTF Level of Support for BACKUP Command Qualifiers

Qualifier	For IBM File Specifications
/[NO]ASSIST	supported
/BACKUP	supported
/BEFORE[= <i>time</i>]	supported
/BLOCK_SIZE= <i>n</i>	supported
/BRIEF	supported
/BUFFER_COUNT= <i>n</i>	supported
/BY_OWNER[= <i>uic</i>]	supported
/COMMENT= <i>string</i>	supported
/COMPARE	supported
/CONFIRM	supported
/CREATED	supported
/[NO]CRC	supported
/DELETE	supported
/DENSITY= <i>n</i>	supported
/EXCLUDE= <i>filespec</i> [,...]	supported
/EXPIRED	supported
/FAST	supported
/FULL	supported
/GROUP_SIZE= <i>n</i>	supported
/[NO]IGNORE= <i>option</i>	supported
/IMAGE	supported
/INCREMENTAL	supported
/[NO]INITIALIZE	supported
/[NO]INTERCHANGE	supported
/JOURNAL= <i>filespec</i>	supported
/LABEL= <i>string</i> [,...]	ignored
/LIST= <i>filespec</i>	supported
/[NO]LOG	supported

(continued on next page)

Table 5–2 (Cont.) DTF Level of Support for BACKUP Command Qualifiers

Qualifier	For IBM File Specifications
/MODIFIED	supported
/NEW_VERSION	supported
/[NO]OVERLAY	supported
/PHYSICAL	not supported
/PROTECTION=(<i>code</i>)	ignored
/RECORD	supported
/[NO]REPLACE	supported
/[NO]REWIND	supported
/SAVE_SET	supported
/SELECT= <i>filespec</i> [...]	supported
/SINCE[= <i>time</i>]	supported
/TAPE_EXPIRATION= <i>date</i>	ignored
/[NO]TRUNCATE	ignored
/[NO]VERIFY	not supported
/VOLUME= <i>n</i>	not supported ¹

¹Command fails with the following error message:
 -RMS-F-SUPPORT, Network operation not supported
 -FAL-F-FOP1, File processing options field rejected

Examples

MVS

```
$ BACKUP/IMAGE/IGNORE=INTERLOCK USER$100 -
_ $ DTFSVR"VAXBACKUPS" :: "DTFMAN.BACKUP/USER:DTFMAN/UNIT:TAPE/NOTRANS" /SAVE
```

The command backs up the OpenVMS disk USER\$100 to the MVS tape-resident file DTFMAN.BACKUP. The OpenVMS/DTF server node is DTFSVR and the server account is VAXBACKUPS. The MVS user ID is DTFMAN. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers. Note that the /NOTRANS qualifier is required for the BACKUP command.

VM

```
$ BACKUP/EXCLUDE=LOGIN.COM BIGGUY"VMACCESS"::"BACKUP SAVE/NOTRANS"/SAVE -  
_ $ MY$DISK:[BURR]
```

This command restores all the files in the save set BACKUP SAVE stored on the VM system with the exception of the file LOGIN.COM. The OpenVMS /DTF server node is BIGGUY and the server account is VMACCESS. Account verification on the VM system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers..

5.4 CLOSE Command

To close a file that was opened with the DCL OPEN command, use the DCL CLOSE command:

CLOSE *logical-name[:]*

Table 5–3 shows the level of support DTF provides for the DCL CLOSE command qualifiers. For more detailed information about the CLOSE command, refer to the *DCL Dictionary*.

Table 5–3 DTF Level of Support for CLOSE Command Qualifiers

Qualifier	For IBM File Specifications
/ERROR= <i>label</i>	supported
/[NO]LOG	supported

Example

```
$ CLOSE FILE1
```

This command closes a file that was opened previously with the OPEN command, and deassigns the logical name specified when the file was opened. FILE1 is the logical name that was assigned by the OPEN command when the IBM data set was opened. Note that because no file name is ever used directly in a CLOSE statement, this command is no different than any other OpenVMS CLOSE command.

5.5 CONVERT Command

To copy records from indexed files or relative files, use the DCL CONVERT command:

```
CONVERT input-file-spec[,...] output-file-spec
```

If you want to change the organization and format of the input file to those of the output file, use the CONVERT command and a suitable OpenVMS file definition file.

By using an IBM file specification as either the input file, the output file, or both, you can initiate a transfer to or from that IBM file. See Chapter 2 for the IBM file specification syntax.

Table 5–4 shows the level of support DTF provides for the DCL CONVERT command qualifiers. For more detailed information about the CONVERT command, refer to the *OpenVMS Convert and Convert/Reclaim Utility Manual*.

Table 5–4 DTF Level of Support for CONVERT Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/[NO]APPEND	supported	–
/[NO]CREATE	supported	–
/[NO]EXCEPTIONS_ FILE[= <i>file-spec</i>]	supported	–
/[NO]EXIT	supported	–
/[NO]FAST_LOAD	ignored	–
/FDL= <i>fdl-file-spec</i>	supported	–
/[NO]FILL_BUCKETS	supported	Ignored for IBM indexed files
/[NO]FIXED_CONTROL	supported	–
/KEY= <i>n</i>	supported	–
/MERGE	supported	–
/[NO]PAD[=[<i>%b</i>] <i>x</i>]	supported	–
/PROLOG= <i>n</i>	supported	–
/[NO]SHARE	supported	–

(continued on next page)

Table 5–4 (Cont.) DTF Level of Support for CONVERT Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/[NO]SPREADSHEET	ignored	–
/[NO]STATISTICS	supported	–
/WORK_FILES= <i>n</i>	supported	–
/[NO]READ_CHECK	ignored	–
/RECLAIM	not supported	–
/[NO]SORT	ignored	–
/[NO]TRUNCATE	supported	–
/[NO]WRITE_CHECK	ignored	–

You may need to use the CONVERT/MERGE command to prevent transfer failures due to differences in collating key sequences between EBCDIC and ASCII.

Examples

MVS

```
$ CONVERT/FDL=FIXED.FDL LKG"IBMSYS2"::"GARY.VARIABLE.LEN" -  
_ $ FIXED_LEN.DAT/PAD
```

This command converts the variable length data set VARIABLE.LEN on the IBM DTF client to the fixed length file FIXED_LEN.DAT located on the current default OpenVMS directory. Nulls are used to pad the record if there are any discrepancies in length. The FDL file used is FIXED.FDL. The OpenVMS/DTF server node is LKG and the server account is IBMSYS2. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

MVS

```
$ CONVERT/MERGE EMPLOYEE_INDEX.DAT LKG"IBMSYS1"::"GARY.LARGE"
```

This command will merge all the records in the EMPLOYEE_INDEX.DAT file into the indexed file GARY.LARGE. Note that GARY.LARGE must be an indexed file, as the /MERGE qualifier can only be used with indexed output files. The OpenVMS/DTF server node is LKG and the server account is IBMSYS1. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

VM

```
$ CONVERT LIST_INDEX.DAT MAIN"DTFVM"::"PETER.LIST/USER:PETER  
/PASS:SECRET/MDADDR:192/MDWPASS:ABC/VSAM"
```

(This command must be entered with no continuators within double quotation marks. Because of line-length restrictions, in this manual the command is represented as a wrapped command line.) This command copies the indexed file LIST_INDEX.DAT located in the current directory to the VSAM file PETER.LIST. Note that the /VSAM qualifier is required for all VM VSAM accesses. The OpenVMS/DTF server node is MAIN and the server account is DTFVM. Account verification on the VM system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

5.6 COPY Command

To transfer sequential files between a DECnet DTF client and an IBM DTF client use the DCL COPY command:

```
COPY input-file-spec[,...] output-file-spec
```

The COPY command allows you to do the following:

- Copy a sequential file from any IBM client to any location in a DECnet network.
- Copy a sequential file from any location in a DECnet network to any IBM client.
- Copy a sequential file from any IBM client to any other IBM client.

The COPY command does not support the checkpoint or recovery feature. If you require the checkpoint and recovery feature, use the TRANSFER/DTF COPY command described in Chapter 6.

The COPY command does not support indexed or relative file transfers. To copy indexed or relative files, use the DCL CONVERT command (see Section 5.5), the EXCHANGE/NETWORK command (see Section 5.11), or the TRANSFER/DTF utility (see Chapter 6).

By using an IBM file specification as either the input file, the output file, or both, you can initiate a transfer to or from that IBM file. See Chapter 2 for the IBM file specification syntax.

Table 5-5 shows the level of support DTF provides for the DCL COPY command qualifiers. For more detailed information about the COPY command, refer to the *DCL Dictionary*.

Note

You must use the TRANSFER/DTF utility to copy from IBM tape files.

Table 5-5 DTF Level of Support for COPY Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/ALLOCATION= <i>n</i>	supported	For IBM files, enough tracks or cylinders are allocated to hold <i>n</i> 512-byte blocks. If /ALIGNMENT:TRACKS is specified, the allocation is in tracks. If /ALIGNMENT:CYLINDERS is specified, the allocation is in cylinders.
/BACKUP	not supported ³	IBM files do not have a BACKUP date
/BEFORE[= <i>time</i>]	supported ⁴	–
/BY_OWNER[= <i>uic</i>]	supported	All IBM-resident files have a UIC of [0,0]. If you specify the /BY_OWNER qualifier with both OpenVMS and IBM input file specifications with a UIC other than [0,0], this qualifier affects only the OpenVMS input file specification.
/[NO]CONCATENATE	supported	–
/[NO]CONFIRM	supported	–
/[NO]CONTIGUOUS	ignored ¹	–
/CREATED	supported ⁴	–

¹If specified for an IBM file specification, the qualifier is ignored.

³If specified for an IBM input file specification, no file will be selected.

⁴See Section 5.10.1 for restrictions.

(continued on next page)

Table 5–5 (Cont.) DTF Level of Support for COPY Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/EXCLUDE= <i>filespec</i> [,...]	ignored ¹	–
/EXPIRED	supported ⁴	–
MVS /EXTENSION= <i>n</i>	supported	For IBM files, each extension allocates enough tracks or cylinders to hold <i>n</i> 512-byte blocks. If /ALIGNMENT:TRACKS is specified, the extension is in tracks. If /ALIGNMENT:CYLINDERS is specified, the extension is in cylinders.
/[NO]LOG	supported	–
/MODIFIED	supported ⁴	–
/[NO]OVERLAY	supported	–
/PROTECTION=(<i>code</i>)	ignored	–
/[NO]READ_CHECK	not supported ²	–
/[NO]REPLACE	supported	–
/SINCE[= <i>time</i>]	supported ⁴	–
/[NO]TRUNCATE	ignored	–
/VOLUME= <i>n</i>	ignored	If you use the /VOLUME qualifier, the following error message is displayed: %COPY-E-BADVALUE, 'VOLUME' is an invalid keyword value The CREATE command will continue to create the file.
/[NO]WRITE_CHECK	not supported ²	–

¹If specified for an IBM file specification, the qualifier is ignored.

²Command fails with the following error message:
-RMS-F-SUPPORT, Network operation not supported
-FAL-F-FOP1, File processing options field rejected

⁴See Section 5.10.1 for restrictions.

Examples

MVS

```
$ COPY RABBIT"DTF"::"PRIMA.ACCOUNTS(PAYABLE)" ACCOUNTS_P.DAT
```

This command copies the PDS member PAYABLE of the PRIMA.ACCOUNTS data set located on the IBM DTF client to the ACCOUNTS_P.DAT file located on the current OpenVMS directory. The OpenVMS/DTF server node is RABBIT and the server account is DTF. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

VM

```
$ COPY/LOG PARIS"DTFVM"::"DTF MACLIB(MEMBER1)/OWNER:JIM  
/MDRPASS:SECRET/USER:JIM/PASSWORD:GUESSME" MEMBER1.MACRO
```

(This command must be entered with no continuators within double quotation marks. Because of line-length restrictions, in this manual the command is represented as a wrapped command line.) This command copies the maclib member MEMBER1 to the local file MEMBER1.MACRO. The server account is DTFVM, the OpenVMS/DTF server node is PARIS, the VM user ID is JIM, and the VM password is GUESSME.

VM

```
$ COPY DEV.LOG -  
_ $ OXEN"DTF"::"TESTS EXEC/USER:GARY/PASS:4BY4/MDADDR:192/MDWPASS:XYZ"
```

This command copies the DEV.LOG file located on the current OpenVMS directory to TESTS EXEC on the IBM VM system. The server account is DTF, the OpenVMS/DTF server node is OXEN, and the VM user ID is GARY. The minidisk address is 192, and its password is XYZ. The file is created as a mode 1 file.

5.7 CREATE Command

To create a file on an IBM client, use the DCL CREATE command:

```
CREATE file-spec[...]
```

Table 5-6 shows the level of support DTF provides for the DCL CREATE command qualifiers. For more detailed information about the CREATE command, refer to the *DCL Dictionary*.

Note

VM You cannot create an empty CMS file. The CMS file system does not support the notion of allocating an empty file. The CREATE operation will succeed but the VM system will delete the empty file when it is closed. Note that you can use the CREATE command with

VM DTF clients as long as you enter data before ending the file data with `CTRLZ`.

Table 5–6 DTF Level of Support for CREATE Command Qualifiers

Qualifier	For IBM File Specifications	Comments
<code>/DIRECTORY</code>	not supported	–
<code>/FDL=<i>fdl-file-spec</i></code>	supported	<code>VM</code> Not supported; this qualifier creates an empty file which will be deleted by VM when the file is closed.
<code>/[NO]LOG</code>	supported	–
<code>/OWNER_UIC=<i>uic</i></code>	ignored	–
<code>/PROTECTION=(<i>code</i>)</code>	ignored	–
<code>/VOLUME=<i>n</i></code>	not supported	If you use the <code>/VOLUME</code> qualifier, the following error message is displayed: %CREATE-E-BADVALUE, 'VOLUME' is an invalid keyword value The CREATE command will continue to create the file.

Examples

`MVS`

```
$ CREATE BRIDGE "MVS DTF" :-
_ $ "GARY.DTF(CREATE)/USER:GARY/PASS:SECRET"
```

This command creates a new member `CREATE` in the `GARY.DTF` data set on the IBM DTF client. The OpenVMS/DTF server node is `BRIDGE` and the server account is `MVS DTF`. You can follow this command with data or press `CTRLZ` to exit.

`VM`

```
$ CREATE MOOSE "VMDTF" : "GAME EXEC/USER:GARY/PASS:TOOHARD"
```

This command creates a new file `GAME EXEC` on the IBM DTF client. The OpenVMS/DTF server node is `MOOSE`, the server account is `VMDTF`, the VM account is `GARY`, and the VM password is `TOOHARD`. You must follow this command with data and then press `CTRLZ` to exit.

5.8 DELETE Command

To delete a file on an IBM client, use the DCL DELETE command:

```
DELETE file-spec[...]
```

Table 5–7 shows the level of support DTF provides for the DCL DELETE command qualifiers. For more detailed information about the DELETE command, refer to the *DCL Dictionary*.

Notes

If you attempt to delete an MVS file that was created with an expiration date and that expiration date has not been reached, MVS will prompt the MVS console operator for permission to delete the file. The file operation will wait until the console operator replies.

You cannot delete IBM tape-resident files.

Table 5–7 DTF Level of Support for DELETE Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/BACKUP	not supported ³	IBM files do not have a BACKUP date
/BEFORE[= <i>time</i>]	supported ⁴	–
/BY_OWNER[= <i>uic</i>]	supported	All IBM-resident files have a UIC of [0,0]. If you specify the /BY_OWNER qualifier with both OpenVMS and IBM file specifications with a UIC other than [0,0], this qualifier affects only the OpenVMS file specification.
/[NO]CONFIRM	supported	–
/CREATED	supported ⁴	–
/[NO]ERASE	ignored	–

³If specified for an IBM input file specification, no file will be selected.

⁴See Section 5.10.1 for restrictions.

(continued on next page)

Table 5–7 (Cont.) DTF Level of Support for DELETE Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/EXCLUDE= <i>file-spec</i> [,...]	ignored ¹	–
/EXPIRED	supported ⁴	–
/[NO]LOG	supported	–
/MODIFIED	supported ⁴	–
/SINCE[= <i>time</i>]	supported ⁴	–

¹If specified an IBM file specification, the qualifier is ignored.

⁴See Section 5.10.1 for restrictions.

Examples

MVS

```
$ DELETE MYFAIR"DTFV200"::-
_ $ "NICK.ACCOUNTS(PAYABLE)/USER:NICK"
```

This command deletes the PDS member PAYABLE of the NICK.ACCOUNTS data set from the account under the user ID NICK. The OpenVMS/DTF server node is MYFAIR and the server account is DTFV200.

MVS

```
$ DELETE MYFAIR"DTFV200"::"NICK.DTF*/USER:NICK"
```

This command deletes all the data sets beginning with NICK.DTF. Caution is recommended when using wild cards. The OpenVMS/DTF server node is MYFAIR and the server account is DTFV200. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

VM

```
$ DELETE/LOG PARIS"DTFVM"::"ASHFLD * 1/USER:ROBERT"
%DELETE-I-FILDEL, PARIS:DTFVM::"ASHFLD C 1" deleted (10 blocks)
%DELETE-I-FILDEL, PARIS:DTFVM::"ASHFLD M 1" deleted (10 blocks)
%DELETE-I-FILDEL, PARIS:DTFVM::"ASHFLD Z 1" deleted (10 blocks)
```

This command deletes all files named ASHFLD with mode 1. The server account is DTFVM, the OpenVMS/DTF server node is PARIS, and the user account on the VM system is ROBERT. Account verification on the VM system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

5.9 DIFFERENCES Command

To compare the contents of two disk files and create a listing of the records that do not match, use the DCL DIFFERENCES command:

```
DIFFERENCES master-file-spec revision-file-spec
```

Notes

Unlike local OpenVMS files, if the master file specification is an IBM specification, you must supply a revision file specification. The revision file does not default to the next lower version of the master file.

The DIFFERENCES command assumes the revision file is in the same location (same node and access information). Therefore you should fully specify each file when comparing a file on a DECnet system to a file on an IBM system.

Table 5–8 shows the level of support DTF provides for the DCL DIFFERENCES command qualifiers. For more detailed information about the DIFFERENCES command, refer to the *DCL Dictionary*.

Table 5–8 DTF Level of Support for DIFFERENCES Command Qualifiers

Qualifier	For IBM File Specifications
/CHANGE_BAR [=format[,...]]	supported
/COMMENT_DELIMITER [=delimiter[,...]]	supported
/IGNORE=(option[,...])	supported
/MATCH=size	supported
/MAXIMUM_DIFFERENCES =n	supported
/MERGED[=n]	supported
/MODE=(radix[,...])	supported
/[NO]NUMBER	supported
/OUTPUT[=file-spec]	supported
/PARALLEL[=n]	supported

(continued on next page)

Table 5–8 (Cont.) DTF Level of Support for DIFFERENCES Command Qualifiers

Qualifier	For IBM File Specifications
/SEPARATED[=(<i>input-file</i> [,...])]]	supported
/SLP	supported
/WIDTH= <i>n</i>	supported
/WINDOW= <i>size</i>	supported

Examples

MVS

```
$ DIFF MYFAIR"DTFMVS_PROD":: -
_ $ "LOU.STATUS(REPORT)/USER:LOU/PASS:SECRET"-
_ $ MYVAX"LOU PASSWORD56"::STATUS.TXT
```

This command displays any differences between the PDS member REPORT of the LOU.STATUS data set located on the IBM DTF client and the STATUS.TXT file located in the default OpenVMS directory for the OpenVMS account LOU. The OpenVMS/DTF server node is MYFAIR and the server account is DTFMVS_PROD. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

MVS

```
$ DIFF MYFAIR"DTF200"::"LOU.STATUS(TEXT1)" -
_ $ MYFAIR"DTF200"::"LOU.STATUS(TEXT2)"
```

This command displays any differences between the PDS member TEXT1 of the LOU.STATUS data set located on the IBM DTF client and the PDS member TEXT2 of the LOU.STATUS data set located on the same IBM DTF client under the user ID LOU. The OpenVMS/DTF server node providing access to both IBM files is MYFAIR and the common server account is DTF200. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

VM

```
$ DIFF SYS$LOGIN:LOGIN.COM BOSTON"DTFVM"::"LOGIN.COM"
```

This command displays any differences between the file LOGIN.COM located in the SYSSLOGIN OpenVMS directory with the file LOGIN.COM on the IBM DTF client. The OpenVMS/DTF server node is BOSTON and the server account is DTFVM. Account verification on the VM system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

5.10 DIRECTORY Command

To list one or more files on an IBM client, use the DCL DIRECTORY command:

```
DIRECTORY file-spec[,...]
```

Refer to the sample directory shown at the end of this command description for information about directory contents.

Note

The DIRECTORY command is not supported for IBM tape volumes.

Table 5–9 shows the level of support DTF provides for the DCL DIRECTORY command qualifiers. For more detailed information about the DIRECTORY command, refer to the *DCL Dictionary*.

Table 5–9 DTF Level of Support for DIRECTORY Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/ACL	supported ²	–
/BACKUP	not supported ³	IBM files do not have a BACKUP date.
/BEFORE= <i>time</i>	supported ²	–
/BRIEF	supported	–

²See Section 5.10.1 for restrictions.

³If specified for an IBM input file specification, no file will be selected.

(continued on next page)

Table 5–9 (Cont.) DTF Level of Support for DIRECTORY Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/BY_OWNER[= <i>uic</i>]	supported	All IBM-resident files have a UIC of [0,0]. If you specify the /BY_OWNER qualifier with both OpenVMS and IBM input file specifications with a UIC other than [0,0], this qualifier affects only the OpenVMS input file specification.
/COLUMNS= <i>n</i>	supported	–
/CREATED	supported ²	–
/DATE= <i>option</i>	supported ²	–
/EXCLUDE= <i>file-spec</i>	ignored ¹	–
/EXPIRED	supported ²	–
/FILE_ID	supported ²	–
/FULL	supported	–
/GRAND_TOTAL	supported	–
/[NO]HEADING	supported	–
/MODIFIED	supported ²	–
/[NO]OUTPUT= <i>file-spec</i>	supported	–
/OWNER	supported ²	–
/PRINTER	supported	–
/[NO]PROTECTION	supported ²	–
/SECURITY	supported ²	–
/SELECT=(<i>keyword</i> [,...])	supported	–
/SINCE= <i>time</i>	supported ²	–
/[NO]SIZE[= <i>option</i>]	supported ²	–
/TOTAL	supported	–
/[NO]TRAILING	supported	–
/VERSION= <i>n</i>	supported	–
/WIDTH=(<i>keyword</i> [,...])	supported	–

¹If specified for an IBM file specification, the qualifier is ignored.

²See Section 5.10.1 for restrictions.

Examples

MVS

```
$ DIR/FULL 0"DTFMVS6"::"SMITH.T*/VOL:TSOVL2"
```

```
Directory 0"DTFMVS6"::
```

```
"TSOVL2:SMITH.TOM"           File ID: None
Size:          1/1           Owner:    [0,0]
Created:      4-OCT-1991 00:00:00.00
Revised:     <None specified>
Expires:     4-OCT-1992 00:00:00.00
Backup:      <No backup done>
File organization: Sequential
File attributes: Allocation: 1 , Extend: 0, Global buffer count: 0
                  Version limit: 0
Record format: Fixed length 10 byte records
Record attributes: Carriage return carriage control
RMS attributes:  None
Journaling enabled: None
File protection: System:, Owner:, Group:, World:
Access Cntrl List: None
```

```
Total of 1 file, 1/1 blocks.
```

This command displays all the fields for all files containing SMITH.T in the beginning of their names for the user SMITH. The OpenVMS/DTF server node is the local node and the server account is DTFMVS6. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers. Note that this example looks only at the specified volume and is not a catalog directory.

The MVS system maintains a date stamp only. It does not maintain a time stamp. As shown in the example, all times will be returned as 00:00:00.00.

VM

```
$ DIR/FULL/SIZE BONN"DTFVM"::"PROFILE EXEC"
```

```
Directory BONN"DTFVM"::
```

```

"PROFILE EXEC 1"           File ID: None
Size:           3/3       Owner:   [0,0]
Created:        <None specified>
Revised:       17-SEP-1991 12:45:07.00 (0)
Expires:       <None specified>
Backup:        <No backup recorded>
File organization: Sequential
File attributes: Allocation: 3, Extend: 0, Global buffer count: 0,
                  Version limit: 0
Record format:  Variable length
Record attributes: Carriage return carriage control
RMS attributes:  None
Journaling enabled: None
File protection: System:, Owner:, Group:, World:
Access Cntrl List: None

```

Total of 1 file, 3/3 blocks.

This command displays a listing of all CMS files for the user SMITH. The server account is DTFVM, the OpenVMS/DTF server node is BONN. Note that the revised date is the only date maintained by the VM/CMS file system.

Additional Information

The DIRECTORY command does not accurately display information about the following fields:

- **File ID**—Always displayed as None.
- **Size**—Always contains two numbers in the format *nn/nn*. The first number is always the same as the second number and does not indicate the amount of space currently in use as it should. These numbers are only approximations of the number of 512-byte blocks in a file.
These numbers are always 0 for partitioned data set (PDS) members.
- **Owner**—Always displayed as [0,0].
- **Creation date**—See Section 5.10.1.
- **Revision date**—See Section 5.10.1.
- **Expiration date**—See Section 5.10.1.
- **Backup date**—Always displayed as <No backup done>.
- **File attributes**
 - Extend—Always displayed as 0.
 - Global buffer count—Always displayed as 0.

- Version limit—Always displayed as 0.
- **Journaling enabled**—Always displayed as None.
- **File protection**—Always indicates that system, owner, group, and world do not have access. Access is really determined by the access control information (user name and password) specified in the IBM file specification.
- **Access control list**—Always displayed as None.

[MVS] If a file is uncataloged, the IBM file specification qualifier /VOLUME is required with the DIRECTORY command. This qualifier indicates the volumes on which the files reside on the IBM system. If /VOLUME is not specified, then an MVS catalog search will be performed.

5.10.1 Date and Time Restrictions for Created, Revised, and Expires Fields

The DTF for IBM software returns date and time information to the DECnet system for the following file types:

[VM]

- CMS files—Date and time will be returned to the revised field. The revision number will always be (0). CMS does not maintain information in the other date and time fields.
- VSAM files—Date and time will be returned to the created and expires fields. Because the IBM file system maintains a date stamp for these file structures only, the time will always be returned as 00:00:00.00.

[MVS]

- All files—Date and time will be returned to the created and expires fields. Because the IBM file system maintains a date stamp for these file structures only, the time will always be returned as 00:00:00.00.

5.11 EXCHANGE/NETWORK Command

To transfer files between a DECnet DTF client and an IBM DTF client use the DCL EXCHANGE/NETWORK command:

EXCHANGE/NETWORK *input-file-spec[,...]* *output-file-spec*

The EXCHANGE/NETWORK command allows you to do the following:

- Copy sequential files to and from IBM clients.

- Copy indexed and relative files to and from IBM clients. When indexed or relative files are copied, you must use the /TRANSFER_MODE=RECORD and /FDL DCL qualifiers.

The EXCHANGE/NETWORK command does not support the checkpoint or recovery feature. If you require the checkpoint and recovery feature, use the TRANSFER/DTF COPY command described in Chapter 6.

See Chapter 2 for the IBM file specification syntax.

Table 5–10 shows the level of support DTF provides for the DCL EXCHANGE/NETWORK command qualifiers. For more detailed information about the EXCHANGE/NETWORK command, refer to the *DCL Dictionary*.

Table 5–10 DTF Level of Support for EXCHANGE/NETWORK Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/BACKUP	not supported ³	–
/BEFORE[= <i>time</i>]	supported ⁴	–
/BY_OWNER[= <i>uic</i>]	supported	All IBM-resident files have a UIC of [0,0]. If you specify the /BY_OWNER qualifier with both OpenVMS and IBM input file specifications with a UIC other than [0,0], this qualifier affects only the OpenVMS input file specification.
/[NO]CONFIRM	supported	–
/CREATED	supported ⁴	–
/EXCLUDE= <i>filespec</i> [,...]	ignored ¹	–
/EXPIRED	supported ⁴	–
/MODIFIED	supported ⁴	–
/[NO]LOG	supported	–
/SINCE[= <i>time</i>]	supported ⁴	–
/TRANSFER_MODE= <i>mode</i>	supported	–

¹If specified for an IBM file specification, the qualifier is ignored.

³If specified for an IBM input file specification, no file will be selected.

⁴See Section 5.10.1 for restrictions.

Examples

MVS

```
$ EXCHANGE/NETWORK RABBIT"DTF"::"PRIMA.ACCOUNTS(PAYABLE)" ACCOUNTS_P.DAT
```

This command copies the PDS member PAYABLE of the PRIMA.ACCOUNTS data set located on the IBM DTF client to the ACCOUNTS_P.DAT file located on the current OpenVMS directory. The OpenVMS/DTF server node is RABBIT and the server account is DTF. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

VM

```
$ EXCHANGE/NETWORK/TRANSFER_MODE=RECORD/FDL=INDEX.FDL MYFILE.IDX -  
_ $ 0"DTFVM"::"ROLAND.DTFIDX/USER:ROLAND/MDADDR:192/MDWPASS:WCMS/VSAM"
```

This command copies the file MYFILE.IDX on the OpenVMS system to the file ROLAND.DTFIDX on the VM system. The OpenVMS/DTF server node is the local OpenVMS system and the server account is DTFVM. The destination minidisk address is 192 and the minidisk write password is WCMS. Account verification on the VM system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers. Note that the /TRANSFER_MODE and /FDL qualifiers are required for VSAM file transfer.

VM

```
$ EXCHANGE/NETWORK/LOG PARIS"DTFVM"::"DTF MACLIB(MEMBER1)/OWNER:JIM  
/MDRPASS:SECRET/USER:JIM/PASSWORD:GUESSME" MEMBER1.MACRO
```

(This command must be entered with no continuators within double quotation marks. Because of line-length restrictions, in this manual the command is represented as a wrapped command line.) This command copies the maclib member MEMBER1 to the local file MEMBER1.MACRO. The server account is DTFVM, the OpenVMS/DTF server node is PARIS, the VM user ID is JIM, and the VM password is GUESSME.

VM

```
$ EXCHANGE/NETWORK DEV.LOG -  
_ $ OXEN"DTF"::"TESTS EXEC/USER:GARY/PASS:4BY4/MDADDR:192/MDWPASS:XYZ"
```

This command copies the DEV.LOG file located on the current OpenVMS directory to TESTS EXEC on the IBM VM system. The server account is DTF, the OpenVMS/DTF server node is OXEN, and the VM user ID is GARY. The minidisk address is 192, and its password is XYZ. The file is created as a mode 1 file.

5.12 OPEN Command

To open a file residing on an IBM client for either reading or writing, use the DCL OPEN command:

```
OPEN logical-name[:] file-spec
```

Table 5–11 shows the level of support DTF provides for the DCL OPEN command qualifiers. For more detailed information about the OPEN command, refer to the *DCL Dictionary*.

Table 5–11 DTF Level of Support for OPEN Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/APPEND	supported	–
/ERROR= <i>label</i>	supported	–
/READ	supported	–
/SHARE[= <i>option</i>]	supported	Not supported for non-VSAM output files.
/WRITE	supported	–

Examples

MVS

```
$ OPEN/READ/WRITE FILE1 TENNIS"PROS"::"GIBSON.DTF(OPEN)"
```

This command opens an existing PDS member OPEN of the GIBSON.DTF data set located on the IBM client. This data set is opened for both read and write operations. Record level access to GIBSON.DTF(OPEN) is performed through READ and WRITE commands that reference FILE1. The OpenVMS/DTF server node is TENNIS and the server account is PROS. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

VM

```
$ OPEN/READ/WRITE INFILE CARS"SNADTF"::"FIFTIES EXEC/USER:PAT"
```

This command opens an existing file FIFTIES EXEC located on the IBM client under the VM user ID PAT. This data set is opened for both read and write operations. Record level access to FIFTIES EXEC is performed through READ and WRITE commands that reference INFILE. The OpenVMS/DTF server node is CARS and the server account is SNADTF. Account verification on the

VM system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

5.13 READ Command

To read a record from any file opened with the DCL OPEN command, use the DCL READ command:

READ *logical-name[:]* *symbol-name*

Table 5–12 shows the level of support DTF provides for the DCL READ command qualifiers. For more detailed information about the READ command, refer to the *DCL Dictionary*.

Table 5–12 DTF Level of Support for READ Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/DELETE	supported	–
/END_OF_FILE= <i>label</i>	supported	–
/ERROR= <i>label</i>	supported	–
/INDEX= <i>n</i>	supported	–
/KEY= <i>string</i>	supported	–
/MATCH= <i>option</i>	supported	When you use the GT(greater than) or GE(greater than or equal to) options, be aware that records are returned in EBCDIC collating order.
/[NO]LOCK	not supported ¹	–
/PROMPT= <i>string</i>	not supported ¹	–
/[NO]TIME_OUT= <i>n</i>	not supported ¹	–

¹Command fails with the following error message:
 -RMS-F-SUPPORT, Network operation not supported
 -FAL-F-FOP1, File processing options field rejected

Example

```
$ READ FILE1 RECORD
```

This command reads a single record from the data set that was opened prior to this command and assigns the contents of the record to the symbol RECORD. Note that because no file name is ever used directly in a READ statement, this command is no different than any other OpenVMS READ command.

5.14 SEARCH Command

To search one or more files for a specified string or strings, use the DCL SEARCH command:

```
SEARCH file-spec[,...] search-string[,...]
```

Every line containing a matched string is displayed.

Table 5–13 shows the level of support DTF provides for the DCL SEARCH command qualifiers. For more detailed information about the SEARCH command, refer to the *DCL Dictionary*.

Table 5–13 DTF Level of Support for SEARCH Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/BACKUP	not supported ³	–
/BEFORE= <i>time</i>	supported ⁴	–
/BY_OWNER[= <i>uic</i>]	supported	All IBM-resident files have a UIC of [0,0]. If you specify the /BY_OWNER qualifier with both OpenVMS and IBM input file specifications with a UIC other than [0,0], this qualifier affects only the OpenVMS input file specification.
/[NO]CONFIRM	supported	–
/CREATED	supported ⁴	–
/[NO]EXACT	supported	–
/EXCLUDE= <i>file-spec</i>	ignored ¹	–
/EXPIRED	supported ⁴	–
/FORMAT= <i>option</i>	supported	–
/[NO]HEADING	supported	–
/[NO]LOG	supported	–
/MATCH= <i>option</i>	supported	–
/MODIFIED	supported ⁴	–

¹If specified for an IBM file specification, the qualifier is ignored.

³If specified for an IBM input file specification, no file will be selected.

⁴See Section 5.10.1 for restrictions.

(continued on next page)

Table 5–13 (Cont.) DTF Level of Support for SEARCH Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/[NO]NUMBERS	supported	–
/[NO]OUTPUT[= <i>file-spec</i>]	supported	–
/[NO]REMAINING	supported	–
/SINCE= <i>time</i>	supported ⁴	–
/[NO]STATISTICS	supported	–
/[NO]WINDOW[=(<i>n1,n2</i>)]	supported	–

⁴See Section 5.10.1 for restrictions.

Examples

MVS

```
$ SEARCH LKG"DTF200"::"GARY.DTF(TEXT)" DTF
```

This command searches the PDS member TEXT of the GARY.DTF data set located on the IBM DTF client for the string “DTF”. It lists all the lines containing this string. The OpenVMS/DTF server node is LKG and the server account is DTF200. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

VM

```
$ SEARCH PARIS"VMDTF"::"DTF DTF/MDADDR:181" DEFINE
```

This command searches the file DTF DTF located on the IBM DTF client for the string “DEFINE”. It lists all the lines containing this string. The IBM system uses the first DTF DTF file it finds, regardless of mode value. The OpenVMS/DTF server node is PARIS and the server account is VMDTF. Account verification on the VM system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers. The minidisk address is 181 and the minidisk passwords are taken from the proxy database default entry.

5.15 SUBMIT/REMOTE Command

To submit selected files for remote execution, use the DCL SUBMIT/REMOTE command:

```
SUBMIT/REMOTE file-spec[...]
```

Table 5–14 shows the level of support DTF provides for the DCL SUBMIT/REMOTE command qualifiers. For more detailed information about the SUBMIT/REMOTE command, refer to the *DCL Dictionary*.

Note

Files submitted to IBM batch subsystems must not contain any records that exceed 80 bytes.

Table 5–14 DTF Level of Support for SUBMIT/REMOTE Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/BACKUP	not supported ³	IBM files do not have a BACKUP date.
/BEFORE[= <i>time</i>]	supported ⁴	–
/BY_OWNER[= <i>uic</i>]	supported	All IBM-resident files have a UIC of [0,0]. If you specify the /BY_OWNER qualifier with both OpenVMS and IBM input file specifications with a UIC other than [0,0], this qualifier affects only the OpenVMS input file specification.
/CONFIRM	supported	–
/CREATED	supported ⁴	–
/EXCLUDE= <i>filespec</i> [...]	ignored ¹	–
/EXPIRED	supported ⁴	–
/MODIFIED	supported ⁴	–

¹If specified for an IBM file specification, the qualifier is ignored.

³If specified for an IBM input file specification, no file will be selected.

⁴See Section 5.10.1 for restrictions.

(continued on next page)

Table 5–14 (Cont.) DTF Level of Support for SUBMIT/REMOTE Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/SINCE[= <i>time</i>]	supported ⁴	–

⁴See Section 5.10.1 for restrictions.

Examples

MVS

```
$ SUBMIT/REMOTE 0"SNADTF"::"SMITH.CNTL(JCL)/USER:SMITH"
```

This command submits the PDS member JCL of the SMITH.CNTL data set to the MVS batch system. The OpenVMS/DTF server node is the local node and the server account is SNADTF. The MVS user ID is SMITH. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers. The file JCL must not contain any records that exceed 80 bytes.

VM

```
$ SUBMIT/REMOTE 0"DTFVM"::"TESTBAT EXEC/USER:GIBSON/BATCH:CMSBATCH/CLASS:A"
```

This command submits the file TESTBAT EXEC to the batch machine CMSBATCH in the class A. The OpenVMS/DTF server node is the local node and the server account is DTFVM. The VM user ID is GIBSON. Account verification on the VM system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers. The file TESTBAT EXEC must not contain any records that exceed 80 bytes.

5.16 TYPE Command

To display on a terminal a file residing on an IBM client, use the DCL TYPE command:

```
TYPE file-spec[...]
```

Table 5-15 shows the level of support DTF provides for the DCL TYPE command qualifiers. For more detailed information about the TYPE command, refer to the *DCL Dictionary*.

Table 5–15 DTF Level of Support for TYPE Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/BACKUP	not supported ³	–
/BEFORE[= <i>time</i>]	supported ⁴	–
/BY_OWNER[= <i>uic</i>]	supported	All IBM-resident files have a UIC of [0,0]. If you specify the /BY_OWNER qualifier with both OpenVMS and IBM input file specifications with a UIC other than [0,0], this qualifier affects only the OpenVMS input file specification.
/[NO]CONFIRM	supported	–
/CREATED	supported ⁴	–
/EXCLUDE= <i>file-spec</i> [,...]	ignored ¹	–
/EXPIRED	supported ⁴	–
/MODIFIED	supported ⁴	–
/[NO]OUTPUT[= <i>file-spec</i>]	supported	–
/[NO]PAGE	supported	–
/SINCE[= <i>time</i>]	supported ⁴	–

¹If specified an IBM file specification, the qualifier is ignored.

³If specified for an IBM input file specification, no file will be selected.

⁴See Section 5.10.1 for restrictions.

Example

MVS

```
$ TYPE TENNIS"PROS"::"GARY.DTF(TYPE) "
```

This command types the contents of the PDS member TYPE of the GARY.DTF data set located on the IBM DTF client. The OpenVMS/DTF server node is TENNIS and the server account is PROS. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

VM

```
$ TYPE CITIES"DTFVM"::"PROFILE EXEC A1/USER:SAM"
```

This command types the contents of the file PROFILE EXEC on the IBM DTF client. The OpenVMS/DTF server node is CITIES, the server account is DTFVM, and the IBM user is SAM. Account verification on the VM system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers. the /USERID and /PASSWORD qualifiers. The /MDADDRESS value and /MDRPASSWORD, /MDWPASSWORD, and /MDMPASSWORD values are taken from the OpenVMS/DTF server's proxy database.

5.17 WRITE Command

To write a record to a file that was opened with the DCL OPEN command, use the DCL WRITE command:

```
WRITE logical-name expression[,...]
```

Table 5–16 shows the level of support DTF provides for the DCL WRITE command qualifiers. For more detailed information about the WRITE command, refer to the *OpenVMS DCL Dictionary*.

Table 5–16 DTF Level of Support for WRITE Command Qualifiers

Qualifier	For IBM File Specifications
/ERROR= <i>label</i>	supported
/SYMBOL	supported
/UPDATE	supported

Example

```
$ WRITE FILE1 RECORD
```

This command writes the specified data to the output data set indicated by the logical name FILE1. The data is written as one record in the output data set. Note that because no file name is ever used directly in a WRITE statement, this command is no different than any other OpenVMS WRITE command.

6

OpenVMS TRANSFER/DTF File Transfer Operations

This chapter describes how to use the TRANSFER/DTF utility's command interface to transfer files. You should use the TRANSFER/DTF command interface to transfer large amounts of data or when you desire transfer operation to occur as batch jobs. The TRANSFER/DTF interface provides you with a checkpoint and recovery feature that resumes a file transfer from the last checkpoint after a network or system failure.

Note

Although the TRANSFER/DTF utility can be used to transfer both VSAM and non-VSAM files, it does not support recoverable copy for VSAM files.

You should use the OpenVMS DCL command interface when you need to copy small amounts of data quickly and interactively. The DCL command interface does not provide you with the checkpoint and recovery feature.

Table 6-1 summarizes the conditions when recoverable file operations are possible.

Table 6-1 Recoverable File Operations

Copy Initiated From	Input File	Output File	Recoverable Copy
OpenVMS client with TRANSFER/DTF	On same OpenVMS node	On IBM DTF client	YES

(continued on next page)

Table 6–1 (Cont.) Recoverable File Operations

Copy Initiated From	Input File	Output File	Recoverable Copy
	On another DTF client node	On IBM DTF client	NO
	On IBM DTF client	On same OpenVMS node	YES
	On IBM DTF client	On another DTF client node	NO
OpenVMS DTF client without TRANSFER/DTF	On any DTF client	On any DTF client	NO

This chapter describes how to use the TRANSFER/DTF utility command interface to perform the following tasks:

- Invoking the TRANSFER/DTF utility.
- Exiting from the TRANSFER/DTF utility.
- Copying files with checkpoint and recovery.
- Defining OpenVMS/DTF queue attributes.
- Displaying the status of your file transfer jobs.
- Displaying OpenVMS/DTF queue characteristics.
- Displaying the OpenVMS/DTF software version number.
- Canceling your file transfer jobs.
- Displaying OpenVMS/DTF error messages.

For information on starting the OpenVMS/DTF software and on creating DTF queues, see the *Digital SNA Data Transfer Facility for OpenVMS Installation* manual.

6.1 Invoking the TRANSFER/DTF Utility

Invoke the TRANSFER/DTF utility by entering the following command:

```
$ TRANSFER/DTF [command]
```

If you specify a command, TRANSFER/DTF will exit back to DCL after the requested operation completes or is queued. Otherwise, TRANSFER/DTF will prompt you for a command with the TRANSFER/DTF prompt:

```
TRANS/DTF>
```

6.2 Exiting the TRANSFER/DTF Utility

Exit the TRANSFER/DTF utility by entering the EXIT command or by typing `CTRL/Z` in response to the TRANSFER/DTF prompt.

6.3 Copying Files with the TRANSFER/DTF Utility

To transfer files between a DECnet DTF client and an IBM DTF client, use the TRANSFER/DTF COPY command:

```
COPY input-file-spec[,...] output-file-spec
```

By default, this command includes a checkpoint and recovery feature (/RECOVER) for non-VSAM files. If communications fail or a system goes down during a copy operation, the checkpoint and recovery feature resumes copying at the last checkpoint when communications are reestablished or when the system comes back up.

To transfer VSAM files you must specify the /NORECOVER option. TRANSFER/DTF does not support recoverable file transfers for VSAM files.

After a user issues a TRANSFER/DTF COPY command, the \$ENTRY symbol is set to the value of the entry number that was assigned to the queued job.

Note

You cannot use wildcard characters either in the input file or output file specification.

COPY Command Qualifiers

Table 6-2 describes the level of support for TRANSFER/DTF COPY qualifiers that are also available through the DCL COPY command. Table 6-3 lists the TRANSFER/DTF COPY qualifiers that are unique to TRANSFER/DTF. These qualifiers are described following Table 6-3.

Table 6–2 TRANSFER/DTF Support for DCL COPY Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/ALLOCATION	supported	For an IBM output file, enough tracks or cylinders are allocated on the IBM system to hold a number <i>n</i> of 512-byte blocks. Enough 512-byte blocks in tracks are allocated if ALIGNMENT:TRACKS is specified as part of the IBM file specification. Enough 512-byte blocks in cylinders are allocated if /ALIGNMENT:CYLINDERS is specified as part of the IBM file specification.
/BACKUP	not supported ³	IBM files do not have a BACKUP date
/BEFORE= <i>time</i>	supported ⁴	–
/BY_OWNER[= <i>uic</i>]	supported	All IBM-resident files have a UIC of [0,0]. If you specify the /BY_OWNER qualifier with both OpenVMS and IBM input file specifications with a UIC other than [0,0], this qualifier affects only the OpenVMS input file specification.
/[NO]CONTIGUOUS	ignored ¹	–
/CREATED	supported ⁴	–
/EXPIRED	supported ⁴	–
/EXTENSION	supported	For IBM files, each extension allocates enough tracks or cylinders to hold <i>n</i> 512-byte blocks. If /ALIGNMENT:TRACKS is specified, the extension is in tracks. If /ALIGNMENT:CYLINDERS is specified, the extension is in cylinders.
/MODIFIED	supported ⁴	–
/[NO]OVERLAY	supported	–

¹If specified for an IBM file specification, the qualifier is ignored.

³If specified for an IBM input file specification, no file will be selected.

⁴See Section 5.10.1 for restrictions.

(continued on next page)

Table 6–2 (Cont.) TRANSFER/DTF Support for DCL COPY Command Qualifiers

Qualifier	For IBM File Specifications	Comments
/PROTECTION	supported	For an IBM output file, protection is set up with a file definition.
/[NO]READ_CHECK	not supported ²	–
/[NO]REPLACE	ignored	–
/SINCE[= <i>time</i>]	supported ⁴	–
/[NO]TRUNCATE	not supported ²	–
/VOLUME= <i>n</i>	not supported ²	–
/[NO]WRITE_CHECK	not supported ²	–

²Command fails with the following error message:
 -RMS-F-SUPPORT, Network operation not supported
 -FAL-F-FOP1, File processing options field rejected

⁴See Section 5.10.1 for restrictions.

Table 6–3 Unique TRANSFER/DTF COPY Command Qualifiers

Qualifier	Default
/AFTER= <i>time</i>	None
/CHARACTERISTICS= (<i>characteristic</i> [,...])	None
/CHECKPOINT_INTERVAL [= <i>interval</i>]	1000 records
/[NO]HOLD	/NOHOLD
/[NO]IDENTIFY	/IDENTIFY
/INTERVAL= <i>delta-time</i>	Default interval time for the queue
/[NO]LOG_FILE[= <i>file-name</i>]	SYSSLOGIN: <i>job-name</i> .LOG
/NAME= <i>job-name</i>	First file name
/[NO]NOTIFY	/NONOTIFY
/[NO]POST_PROCESSING= [<i>file-spec</i>]	/NOPOST_PROCESSING
/PRIORITY= <i>n</i>	DEFQUEPRI SYSGEN value
/[NO]QUEUE[= <i>queue-name</i> [:]]	SNADTF\$QUEUE

(continued on next page)

Table 6–3 (Cont.) Unique TRANSFER/DTF COPY Command Qualifiers

Qualifier	Default
/[NO]RECOVER	/RECOVER
/RETRIES= <i>n</i>	Default retry limit for the queue
/SINCE[= <i>time</i>]	/SINCE=TODAY
/TIME= <i>time</i>	Default time value for the queue

Command Qualifiers Unique to TRANSFER/DTF

/AFTER=time

Specifies holding the copy operation until after a time that you have specified. If the time specified with the */AFTER* qualifier has passed, the job is queued for immediate processing.

You can specify either an absolute time or a combination of absolute and delta times. Refer to the *DCL Concepts Manual* for more complete information about specifying time values.

/CHARACTERISTICS=(characteristic[,...])

Specifies one or more queue characteristics for transferring files. If you specify only one characteristic, you can omit the parentheses. Codes for characteristics consist either of names or of values from 0 to 127 and are installation defined. Use the DCL SHOW QUEUE/CHARACTERISTICS command to display the characteristics defined for your system. Use the TRANSFER/DTF SHOW QUEUE/FULL command to display the characteristics defined for a particular queue.

A copy operation can execute only if each characteristic specified with the COPY command is also specified on the DTF queue. If you specify a characteristic not defined for the DTF queue, the job remains in a pending state. In order to start your job, the system manager must specify new values with the DCL SET QUEUE command using the */CHARACTERISTICS* qualifier.

A COPY command that does not list all the characteristics defined for a DTF queue executes without pending.

/CHECKPOINT_INTERVAL[=interval]

Specifies a number in the range 1 to 65,376 that controls the number of records transferred between checkpoints during a recoverable file transfer; the default interval is 1,000 records. This qualifier is not supported for VSAM file transfers.

/[NO]HOLD

Specifies whether or not a job is available for immediate processing. If you specify the /HOLD qualifier, the job is not released for processing until you release it with a DCL SET QUEUE/ENTRY command with one of the following qualifiers:

- /NOHOLD
- /RELEASE

/[NO]IDENTIFY

Specifies whether the COPY command displays a message containing the job number and job name of the file transfer and the name of the OpenVMS/DTF queue in which the file transfer is entered.

The default qualifier /IDENTIFY displays the job number, the job name, and the queue name to which the job is queued.

/INTERVAL=*delta-time*

Specifies a delta time interval applied to a recoverable copy operation. The delta time is the interval that elapses before DTF attempts to resume a stalled copy operation (because of a network link going down). If you specify a short time interval, DTF attempts to resume the transfer more frequently. This qualifier is not supported for VSAM file transfers.

The /INTERVAL value you specify can not be less than the minimum interval specified on the queue by the system manager. You can use the SHOW QUEUE /FULL command to determine what this value is.

If you specify an /INTERVAL value less than the system minimum, you get an error message and the system minimum interval value is used in its place. If you do not specify an /INTERVAL value, the system default interval is used. To determine what this value is use the TRANSFER/DTF SHOW QUEUE/FULL command.

/[NO]LOG_FILE[=*file-name*]

Specifies whether a log file is created. If you use the /LOG_FILE qualifier, DTF writes the log file to a file you specify. The log file you specify must be located on the local node, and wildcard characters are not allowed in the file specification. The default is /LOG_FILE=SYSS\$LOGIN:*job-name*.LOG. If you use the /NOLOG_FILE qualifier, a log file is not created.

You can also use the /LOG_FILE qualifier to send the log file to a specific device. The logical names that you include in the file specification are translated in the context of the process that submits the job.

For more information about specifying a job name, refer to the /NAME qualifier.

/NAME=job-name

Specifies a job name consisting of 1 to 39 characters to identify a job. The SHOW JOBS command displays all the job names you specified.

If you do not specify a job name with the /NAME qualifier, the name defaults to the first or only input file in the job. In this case, the job name is based on the input file specification. The following list explains how different input file specifications are used to create a job name.

1. **OpenVMS file specification.** Uses the name portion of the input file specification as the job name. For example,
DRA1:[SMART]RAW_DATA.DAT;6
produces a job name of RAW_DATA.
2. **IBM partitioned data set or maclib member.** Uses the PDS member name or maclib member name as the job name. For example,
MAIN.JCL.COM(BACKUP) and DTF MACLIB(DTFMCS)
produce job names of BACKUP and DTFMCS, respectively.
3. **IBM generation data set.** Uses the last portion of the file specification before the parentheses as the job name. For example,
TOASTER.OVEN(-1)
produces a job name of OVEN.
4. **All other MVS or VSAM IBM data sets.** Uses the last portion of the name as the job name. For example,
TOP.LEVEL1.LEVEL2.LEVEL3.BOTTOM
produces a job name of BOTTOM.
5. **All other VM file names.** Uses the whole file name to create the job name. The job name is formed by concatenating the parts of the file name together using the underline character. For example,
PROFILE EXEC 1
produces a job name of PROFILE_EXEC_1.

/[NO]NOTIFY

Specifies whether a message is broadcast to any terminal at which you are logged in, notifying you when your print job completes or aborts. If you delete a job while it is not currently executing, you do not receive a deletion message. By default, a message is not broadcast.

/[NO]POST_PROCESSING[=*file-spec*]

Specifies a file that DTF should submit if an operation completes successfully. The file can be an OpenVMS command procedure, an MVS JCL file, or a VM proc file. The default is /NOPOST_PROCESSING.

If you specify a DECnet file specification or a local OpenVMS file specification, the file is submitted on the indicated DECnet node, or on the local OpenVMS system.

If you specify an IBM file specification using the qualifiers described in Chapter 2, the file is submitted to the batch subsystem on the indicated IBM system. See the examples that follow this section.

/PRIORITY=*n*

Requires OPER (operator) or ALTPRI (alter privilege) to raise the priority value of the SYSGEN parameter MAXQUEPRI.

Specifies the scheduling priority for the specified job. The priority value can be in the range of 0 to 255, where 0 is the lowest priority and 255 is the highest.

The default value for /PRIORITY is the value of the SYSGEN parameter DEFQUEPRI. No privilege is needed to set the priority lower than the MAXQUEPRI value.

The /PRIORITY qualifier has no effect on the process priority. The queue establishes the process priority.

/[NO]QUEUE[=*queue-name*[:]]

Specifies the name of the queue in which the job is entered. If the transfer fails, the job is rescheduled for a later transfer. By default, the file transfer job is entered in the SNADTF\$QUEUE queue. You can override this queue by specifying another queue with the /QUEUE qualifier.

The /NOQUEUE qualifier allows you to specify that the copy operation be performed from the current process. The /NOQUEUE qualifier cannot be used with the /RECOVER qualifier.

/[NO]RECOVER

Indicates whether an attempt to retry a failed copy operation should be made. The /RECOVER qualifier cannot be used with the /NOQUEUE qualifier. The default is /RECOVER. You must specify /NORECOVER for VSAM files.

/RETRIES=*n*

Specifies the number of times that DTF attempts to resume a file transfer operation after an initial failure. This qualifier is not supported for VSAM file transfers.

The value for *n* cannot exceed the queue maximum retries value set by the system manager. You can use the TRANSFER/DTF SHOW QUEUE/FULL command to determine what this value is. If the value you set with the /RETRIES qualifier is greater than the queue maximum, you get an error message and the system maximum retries value is used in its place.

If you do not specify a value with the /RETRIES qualifier, then the queue default retries value is used. You can use the SHOW QUEUE/FULL command to determine what this value is.

You cannot use the /RETRIES qualifier with the /NOQUEUE qualifier.

/SINCE[=*time*]

Specifies files dated after a specified time. You can specify an absolute time or a combination of absolute and delta times. Refer to the *OpenVMS General User's Manual* or the on-line help topic SPECIFY DATE_TIME for more complete information about setting time values. See also the TODAY, TOMORROW, and YESTERDAY keywords. If you do not specify a time, /SINCE=TODAY is the default qualifier.

/TIME=*time*

Specifies the duration of a file transfer operation before it is aborted. If the assigned time is exceeded during a copy operation, the job continues and is not aborted. If the copy operation fails before completing and the assigned time passes or has passed, the copy operation is aborted. This qualifier is not supported for VSAM file transfers.

The time value you select cannot exceed the queue maximum time value set by the system manager. You can use the TRANSFER/DTF SHOW QUEUE/FULL command to determine what this value is. If the value associated with the /TIME qualifier is greater than the system maximum, the value you specified causes an error and the system maximum value is used in its place.

If you do not specify the /TIME qualifier, the queue default time value is substituted. You can use the TRANSFER/DTF SHOW QUEUE/FULL command to determine what this value is.

You cannot use the /TIME qualifier with the /NOQUEUE qualifier.

You may need to use the IBM file specification qualifier /NOTRANSLATE to prevent transfer failures due to differences in collating key sequences between EBCDIC and ASCII.

Examples

MVS

```
$ TRANSFER/DTF
TRANS/DTF> COPY DAILY_LEDGER.DAT NODE5"SNADTF":: -
_TRANS/DTF> "ACCT.LEDGER/USER:ACCT/PASS:SECRET/RECORD:LEDGER_RECORD"
Job DAILY_LEDGER (queue SNADTF$QUEUE, entry 1075) started on SNADTF$QUEUE
```

This COPY command copies the latest version of the file DAILY_LEDGER.DAT in the current OpenVMS default directory to the IBM system using the SNADTF server account on OpenVMS/DTF server node NODE5. It creates an IBM data set called ACCT.LEDGER.

The IBM file specification contains three qualifiers. The /USERID and /PASSWORD qualifiers specify a user account (ACCT) and password (SECRET) used by the appropriate security product to allow access to the IBM system. The /RECORD_DEFINITION qualifier directs DTF to perform a field-by-field translation on each record of the file using the LEDGER_RECORD entry in the VAX Common Data Dictionary.

Notice that a hyphen (-) continues a command line that is too long for the screen. After you execute this command, the system (by default) enters the copy request on the OpenVMS/DTF queue named SNADTF\$QUEUE.

MVS

```
$ TRANSFER/DTF
TRANS/DTF> COPY/QUEUE=NIGHTLY/NAME=NIGHTLY_UPDATE/NOTIFY -
_From:NODE5"INVENTORY"::"INV.STOCK.ABC/FILE:INV_UPDATE/PASS:SECRET" -
_To: $1$DUA3:[INVENTORY.DATA]STOCKROOM_UPDATE.DAT
Job NIGHTLY_UPDATE (queue NIGHTLY, entry 106) started on NIGHTLY
```

This COPY command copies the data set in INV.STOCK.ABC from the IBM DTF client using the INVENTORY server account on OpenVMS/DTF server node NODE5. The data set is copied to a file called STOCKROOM_UPDATE.DAT in the [INVENTORY.DATA] directory on the OpenVMS system (user disk \$1\$DUA3).

The command qualifiers specify that the copy request be entered on the queue called NIGHTLY, that the job name be NIGHTLY_UPDATE, and that the system notify you with a broadcast message when it completes the job.

The IBM file specification qualifier /FILE_DEFINITION supplies all of the values other than the password (including the user ID) required for the transfer. These values have been previously defined in a record named INV_UPDATE. The system (by default) performs EBCDIC-to-ASCII text translation on the file.

Notice that the system prompts you for an input file (From:) and an output file (To:) if you execute the COPY command without including the file names.

MVS

```
$ TRANSFER/DTF
TRANS/DTF> COPY -
  _TRANS/DTF> 0"SNADTF"::"DEPT.DATA/USER:JONES/PASS:SECRET/NOTRANS"-
  _TRANS/DTF> RAW_DATA.DAT/CONTIGUOUS
Job RAWDATA (queue SNADTF$QUEUE, entry 62) started on SNADTF$QUEUE

TRANS/DTF> EXIT

$ SHOW SYMBOL $ENTRY
$ENTRY = "62"
$ SHOW ENTRY '$ENTRY
Jobname      Username      Entry   Blocks   Status
-----
RAW_DATA     JONES         62      500      Executing
On Server queue SNADTF$QUEUE

$ DELETE/ENTRY='$ENTRY
```

This COPY command copies the IBM file DEPT.DATA using the SNADTF server account on the local OpenVMS/DTF server node. The file is copied to the file RAW_DATA.DAT in the current OpenVMS directory. After exiting the TRANSFER/DTF utility, the user issued the SHOW SYMBOL command to get the job entry number stored in \$ENTRY. The SHOW ENTRY command displayed the status of the entry. The user then issued the DELETE ENTRY command to delete the job.

The IBM file specification has three qualifiers. The /USERID and /PASSWORD qualifiers specify the user account and password needed to gain access to the IBM system. The /NOTRANSLATE qualifier specifies that no translation is to be performed on the file during transfer.

The /CONTIGUOUS qualifier specifies that the OpenVMS output file should occupy consecutive physical disk blocks.

VM

```
$ TRANSFER/DTF
TRANS/DTF> COPY/INTERVAL=00:10/TIME=02:00 MEMO.TXT -
  _TRANS/DTF> DAPPER"DTFVM"::"RAISE SCRIPT/USER:SMITH/PASS:JOHN"
Job MEMO (queue SNADTF$QUEUE, entry 17) started on SNADTF$QUEUE
```

This COPY command copies the OpenVMS file MEMO.TXT, using the DTFVM server account on OpenVMS/DTF server node DAPPER, to the IBM file RAISE of type SCRIPT.

The command qualifiers specify what you want to happen if the copy operation fails. The /INTERVAL qualifier causes DTF to attempt to resume the copy operation every 10 minutes. The /TIME qualifier aborts the operation if the file transfer does not complete within 2 hours.

The IBM file specification qualifiers specify the user account named SMITH and the password JOHN.

VM

```
$ TRANSFER/DTF
TRANS/DTF> COPY/POST=0"DTFVM"::"POST EXEC" -
_TRANS/DTF> SYS$LOGIN:LEDGER.DAT 0"DTFVM"::"LEDGER DAT"
```

This COPY command requests that once the file SYS\$LOGIN:LEDGER.DAT is copied to 0"DTFVM"::"LEDGER DAT", and the file transfer completes successfully, then DTF will submit the 0"DTFVM"::"POST EXEC" file on the remote batch system. The post-processing file is an EXEC file on a DTF for IBM VM system.

MVS

```
$ TRANSFER/DTF
TRANS/DTF> COPY SYS$LOGIN:LEDGER.DAT -
_TRANS/DTF> 0"SNADTF"::"USERID.LEDGER.ACCOUNT" -
_TRANS/DTF> /POST=0"SNADTF"::"USERID.JCL(POST)"
```

This COPY command requests that once the file SYS\$LOGIN:LEDGER.DAT is copied to 0"SNADTF"::"USERID.LEDGER.ACCOUNT", and the file transfer completes successfully, then DTF will submit the 0"SNADTF"::"USERID.JCL(POST)" file to a JES subsystem. The post-processing file is an MVS JCL file on a DTF for IBM MVS system.

OpenVMS

```
$ TRANSFER/DTF
TRANS/DTF> COPY /POST=USER$:[DTFUSER]POST.COM -
_TRANS/DTF> USER$:[DTFUSER]LEDGER.DAT 0"DTFVM"::"ACCOUNT DAT"
```

This COPY command requests that once the file LEDGER.DAT is copied to 0"DTFVM"::"ACCOUNT.DAT", and the transfer completes successfully, then DTF will submit the file POST.COM to SYS\$BATCH. The post-processing file is an OpenVMS command procedure file on an OpenVMS system.

6.4 Defining OpenVMS/DTF Queue Attributes

To define and modify OpenVMS/DTF queue attributes, use the TRANSFER /DTF SET QUEUE command:

```
SET QUEUE [queue-name][/qualifiers]
```

Operator (OPER) privileges are required to use the SET QUEUE command.

Command Parameter

queue-name

Specifies an OpenVMS/DTF queue name. The queue name must be an OpenVMS/DTF queue name. The default for this parameter is SNADTF\$QUEUE.

You cannot use wildcard characters in the *queue-name* parameter. You cannot use a generic or logical queue name as a parameter.

SET QUEUE Command Qualifiers

SET QUEUE Qualifier	Default
/DEFAULT=(<i>option</i> [,...])	none
/MAXIMUM=(<i>option</i> [,...])	none
/MINIMUM=(<i>option</i> [,...])	none

/DEFAULT=(*option*[,...])

You can use the /DEFAULT qualifier with the following options:

- INTERVAL=*hh:mm:ss*
- RETRIES=*n*
- TIME=*day-hh:mm:ss.cc*

The definitions for the options that you can use with the /DEFAULT qualifier are as follows.

INTERVAL=
hh:mm:ss defines the queue's default interval attribute. The value *hh:mm:ss* specifies the default delta time that elapses before a stalled DTF operation (because of a network failure) is resumed. If you specify a short time interval, the queue restarts and tests more frequently to determine whether the DTF operation has restarted.

You cannot set a default interval value that is less than the value for the queue's minimum interval attribute. By default, the default interval value is 30 minutes.

RETRIES=*n* defines the queue's default retries attribute. The value *n* specifies the default number of times a file transfer is attempted before the operation is aborted.

You cannot set a default retries value that is greater than the value for the queue's maximum retries attribute. By default, the default retry value is 48.

(48 retries * 30 minutes = 1 day)

TIME=*day-hh:mm:ss.cc* defines the queue's default time attribute. The value *day-hh:mm:ss.cc* specifies the time a transfer operation can exist before it is aborted. If the assigned time is exceeded during a copy operation, however, the job is not aborted and finishes copying.

You cannot set a default time value that is more than the value for the queue's maximum time attribute. By default, the default time value is:

1-00:00:00.00 (1 day)

/MAXIMUM=(*option*[,...])

You can use the /MAXIMUM qualifier with the following options:

- RETRIES=*n*
- TIME=*time*

The definitions for the options that you can use with the /MAXIMUM qualifier are:

RETRIES=*n* defines the queue's maximum retries attribute. The value *n* specifies the maximum retries a user can request with the /RETRIES qualifier.

If the number specified with the /RETRIES qualifier on the COPY command is greater than the queue's maximum retries value, the rate specified with the COPY command is ignored and the queue's maximum retries value is used in its place.

To clear the maximum retries attribute, enter the word RETRIES without typing a number after the equals sign, as in the following example:

```
$ SET QUEUE/MAXIMUM=RETRIES
```

TIME=*time* defines the queue's maximum time attribute. The value *time* specifies the maximum time a user can request with the /TIME qualifier.

If the value specified with the /TIME qualifier on the COPY command is greater than the queue's maximum time value, the rate specified with the COPY command is ignored and the queue's maximum value is used in its place.

To clear the maximum time attribute, enter the word TIME without typing a number after the equals sign, as in the following example:

```
$ SET QUEUE/MAXIMUM=TIME
```

/MINIMUM=(*option*[,...])

You can use the /MINIMUM qualifier with the following option:

- **INTERVAL=*time***

The definition for the option associated with the /MINIMUM qualifier is as follows:

INTERVAL=*time* defines the queue's minimum interval attribute. The value *time* specifies the minimum interval a user can request with the /INTERVAL qualifier.

If the number specified with the /INTERVAL qualifier on the COPY command is less than the queue's minimum interval value, the rate specified with the COPY command is ignored and the queue's minimum interval value is used in its place.

To clear the minimum interval attribute, simply enter the word INTERVAL without typing a number after the equals sign, as in the following example:

```
$ SET QUEUE/MINIMUM=INTERVAL
```

Examples

```
TRANS/DTF> SET QUEUE-  
_TRANS/DTF> /DEFAULT=(INTERVAL=00:45,RETRIES=50) NIGHTLY
```

This SET QUEUE command defines a default interval of 45 minutes and a default retry of 50 for the DTF queue named NIGHTLY.

```
TRANS/DTF> SET QUEUE-  
_TRANS/DTF> /MAXIMUM=(TIME=5-)/DEFAULT=(TIME=23:00:00)
```

This SET QUEUE command defines a maximum time of 5 days and a default time of 23 hours for the default DTF queue named SNADTF\$QUEUE.

6.5 Displaying the Status of File Transfer Jobs

To display the status of all your jobs or a single job identified by a job name, use the TRANSFER/DTF SHOW JOBS command:

```
SHOW JOBS [job-name][/qualifiers]
```

Job names for a file transfer operation are specified with the /NAME qualifier on the TRANSFER/DTF COPY command. By default, the status of all your jobs is displayed.

To display a complete listing of other users' jobs with the /FULL qualifier, you must have group or world privileges. For information about protection privileges, refer to the *DCL Dictionary*.

Each job displayed with the SHOW JOBS command is listed as being in one of the following four states:

QUEUED	A job has not yet started copying. This state occurs when the /AFTER qualifier is specified with the COPY command or if another job is already executing on the DTF queue.
INITIALIZING	A job has started parsing the input and output files but has not yet started copying. This state can last for some time before copying begins. If the job stops at any time, the status automatically changes to PAUSED.
ACTIVE	A job is currently running and copying is in progress.
PAUSED	A job has stopped running because a file transfer operation failed and the job is awaiting retry.

Command Parameter

job-name

Specifies the name of the job whose status you want to display. If this parameter is not specified, information on all your jobs is displayed.

SHOW JOBS Command Qualifiers

SHOW JOBS Qualifier	Default
/ALL	/ALL
/BRIEF	/BRIEF
/FULL	None
/QUEUE= <i>queue-name</i>	/QUEUE=*
/USER= <i>user-name</i>	None

/ALL

Displays the names of all DTF job entries. By default, the SHOW JOB command displays only jobs owned by the current process. This qualifier is mutually exclusive with the /USER qualifier.

/BRIEF

Displays a shortened listing of information about job entries in the queue. When you use the /BRIEF qualifier, the following information is displayed:

- User name
- Job name
- Queue name

/FULL

Displays complete information about all jobs in DTF queues owned by the current process. The information on jobs includes the following:

- The date and time the COPY command was submitted
- The entire COPY command you entered
- The status of the the COPY operation.

/QUEUE=*queue-name*

Specifies a queue name. If you omit this qualifier, jobs on all queues are displayed. You can use wildcard characters (*, %) with this qualifier.

/USER=*user-name*

Displays information about jobs owned by the specified user.

Examples

```
$ TRANSFER/DTF
TRANS/DTF> SHOW JOB/BRIEF
Job 1234 is [state], submitted on 26-NOV-1984 15:50:24 by SMITH
```

With the SHOW JOB/BRIEF command, *state* is either QUEUED, INITIALIZED, ACTIVE, or PAUSED.

```
$ TRANSFER/DTF
TRANS/DTF> SHOW JOB/FULL
Job 1234 is [state], submitted on 20-JUL-1984 15:50:24 by SMITH
```

```
Command:      COPY/NOTIFY BOSTON"SMITH password"::DTFTEST.A USER[SMITH]
Last error:   No error has occurred
Elapsed time  0 01:10:51      Number of Retries      2
Maximum time  3 00:00:00      Maximum Retries        100
```

```
BOSTON"SMITH password"::DQA0:[SMITH]DTFTEST.A;3
Transferred 64 blocks out of 872 blocks for the current file
```

6.6 Displaying OpenVMS/DTF Queue Characteristics

To display the name and characteristics of the queues to which DTF jobs are submitted, use the TRANSFER/DTF SHOW QUEUE command:

```
SHOW QUEUE [queue-name[:]][/qualifiers]
```

Command Parameter

queue-name

Specifies an OpenVMS/DTF queue name. The queue name must be an OpenVMS/DTF queue name. If you do not specify a queue name, DTF displays information about all the OpenVMS/DTF execution queues and the generic queues that regulate OpenVMS/DTF execution queues. For further information about queues, see the SET QUEUE command.

You can use wildcard characters in the *queue-name* parameter. The rules that apply to wildcard characters for file specifications also apply to queue names. The default queue name is an asterisk (*), which displays information about all initialized DTF queues.

You can also use generic and logical queue names with the *queue-name* parameter.

SHOW QUEUE Command Qualifiers

SHOW QUEUE Qualifier	Default
/BRIEF	/BRIEF
/FULL	None

/BRIEF

Displays a shortened listing of information about the specified DTF queues. The following information is displayed:

- Queue name
- Queue status

/FULL

Displays complete information about the specified DTF queues. The following information is displayed:

- Queue name
- Queue settings

Example

```
$ TRANSFER/DTF
TRANS/DTF> SHOW QUEUE/FULL
SNADTF queue SNADTF$QUEUE, on DRAGON::
  /BASE_PRIORITY=4 /OWNER=[SYSTEM] /PROTECTION=(S:E,O:D,G:R,W:W)

Default INTERVAL:    0 00:01:00.00  Minimum INTERVAL:  Unspecified
Default RETRIES:     48                Maximum RETRIES:   Unspecified
Default TIME:        1 00:00:00.00  Maximum TIME:     Unspecified
```

This example shows the output from a SHOW QUEUE/FULL command.

6.7 Displaying the OpenVMS/DTF Software Version Number

To display the current version of the OpenVMS/DTF software, use the TRANSFER/DTF SHOW VERSION command:

```
SHOW VERSION
```

This version number should be listed on all submitted Software Performance Reports (SPRs).

Example

```
$ TRANSFER/DTF
TRANS/DTF> SHOW VERSION
Digital SNA Data Transfer Facility for OpenVMS Vn.n
```

This SHOW VERSION command displays the current version of the OpenVMS/DTF software.

6.8 Canceling File Transfer Jobs

To remove a job from a queue before it executes or to abort a currently running job, use the TRANSFER/DTF CANCEL JOB command:

```
CANCEL JOB job-name[/qualifier]
```

A privileged user can cancel any job that standard OpenVMS protection mechanisms allow. For example, a user with group privileges can cancel a job owned by another person in the same group.

To delete a job from a queue you must have operator (OPER) privileges, execute (E) access to the specified queue, or delete (D) access to the specified job.

Command Parameter

job-name

Specifies the job name that you want to cancel. You can display a list of job names by entering the TRANSFER/DTF SHOW JOBS command.

CANCEL JOB Command Qualifier

CANCEL JOB Qualifier

Default

/QUEUE=*queue-name*]

/QUEUE=SNADTF\$QUEUE

/QUEUE

Specifies the name of the DTF queue in which the job you want to cancel was entered. If the /QUEUE qualifier is not specified, the CANCEL JOB command checks for the job on the SNADTF\$QUEUE queue.

Examples

```
$ TRANSFER/DTF
TRANS/DTF> CANCEL JOB DATA010
```

This CANCEL JOB command removes the job named DATA010 from the DTF default queue (SNADTF\$QUEUE). If this job is waiting to execute, the system deletes it. If the job is already executing, the system aborts it and then deletes it.

```
$ TRANSFER/DTF
TRANS/DTF> CANCEL JOB/QUEUE=NIGHTLY ACCOUNTING
```

This CANCEL JOB command removes the job ACCOUNTING from the DTF queue called NIGHTLY.

6.9 Displaying OpenVMS/DTF Error Messages

Most DTF error messages are RMS error messages that include a secondary numeric status code. To display the text associated with the secondary status code in an RMS error message, use the TRANSFER/DTF SHOW ERROR command:

```
SHOW ERROR status-code
```

You can use this command to help discover the cause of an error when using DTF-supported DCL commands.

Command Parameter

status-code

Specifies the status code for which you are requesting the error message text. The status code is a hexadecimal number.

Note

The *Digital SNA Data Transfer Facility for OpenVMS Problem Solving and Messages* manual contains two appendixes: one which lists the numeric status codes and their abbreviated text and one which lists the full error message along with an explanation and suggested action.

Example

```
$ TRANSFER/DTF
TRANS/DTF> SHOW ERROR %X023A80B2
%SNADTF-E-APPNOTAVA, DTF for IBM is not available (sense code %X'0801')
```

This example displays information about reason code %X023A80B2.

Using the DTF for IBM Panel Interface

This chapter explains how to use the DTF for IBM panel interface to transfer files. It describes all of the fields on the panels and provides examples of what the panels look like. The sections show you how to move from panel to panel in order to copy files.

7.1 The DTF Panel Interface

The DTF panel interface provides a series of panels that prompt you through the process of transferring files between an IBM system and a DECnet node or between two IBM systems.

The panels can be used by CMS and TSO/E Interactive System Productivity Facility (ISPF) users and by CMS interactive users. CMS and TSO/E ISPF users must select an option from an ISPF menu to use the DTF panel interface. If ISPF is not installed, you can use the DTF panel interface from CMS by typing one of the DTF single-line file transfer commands (see Chapter 8) without any parameters.

Note

The DTF panels for ISPF and non-ISPF users are identical except for two characteristics:

- The ISPF panel sequence has an Operation Selection panel from which you can select one of three options. After you select an option DTF displays one of three operation panels. The non-ISPF panel sequence does not have the Operation Selection panel; instead DTF immediately displays one of the three operation panels. The panel DTF displays depends on the DTF single-line command you entered (DTFRECV, DTFRESUM, or DTFSEND).

- The three non-ISPF operation panels have two extra fields that allow you to specify the information that ISPF users supply with the Operation Selection panel. These fields allow you to specify the network subsystem name and the server node name.
-

7.2 The Operation Selection Panel

Figure 7-1 Operation Selection Panel

```
----- DATA TRANSFER FACILITY/ISPF - OPERATION SELECTION -----  
COMMAND ===> █  
  
File transfer commands:  
S SEND local file to remote system  
R RECEIVE file from remote system  
C RESUME previously failed transfer  
  
DTF SERVICES provided by:  
NETWORK SUBSYSTEM ===> DTFNET  
SERVER NODE          ===>
```

LKG-8307-93R

The Operation Selection panel (see Figure 7-1) is the first panel that you see when you select the DTF option from the ISPF main menu. This panel does not appear when you use the non-ISPF panel interface; you will immediately see one of the operation panels. Use the Operation Selection panel to select one of the three file transfer operations:

- You can send a local file from the system you are on to a file on a remote system.
- You can receive a file from a remote system and store it in a local file on your system.

- You can resume a recoverable file transfer operation that was previously interrupted.

To begin an operation, select one of the menu options and press the ENTER key. This will cause DTF to display one of the three operation panels. These panels are described in Section 7.3, Section 7.4, and Section 7.5.

The Operation Selection panel lets you choose the network subsystem and server node that you want to use to transfer files. DTF supplies defaults for these names, but you can type over the names that are provided.

DTF SERVICES provided by:

NETWORK SUBSYSTEM

Selects the Network Manager component which processes the transfer request. For VM systems, enter a 1- to 8-character Network Manager machine name. For MVS systems, enter a 1- to 4-character subsystem ID.

SERVER NODE

Selects the server which processes the transfer request. Type in the DECnet server node name that you wish to use to perform the transfer.

Note

If you specify a value for the SERVER NODE field, there must be an active server session that uses the same server name. If there is no such server session, DTF rejects the transfer request. If you do not fill in the SERVER NODE field, DTF selects a server node from the pool of active server sessions. DTF attempts to select a server node with the same name as the client node from the request. If the request does not specify a client node or if the client node is not a server node, DTF selects the first available server node.

7.3 The Send to Remote Panel

Figure 7-2 Send to Remote Panel

```
----- DTF/ISPF - SEND TO REMOTE -----
COMMAND ===>

SEND this local file:
FILENAME  ===> █
OWNER     ===>
DEVICE ADDR: ===>      PASSWORD ===>
OPTIONS?  ===>          (Y/N)

TO this USER at this remote NODE:
NODENAME  ===>
USERID    ===>
PASSWORD  ===>
OPTIONS   ===>          (Y/N)

STORE on your local system AS this file name:
FILENAME  ===>
DISP      ===>          (NEW/OLD/REP)
OPTIONS   ===>          (Y/N)

USING these options:
RECOVERABLE ===>          (Y/N) CHECKPOINT ===>
TRANSLATE   ===>          (Y/N) EXTENDED   ===>          (Y/N)
POSTPROCESS ===>          (Y/N)
BATCH SUB.  ===>          (Y/N)
```

LKG-8306-93R

DTF displays the Send to Remote panel (Figure 7-2 shows the VM version) after you select the **SEND** local file to remote system menu option from the Operation Selection panel or after you enter the DTFSEND command (see Section 8.1). The Send to Remote panel prompts you for all the information you need to transfer a file from your system to a remote system.

You enter information on this panel in four categories. The following pages describe each of the fields on this panel.

You can press **PF3** to return to the Operation Selection panel without initiating the send operation.

SEND this local file:

FILENAME

This required field specifies the name of the file you wish to copy from your system.

OWNER

This field specifies the owner ID of the minidisk containing the local file. This field is optional if you currently have the minidisk linked and accessed.

DEVICE ADDR.

This field specifies the device address of the minidisk containing the local file. This field is optional if you currently have the minidisk linked and accessed.

PASSWORD

This value is the read password for the minidisk containing the local file. This field is optional if you currently have the minidisk linked and accessed. For more information on minidisk passwords, see the /MDxPASSWORD qualifier description in Section 2.3.3.

OPTIONS?

There may be additional information about the local input file that you can specify. To see the panel for entering this information, type the letter Y in the OPTIONS field and press the ENTER key. These panels are described in Section 7.8 and Section 7.9.

TO this USER at this remote NODE:

NODENAME

Specify the name of the node you are copying the file to in this field. You can specify either a DECnet node name or an abbreviation that names an IBM system. See your system programmer for a list of the DTF system abbreviations used at your site. If you leave this field blank, DTF displays the Remote Node Options panel (see Section 7.7). The Remote Node Options panel always precedes all other panels because it determines the operating system at the remote node. If an abbreviation will not fit in the space provided, leave the field blank and specify the abbreviation on the Remote Node Options panel.

USERID

Specifies the user ID that should be used to access the output file on the remote system. See Section 4.2 for information on user ID syntax.

Note

If the remote client node is running Digital UNIX or ULTRIX, you must enter the user ID with the proper case.

PASSWORD

Specifies the password of the user ID on the remote system. See Section 4.2 for information on password syntax.

Note

If the remote client node is running Digital UNIX, ULTRIX, you must enter the password with the proper case.

OPTIONS

There may be additional information about the remote node that you can specify. To see the panel for entering this information, type the letter Y in the OPTIONS field and press the ENTER key. This panel is described in Section 7.7.

STORE on the remote node AS this file name:

FILENAME

This is the name that the file you are copying will have on the remote system after the transfer completes. You must supply this file name. (See Chapter 2 for more information on entering IBM file names when using DTF.)

Note

If the remote client node is running Digital UNIX or ULTRIX, you must enter the file name with the proper case.

DISP

This field specifies the disposition of the new file. You can choose one of the following three values:

1. NEW
2. OLD
3. REP

NEW creates a new output file. If a file with the same name already exists, DTF does not copy the file. This is the default.

OLD creates a new output file that overlays an existing file of the same name. If the file does not exist, DTF issues an error message.

REP creates a new output file even if a file with the same name already exists.

OPTIONS

There may be additional information about the remote output file that you can specify. To see the panel for entering this information, type the letter Y in the OPTIONS field and press the ENTER key. These panels are described in Section 7.10 and Section 7.11.

USING these options:

RECOVERABLE

This field controls whether the transfer runs in recoverable mode or nonrecoverable mode. Chapter 1 provides additional information about recoverable copy. The default action is set up when you install DTF.

You may select one of the following options:

- Y** The transfer takes place in recoverable mode.
- N** The transfer operation takes place in nonrecoverable mode.

Note

If you specify the RECOVERABLE field as Y, but the DECnet-VAX node does not have the DTF utilities software installed (DECnet client nodes running MS-DOS, OS/2, Digital UNIX, or ULTRIX do not have recoverable copy capability), DTF rejects the transfer request.

CHECKPOINT

This field allows you to set the number of logical records between checkpoints. The number of logical records can be from 1 to 65535.

Note

- If you do not specify a value for this field, the TRANSFER/DTF on the client node uses a default value of 1000.

- The CHECKPOINT field is valid only if you specify the RECOVERABLE field as YES. If you specify the RECOVERABLE field as NO, DTF does not permit you to enter a value in the CHECKPOINT field.
-

TRANSLATE

Type N in this field to disable the translation of data from DMCS to EBCDIC. No translation takes place if you copy a file from one IBM system to another.

EXTENDED

This field lets you use the Data Translation Options panel to specify data translation options other than standard DMCS to EBCDIC translation. Typing Y lets you see this panel. This panel is described in Section 7.13.

POSTPROCESS

This field lets you use the Post-Processing Options panel to specify a file at the remote node that should be submitted to the remote node's batch subsystem after the transfer completes. Typing Y lets you see this panel. This panel is described in Section 7.14.

BATCH SUB.

This field lets you use the Batch Options panel to specify job control information needed to create a batch job for the file transfer operation. Optionally, this job can be directly submitted after all panels have been completed. Typing Y lets you see the Batch Options panel. This panel is described in Section 7.15.

7.4 The Receive from Remote Panel

Figure 7-3 Receive from Remote Panel

```
COMMAND ===>

RECEIVE this remote file:
FILENAME   ===> █
OPTIONS?   ===>

FROM this USER at this remote NODE:
NODENAME   ===>
USERID     ===>
PASSWORD   ===>
OPTIONS    ===>          (Y/N)

STORE on your local system AS this file name:
FILENAME   ===>
DISP       ===>          (NEW/OLD/REP)
OWNER      ===>          DEVICE ADDR. ===>          PASSWORD ===>
OPTIONS    ===>          (Y/N)

USING these options:
RECOVERABLE ===>          (Y/N) CHECKPOINT ===>
TRANSLATE   ===>          (Y/N) EXTENDED   ===>          (Y/N)
POSTPROCESS ===>          (Y/N)
```

LKG-8305-93R

DTF displays the Receive from Remote panel (Figure 7-3 shows the VM version) after you select the **RECEIVE** file from remote system menu option from the Operation Selection panel or after you enter the DTFRECV command (see Section 8.2). The Receive from Remote panel prompts you for all of the information needed to transfer a file from a remote system to your system.

Similar to the Send to Remote panel, there are four categories of information that you enter on this panel. The following pages describe the fields on this panel.

You can press **PF3** to return to the Operation Selection panel without initiating the receive operation.

RECEIVE this remote file:

FILENAME

This field contains the name of the file you are copying from the remote system. You must enter a file name in this field. (See Chapter 2 for more information on entering file names when using DTF.)

Note

If the remote client node is running Digital UNIX or ULTRIX, you must enter the file name with the proper case.

OPTIONS?

There may be additional information about the remote input file that you can enter. To see the panel for entering this information, type the letter Y in the OPTIONS field and press the ENTER key. These panels are described in Section 7.8 and Section 7.9.

FROM this USER at this remote NODE:

NODENAME

Specify the name of the node you are copying the file from in this field. You can specify either a DECnet node name or an abbreviation that names an IBM system. See your system programmer for a list of the DTF system abbreviations used at your site. If you leave this field blank, DTF displays the Remote Node Options panel (see Section 7.7). The Remote Node Options panel always precedes all other panels because it determines the operating system at the remote node. If an abbreviation will not fit in the space provided, leave the field blank and specify the abbreviation on the Remote Node Options panel.

USERID

Specifies the user ID that should be used to access the input file on the remote system. See Section 4.2 for information on user ID syntax.

Note

If the remote client node is running Digital UNIX or ULTRIX, you must enter the user ID with the proper case.

PASSWORD

Specifies the password of the user ID on the remote system. See Section 4.2 for information on password syntax.

Note

If the remote client node is running Digital UNIX or ULTRIX, you must enter the password with the proper case.

OPTIONS

There may be additional information about the remote node that you can specify. To see the panel for entering this information, type the letter Y in the OPTIONS field and press the ENTER key. This panel is described in Section 7.7.

STORE on your local system AS this file name:

FILENAME

This required field specifies the name the file you are copying from the remote system will have on your system after the transfer completes.

DISP

This field specifies the disposition of the new file. You can choose one of the following three values:

1. NEW
2. OLD
3. REP

NEW creates a new output file. If a file with the same name already exists, DTF does not copy the file. This is the default.

OLD creates a new output file that overlays an existing file of the same name. If the file does not exist, DTF issues an error message.

REP creates a new output file even if a file with the same name already exists.

OWNER

This field specifies the owner ID of the minidisk that will contain the output file. This field is optional if you currently have the minidisk linked and accessed.

DEVICE ADDR.

This field specifies the device address of the minidisk that will contain the output file. This field is optional if you currently have the minidisk linked and accessed.

PASSWORD VM

This field specifies the multiwrite password of the minidisk that will contain the output file. This field is optional if you currently have the minidisk linked and accessed. For more information on minidisk passwords, see the /MDxPASSWORD qualifier description in Section 2.3.3.

OPTIONS

There may be additional information about the remote output file that you can specify. To see the panel for entering this information, type the letter Y in the OPTIONS field and press the ENTER key. These panels are described in Section 7.10 and Section 7.11.

USING these options:**RECOVERABLE**

Controls whether the transfer runs in recoverable mode or nonrecoverable mode. Chapter 1 provides additional information about recoverable copy.

You may select one of the following options:

- Y** The transfer executes in recoverable mode.
- N** The transfer operation executes in nonrecoverable mode.

Note

If you specify the RECOVERABLE field as Y, but the DECnet-VAX node does not have the DTF utilities software installed (DECnet client nodes running MS-DOS, OS/2, Digital UNIX, or ULTRIX do not have recoverable copy capability), DTF rejects the transfer request.

CHECKPOINT

This field allows you to set the number of logical records between checkpoints. The number of logical records can be from 1 to 65535.

Note

If you do not specify a value for this field, the TRANSFER/DTF on the client node uses a default value, which is defined during the DTF installation. The CHECKPOINT field is valid only if you specify the RECOVERABLE field as YES. If you specify the RECOVERABLE field as NO, DTF does not permit you to enter a value in the CHECKPOINT field.

TRANSLATE

Type N in this field to disable the translation of data from DMCS to EBCDIC. No translation takes place if you copy a file from one IBM system to another.

EXTENDED

This field lets you use the Data Translation Options panel to specify data translation options other than standard DMCS to EBCDIC translation. Typing Y lets you see this panel. This panel is described in Section 7.13.

POSTPROCESS

This field lets you use the Post-Processing Options panel to specify a file at the remote node that should be submitted to the remote node's batch subsystem after the transfer completes. Typing Y lets you see this panel. This panel is described in Section 7.14.

BATCH SUB.

This field lets you use the Batch Options panel to specify job control information needed to create a batch job for the file transfer operation. Optionally, this job can be directly submitted after all panels have been completed. Typing Y lets you see the Batch Options panel. This panel is described in Section 7.15.

7.5 The Resume Previously Failed Transfer Panel

Figure 7-4 Resume Previously Failed Transfer Panel

```
----- DTF/ISPF - RESUME PREVIOUSLY FAILED TRANSFER -----
COMMAND ===>

  Fill in the required fields and press ENTER to continue

RESUME this transfer:
RETRY NUMBER ===> █

WITH this remote node:
NODENAME      ===>
USERID        ===>
PASSWORD      ===>
OPTIONS       ===>          (Y/N)

USING these options:
BATCH SUB.    ===>          (Y/N)
```

LKG-8304-93R

DTF displays the Resume Previously Failed Transfer panel (see Figure 7-4) after you select the **RESUME** previously failed transfer menu option from the Operation Selection panel or after you enter the DTFRESUM command (see Section 8.3).

You can press **PF3** to return to the Operation Selection panel without initiating the resume operation.

The following pages describe the fields on this panel.

RESUME this transfer:

RETRY NUMBER

The five-digit number you see in message DTF00621 when you submit the original request.

Note

If your site uses TSO/E or CMS, the retry number is automatically set whenever a transfer request executes with the RECOVERABLE COPY option set to YES.

WITH this remote node:

NODENAME

Specifies the name of the client node that processed the original transfer request.

- If your site uses ISPF, this is automatically set whenever a transfer request has failed with a recoverable error and the RECOVERABLE option was set to YES.
- Do not make an entry in the NODENAME field if the original transfer request involved two IBM files.

USERID

Specifies the user ID specified for the DECnet file involved in the original transfer request.

- If your site uses ISPF, this is automatically set whenever a transfer request has failed with a recoverable error and the RECOVERABLE option set to YES.
- Do not make an entry in the USERID field in the following situations:
 - If the original transfer request involved two IBM files
 - If the original transfer request involved a DECnet file that had no user ID associated with it.

PASSWORD

Specifies the password associated with the DECnet file involved in the original transfer request.

- If your site uses ISPF, this is automatically set whenever a transfer request has failed with a recoverable error and the RECOVERABLE option set to YES. However, the value is not retained if the current ISPF dialog ends.
- Do not make an entry in the PASSWORD field in the following situations:
 - If the original request involved two IBM files
 - If the original request involved a DECnet file that did not have a password associated with it

OPTIONS

There may be additional information about the remote node that you can specify. To see the panel for entering this information, type the letter Y in the OPTIONS field and press the ENTER key.

USING these options:**BATCH SUB.**

This field lets you use the Batch Options panel to specify job control information needed to create a batch job for the file transfer operation. Optionally, this job can be directly submitted after all panels have been completed. Typing Y lets you see the Batch Options panel. This panel is described in Section 7.15.

7.6 The Transfer Execution Panel

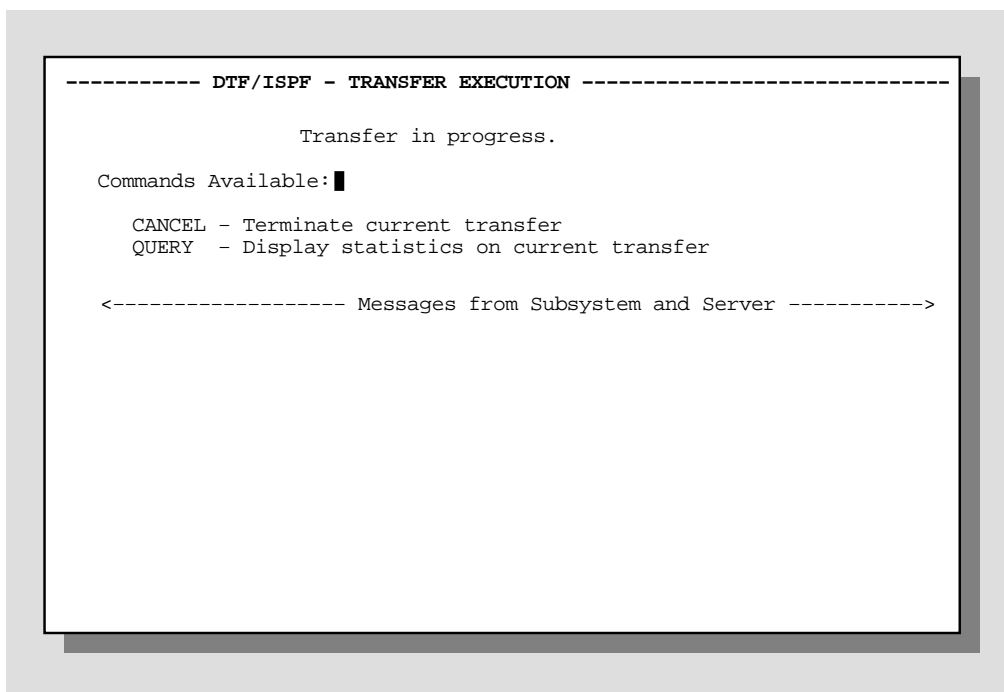
Figure 7-5 Transfer Execution Panel (MVS Systems)

```
----- DTF/ISPF - TRANSFER EXECUTION -----  
  
                Transfer in progress.  
  
SNA terminals: Press the ATTN key  
Non-SNA terminals: Press RESET then the PA1 key  
  
Commands Available: █  
  C = CANCEL current transfer  
  Q = QUERY statistics on current transfer  
  W = RETURN to WAIT state from Query auto update.  
  
<----- Messages from Subsystem and Server ----->
```

LKG-8303-93R

DTF displays the Transfer Execution panel (see Figure 7-5 and Figure 7-6) after you initiate a transfer using one of the operation panels. All usage notes for the two panels follow Figure 7-6.

Figure 7-6 Transfer Execution Panel (VM Systems)



LKG-8302-93R

[MVS] You can cancel a transfer that is in progress at any time. The method used to cancel a transfer depends on the terminal you are using. On non-SNA terminals, press **[Reset]**, then **[PA1]**, and then type **C** followed by **[Enter]**. On SNA terminals, press **[Attn]** and then type **C** followed by **[Enter]**.

[MVS] You can request a status report for transfer in progress at any time. The method used to get a status report for a transfer depends on the terminal you are using. On SNA terminals, press **[Reset]**, then **[PA1]**, and then type **Q** followed by **[Enter]**. On non-SNA terminals, press **[Attn]** and then type **Q** followed by **[Enter]**. After typing **Q** the system provides a continuously updated display of the transfer in progress. To return to the wait state from SNA terminals, press **[Reset]**, then **[PA1]**, and then type **W** followed by **[Enter]**. To return to the wait state from non-SNA terminals, press **[Attn]** then type **W** followed by **[Enter]**.

On VM systems, you can cancel a transfer that is in progress at any time by typing the command **CANCEL** and pressing **[Enter]**. You can request a status report for a transfer in progress by typing the command **QUERY** and pressing **[Enter]**.

The QUERY command results in a display similar to the following:

```
DTFM093I <----- REQUEST DATA ----->
DTFM097I DTF WORK UNITS: ACCESS(00039) RETRY NUMBER(      )
DTFM098I DTF SERVICES: NETMGR(DTF3      ) V( N/A ) SERVER(SMAUG      ) V(3.1.0)
DTFM099I DTF SERVICES: CLIENT(SMAUG      ) V(3.1.0)
DTFM093I <----- TRANSFER DATA ----->
DTFM094I NET PATH : LU(H01AA073) ACCESS(00040)
DTFM096I VTAM COUNTS: TYPE(SEND ) TOTAL(0000071) BYTE(0005480) MBYTE(0000000)
DTFM096I VTAM COUNTS: TYPE(RECV.) TOTAL(0000081) BYTE(0001263) MBYTE(0000000)
DTFM093I <----- FILE DATA ----->
DTFM100I FILE ACCESS: DTF.TEST.PDS(MEMBER)
DTFM096I FILE COUNTS: TYPE(READ ) TOTAL(0000050) BYTE(0006160) MBYTE(0000000)
DTFM096I FILE COUNTS: TYPE(WRITE) TOTAL(0000000) BYTE(0000000) MBYTE(0000000)
```

The message fields are explained in Appendix A of the *Digital SNA Data Transfer Facility for OpenVMS Problem Solving and Messages* manual.

7.7 The Remote Node Options Panel

Figure 7-7 Remote Node Options Panel

```
----- DTF/ISPF - REMOTE NODE OPTIONS -----
COMMAND ===>

Fill in the required fields and press ENTER to continue

NODE NAME (if more space is required);
  EXTENDED NODE NAME ===>█

SYNTAX of remote file names follows the rules of:
  NODE TYPE          ===>

ACCESS of remote node provided by this SERVER or GATEWAY:
  NODENAME           ===>
  SERVER ACCOUNT     ===>
  SERVER PASSWORD    ===>
```

LKG-8301-93R

DTF displays the Remote Node Options panel (see Figure 7-7) if you type Y in the OPTIONS field for a remote node in the Send to Remote or Receive from Remote panel.

This panel contains information about the remote node that DTF uses to perform transfers. Fill in the appropriate fields and press the ENTER key. If you did not specify any other options, the transfer begins immediately.

You can press **PF3** to return to the operation panel that invoked this panel without initiating the operation.

The following pages describe the fields on this panel.

NODE NAME (if more space is required):

EXTENDED NODE NAME

Specify the name of the node you are copying the file to in this field. You can specify either a DECnet node name or an abbreviation that names an IBM system. See your system programmer for a list of the DTF system abbreviations used at your site. You can enter node names that are up to 255 characters long. If you made an entry on the operation panel that invoked this panel, the entry will be displayed on this panel.

SYNTAX of remote file names follows the rules of:

NODE TYPE

This field lets you indicate the operating system on the remote node. This allows DTF to check that the file name you specify is syntactically correct for that operating system.

DTF fills in this field only if the name you entered in the NODENAME field on the previous panel matches an entry in the abbreviations list. Otherwise, DTF displays the value **UNKNOWN**. You can override this value by typing in a different operating system keyword. The valid operating system names are: MVS, VM, OpenVMS, Digital UNIX, ULTRIX, RSX, MS-DOS, OS/2, and TCP.

Note

If you specified Y in the OPTIONS field for a remote file, then you must enter a node type value in order to see the correct file options panel.

ACCESS to remote node provided by this SERVER or GATEWAY:

NODENAME

This is the node name of the OpenVMS/DTF server node providing access to the IBM DTF client system or the DECnet/Internet Gateway node providing access to the TCP network. You can enter a node name only if the node type was specified as MVS, VM, or TCP.

SERVER ACCOUNT

The account on the OpenVMS/DTF server node that is used to perform transfers to the IBM DTF client. See your OpenVMS/DTF server manager for a list of accounts and IBM systems available on the OpenVMS/DTF server. No account is necessary to connect to a TCP/IP network.

SERVER PASSWORD

This is the password for the account on the OpenVMS/DTF server node. Not all OpenVMS/DTF server accounts use passwords; see your OpenVMS/DTF server manager. No password is necessary to connect to a TCP/IP network.

7.8 The MVS Input File Options Panel

Figure 7-8 MVS Input File Options Panel

```
COMMAND ===>

Fill in the required fields and press ENTER to continue

FILE options:                                TAPE options:
MIXED CASE NAME  ===> █                      TAPE LABEL      ===>
VOLUME SERIAL   ===>                          FILE NUMBER     ===>
UNIT            ===>                          DENSITY         ===>

DTF options:
FILE DEFINITION ===>
```

LKG-8300-93R

DTF displays the MVS Input File Options panel (see Figure 7-8) if you type Y in the OPTIONS field for an MVS input file in the Send to Remote or Receive from Remote panel.

You can press **PF3** to return to the operation panel that invoked this panel without initiating the operation.

The following pages explain the fields on this panel.

FILE options:

MIXED CASE NAME

This field specifies whether DTF should accept the case in which the file name was entered. A value of Y will cause DTF to open the file in a case sensitive fashion. N or blank will default to uppercase.

Note

This option has no effect unless NODEPARMS LOWERCASECREATE is set to ON.

See the /CASE qualifier in Chapter 2 for information on MIXED CASE.

VOLUME SERIAL

Specifies the volume serial number that contains the IBM input file. This number can be up to 6 characters long. If the input file you are trying to copy is not cataloged, or you want to avoid a catalog search to find the file, you should specify both the volume name and the unit name.

UNIT

Specifies a generic device name that describes the device type that holds the IBM input file. The length can be 1 to 8 characters. If the input file you are trying to copy is not cataloged, or you want to avoid a catalog search to find the file, you should specify both the volume name and the unit name.

TAPE options:**TAPE LABEL**

Specifies the format of the tape label on the tape that is read. You can specify one of the following tape labels:

- AL** ANSI V1 labels or ISO/ANSI/FIPS V3 labels
- SL** Standard labels

- Tape restrictions may apply.
- You should fill in this field only if the input file resides on a tape.
- If the input file resides on disk, DTF ignores this field.
- If you do not specify this field and the input file resides on tape, DTF assumes that standard tape labels are used.

FILE NUMBER

Specifies a sequential file number on tape between 1 and 9999.

- Tape restrictions may apply.
- If the input file resides on disk, DTF ignores this field.
- If you do not specify this field and the input file resides on tape, DTF assumes that the FILE NUMBER is 1.

DENSITY

Specifies the density of the tape volume from which the input file is read. Accepted values are 1600 or 6250.

- Tape restrictions may apply.
- If the input file is not on tape, DTF ignores this field.
- If you do not specify this field and the input file is on tape, DTF assumes the highest density allowed for the tape unit.

DTF options:**FILE DEFINITION**

Specifies the name of an entry in the DTF file definition database on the OpenVMS/DTF server node. Chapter 2 explains file definition records.

7.9 The VM Input File Options Panel

Figure 7-9 VM Input File Options Panel

```
COMMAND ===>

Fill in the required fields and press ENTER to continue

MINIDISK information:                TAPE options:
DEVICE OWNER      ===> █              TAPE LABEL      ===>
DEVICE ADDRESS    ===>                FILE NUMBER     ===>
PASSWORD          ===>                DENSITY         ===>

FILE options:
VSAM              ===>                (Y/N)
TAPE UNIT         ===>
MIXED CASE NAME  ===>                (Y/N)

DTF options:
FILE DEFINITION  ===>
```

LKG-8298-93R

DTF displays the VM Input File Options panel (see Figure 7-9) if you type Y in the OPTIONS field for a VM input file in the Send to Remote or Receive from Remote panel.

You can press **[PF3]** to return to the operation panel that invoked this panel without initiating the operation.

The following pages describe the fields on this panel.

MINIDISK information:

DEVICE OWNER

This field specifies the owner of the minidisk containing the input file. For a local input file, this field is optional if you have the minidisk linked and accessed. For a remote input file, see the description of the /OWNERID qualifier in Chapter 2.

DEVICE ADDRESS

This field specifies the device address of the minidisk containing the input file. For a local input file, this field is optional if you have the minidisk linked and accessed. For a remote input file, see the description of the /MDADDRESS qualifier in Chapter 2.

PASSWORD

This field specifies the read password of the minidisk containing the input file. For a local input file, this field is optional if you have the minidisk linked and accessed. For more information on minidisk passwords, see the /MDxPASSWORD qualifier description in Section 2.3.3.

FILE options:**VSAM**

If the input file is VSAM format, you must indicate this by typing a Y in this field.

TAPE UNIT

If the input file is on tape, you must indicate this by specifying a tape unit in this field. The TAPE UNIT must match what is defined in the DTFUSER PARMS for NODEPARMS TAPEUNITS. An IBM site can use additional or different unit names. Check with the system programmer of the site where the input file resides to get the unit names in use at the site.

MIXED CASE NAME

This field specifies whether DTF should accept the case in which the file name was entered. A value of Y will cause DTF to open the file in a case sensitive fashion. N or blank will default to uppercase.

Note

This option has no effect unless NODEPARMS LOWERCASECREATE is set to ON.

See the /CASE qualifier in Chapter 2 for information on MIXED CASE.

TAPE options:**TAPE LABEL**

Specifies the format of the tape label on the tape that is read. You can specify one of the following tape labels:

AL ANSI V1 labels or ISO/ANSI/FIPS V3 labels

SL Standard labels

- Tape restrictions may apply.
- You should fill in this field only if the input file resides on a tape.
- If the input file resides on disk, DTF ignores this field.
- If you do not specify this field and the input file resides on tape, DTF assumes that standard tape labels are used.

FILE NUMBER

Specifies a sequential file number on tape between 1 and 9999.

- Tape restrictions may apply.
- If the input file resides on disk, DTF ignores this field.
- If you do not specify this field and the input file resides on tape, DTF assumes that the FILE NUMBER is 1.

DENSITY

Specifies the density of the tape volume from which the input file is read. Accepted values are 1600 or 6250.

- Tape restrictions may apply.
- If the input file is not on tape, DTF ignores this field.
- If you do not specify this field and the input file is on tape, DTF assumes the highest density allowed for the tape unit.

DTF options:

FILE DEFINITION

Specifies the name of an entry in the DTF file definition database on the OpenVMS/DTF server node. Chapter 2 explains file definition records.

7.10 The MVS Output File Options Panel

Figure 7-10 MVS Output File Options Panel

```
COMMAND ==>>

Fill in the required fields and press ENTER to continue

FILE options:
MIXED CASE NAME ==>>█ (Y/N)
VOLUME SERIAL   ==>>
UNIT            ==>>
CATALOG         ==>>
VSAM            ==>>
NULL RECORDS   ==>>
PRIMARY ALLOC  ==>>
SECONDARY ALLOC ==>>

DTF options:
FILE DEFINITION ==>>

TAPE options:
TAPE LABEL     ==>>
FILE NUMBER    ==>>
DENSITY        ==>>
(Y/N)
(Y/N)
(REJECT/IGNORE/SPACE/NONE)
```

LKG-8299-93R

DTF displays the MVS Output File Options panel (see Figure 7-10) if you type Y in the OPTIONS field for an MVS output file in the Send to Remote or Receive from Remote panel.

You can press **PF3** to return to the operation panel that invoked this panel without initiating the operation.

The following pages describe the fields on this panel.

FILE options:

MIXED CASE NAME

This field specifies whether DTF should accept the case in which the file name was entered. A value of Y will cause DTF to open the file in a case sensitive fashion. N or blank will default to uppercase.

Note

This option has no effect unless NODEPARMS LOWERCASECREATE is set to ON.

See the /CASE qualifier in Chapter 2 for information on MIXED CASE.

VOLUME SERIAL

Specifies the volume serial number that will contain the output file.

UNIT

Specifies a generic device name that describes the device type that holds the IBM output file. Length: 1 to 8 characters.

CATALOG

The MVS system which creates the output file provides the default for this. Type the letter N in this field if you do not want to catalog this file. See the description of the /CATALOG qualifier in Chapter 2 for more information about cataloging files.

NULL RECORDS

This keyword indicates how DTF handles null records (length = 0) in an IBM environment. Since the CMS file system does not support null length records, DTF cannot write them to the indicated device. You can enter one of the following four values:

- REJECT returns an error message.
- IGNORE ignores all null records. The output file may have fewer records and the null records will not be restored if DTF is used to transfer the file from the VM system.
- SPACE inserts a 1-byte record in place of the zero length record. The one byte is set to an EBCDIC space character.
- NONE accepts the record as is in an MVS environment. In a VM environment, an error will be given if a null record is encountered.

See the /NULL qualifier in Chapter 2 for information on /NULL.

PRIMARY ALLOCATION

A 5-digit number specifying the number of 512-byte blocks that DTF will initially allocate when creating a file on the IBM system.

See the /ALLOCATION qualifier in Chapter 2 for information on ALLOCATION.

SECONDARY ALLOCATION

A 5-digit number specifying the number of 512-byte blocks that DTF will use to extend the file if the initial allocation is insufficient to hold the transferred file. See the /SECONDARY_ALLOCATION qualifier in Chapter 2 for information on SECONDARY_ALLOCATION.

VSAM

If the output file is VSAM format, you must indicate this by typing a Y in this field.

TAPE options:**TAPE LABEL**

Specifies the format of the tape labels to be used for the output tape. You can specify one of the following tape labels:

- AL** ANSI V1 labels or ISO/ANSI/FIPS V3 labels
- BLP** Bypass label processing
- NL** Non-labeled
- SL** Standard labels

- Section 3.3.5 describes additional restrictions that may apply for tape-resident files.
- You should fill in this field only if the output file resides on a tape.
- If the output file resides on disk, DTF ignores this field.
- If you do not specify this field and the output file resides on tape, DTF assumes that standard tape labels (SL) are used.

FILE NUMBER

Specifies a sequential file number on tape between 1 and 9999.

- Section 3.3.5 describes additional restrictions that may apply for tape-resident files.
- If the output file resides on disk, DTF ignores this field.
- If you do not specify this field and the output file resides on tape, DTF assumes that the FILE NUMBER is 1.

DENSITY

Specifies the density of the tape volume where the output file will be written. Accepted values are 1600 or 6250.

- Section 3.3.5 describes additional restrictions that may apply for tape-resident files.
- If the output file is not on tape, DTF ignores this field.
- If you do not specify this field and the output file is on tape, DTF assumes the highest density allowed for the tape unit.

DTF options:

FILE DEFINITION

Specifies the name of an entry in the DTF file definition database on the OpenVMS/DTF server node. Chapter 2 explains file definition records.

7.11 The VM Output File Options Panel

Figure 7–11 VM Output File Options Panel

```
COMMAND ===>

Fill in the required fields and press ENTER to continue

MINIDISK information:                TAPE options:
DEVICE OWNER      ===>                TAPE LABEL      ===>
DEVICE ADDRESS    ===>                FILE NUMBER     ===>
PASSWORD          ===>                DENSITY         ===>

FILE options:
VSAM              ===>                (Y/N)
TAPE UNIT         ===>
NULL RECORDS     ===>                (REJECT/IGNORE/SPACE/NONE)
MIXED CASE NAME  ===>                (Y/N)
PRIMARY ALLOC    ===>
SECONDARY ALLOC  ===>

DTF options:
FILE DEFINITION  ===>
```

LKG-8297-93R

DTF displays the VM Output File Options panel (see Figure 7–11) if you type Y in the OPTIONS field for a VM output file in the Send to Remote or Receive from Remote panel.

You can press **PF3** to return to the operation panel that invoked this panel without initiating the operation.

The following pages describe each field on this panel.

MINIDISK information:

DEVICE OWNER

This field specifies the owner of the minidisk that will contain the output file. For a local output file, this field is optional if you have the minidisk linked and accessed. For a remote output file, see the description of the /OWNER_ID qualifier in Chapter 2.

DEVICE ADDRESS

This field specifies the device address of the minidisk that will contain the output file. For a local output file, this field is optional if you have the minidisk linked and accessed. For a remote output file, see the description of the /MDADDRESS qualifier in Chapter 2.

PASSWORD

This field specifies the read password of the minidisk that will contain the output file. For a local output file, this field is optional if you have the minidisk linked and accessed. For more information on minidisk passwords, see the /MDxPASSWORD qualifier description in Section 2.3.3.

FILE options:**VSAM**

If the output file is in VSAM format, you must indicate this by typing a Y in this field.

TAPE UNIT

If the output file is being written to tape, you must indicate this by specifying a tape unit in this field. The TAPE UNIT must match what is defined in the DTFUSER PARMS for NODEPARMS TAPEUNITS. An IBM site can use additional or different unit names. Check with the system programmer of the site where the output file will reside to get the unit names in use at the site.

NULL RECORDS

This keyword indicates how DTF handles null records (length = 0) in an IBM environment. Since the CMS file system does not support null length records, DTF cannot write them to the indicated device. You can enter one of the following four values:

- REJECT returns an error message.
- IGNORE ignores all null records. The output file may have fewer records and the null records will not be restored if DTF is used to transfer the file from the VM system.
- SPACE inserts a 1-byte record in place of the zero length record. The one byte is set to an EBCDIC space character.
- NONE accepts the record as is in an MVS environment. In a VM environment, an error will be given if a null record is encountered.

See the /NULL qualifier in Chapter 2 for information on /NULL.

MIXED CASE NAME

This field specifies whether DTF should accept the case in which the file name was entered. A value of Y will cause DTF to open the file in a case sensitive fashion. N or blank will default to uppercase.

Note

This option has no effect unless NODEPARMS LOWERCASECREATE is set to ON.

See the /CASE qualifier in Chapter 2 for information on MIXED CASE.

PRIMARY ALLOCATION

A 5-digit number specifying the number of 512-byte blocks that DTF will initially allocate when creating a file on the IBM system.

See the /ALLOCATION qualifier in Chapter 2 for information on ALLOCATION.

SECONDARY ALLOCATION

A 5-digit number specifying the number of 512-byte blocks that DTF will use to extend the file if the initial allocation is insufficient to hold the transferred file. See the /SECONDARY_ALLOCATION qualifier in Chapter 2 for information on SECONDARY_ALLOCATION.

TAPE options:

TAPE LABEL

Specifies the format of the tape labels to be used for the output tape. You can specify one of the following tape labels:

- AL** ANSI V1 labels or ISO/ANSI/FIPS V3 labels
- BLP** Bypass label processing
- NL** Non-labeled
- SL** Standard labels

- Section 3.3.5 describes additional restrictions that may apply for tape-resident files.
- You should fill in this field only if the output file resides on a tape.
- If the output file resides on disk, DTF ignores this field.

- If you do not specify this field and the output file resides on tape, DTF assumes that standard tape labels (SL) are used.

FILE NUMBER

Specifies a sequential file number on tape between 1 and 9999.

- Section 3.3.5 describes additional restrictions that may apply for tape-resident files.
- If the output file resides on disk, DTF ignores this field.
- If you do not specify this field and the output file resides on tape, DTF assumes that the FILE NUMBER is 1.

DENSITY

Specifies the density of the tape volume where the output file will be written. Accepted values are 1600 or 6250.

- Section 3.3.5 describes additional restrictions that may apply for tape-resident files.
- If the output file is not on tape, DTF ignores this field.
- If you do not specify this field and the output file is on tape, DTF assumes the highest density allowed for the tape unit.

DTF options:**FILE DEFINITION**

Specifies the name of an entry in the DTF file definition database on the OpenVMS/DTF server node. Chapter 2 explains file definition records.

7.12 The Network File Options Panel

Figure 7-12 Network File Options Panel

```
----- DTF/ISPF - NETWORK FILE OPTIONS -----  
COMMAND ===>  
  
Fill in the required fields and press ENTER to continue  
  
NETWORK FILE NAME (if more space is required);  
  
FILE NAME ===>
```

LKG-8293-93R

DTF displays the Network File Options panel (see Figure 7-12) if you type Y in the OPTIONS field for a non-IBM input or output file in the Send to Remote or Receive from Remote panel.

You can press **PF3** to return to the operation panel that invoked this panel without initiating the operation.

The following describes the field on this panel.

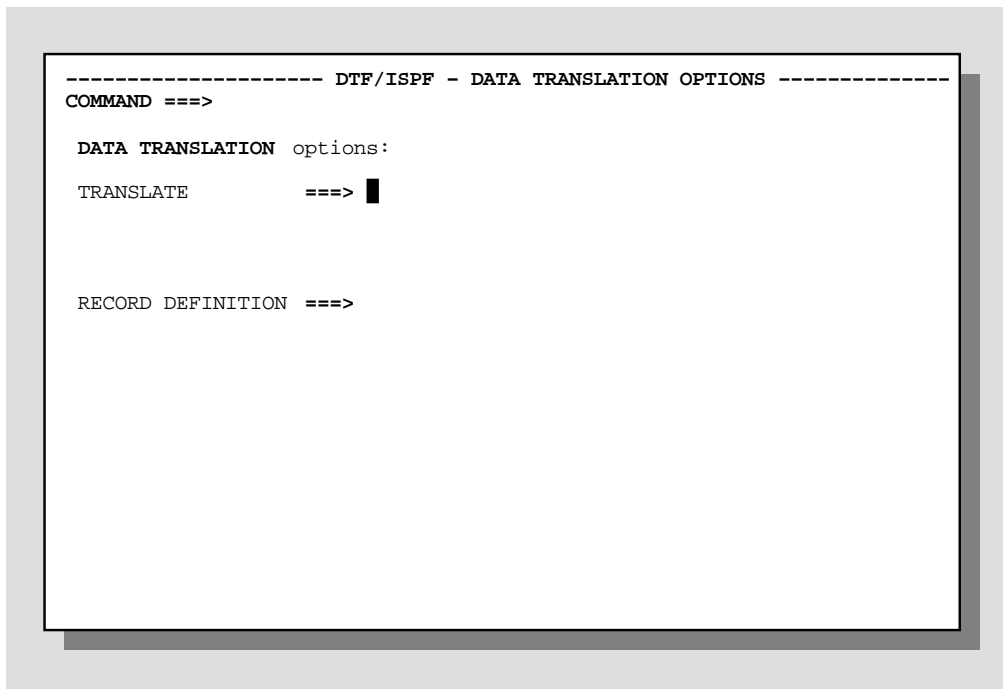
NETWORK FILE NAME (if more space is required):

FILE NAME

This is the name of the file you are copying as it exists or will exist on the remote system. You must supply this file name. (See Chapter 2 for more information on specifying file names for DTF.)

7.13 The Data Translation Options Panel

Figure 7–13 Data Translation Options Panel



LKG-8294-93R

DTF displays the Data Translation Options panel (see Figure 7–13) if you type Y in the EXTENDED field in the Send to Remote or Receive from Remote panel.

You can press **[PF3]** to return to the operation panel that invoked this panel without initiating the operation.

The following paragraphs describe the fields on this panel.

DATA TRANSLATION options:

TRANSLATE

Specifies whether you want your files translated. Appendix F provides additional information about data translation. If you type YES, DTF uses the standard EBCDIC to DMCS translation table. To translate using a different translation table, specify the file name of the file on the OpenVMS/DTF server

that contains the desired translation table. If you type NO, the file is not translated. This is the default action for IBM to IBM transfers.

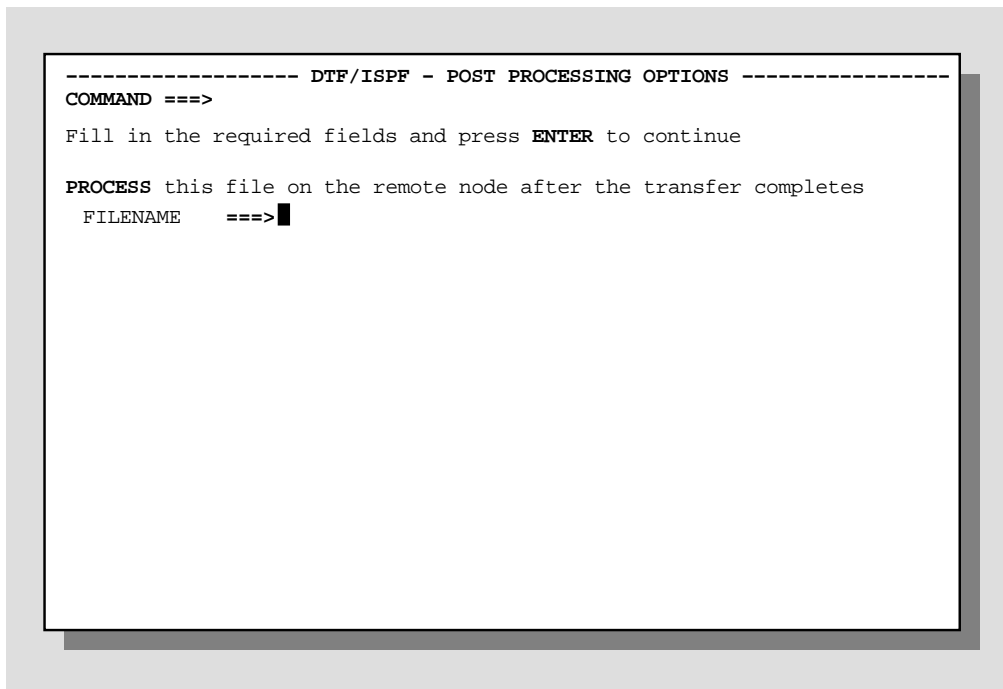
RECORD DEFINITION

Specifies the path name that is used to access a record definition in the VAX Common Data Dictionary (CDD) on the OpenVMS/DTF server node. The maximum length for this value is 255 characters.

Appendix F provides additional information about field level translation with the CDD.

7.14 The Post-Processing Options Panel

Figure 7-14 Post-Processing Options Panel



LKG-8308-93R

DTF displays the Post-Processing Options panel (see Figure 7-14) if you type Y in the POSTPROCESS field of the Transfer to Remote panel.

You can press **[PF3]** to return to the operation panel that invoked this panel without initiating the operation.

The following paragraph describes the field on this panel.

PROCESS this file on the remote node after the transfer completes:

FILENAME

Supply a file name on the remote system that you want to submit to the remote node's batch system when the transfer completes.

7.15 The Batch Options Panel

Figure 7–15 Batch Options Panel (VM Systems)

```
----- DTF/ISPF - BATCH OPTIONS -----
COMMAND ===>

Fill in the required fields and press ENTER to continue

FUNCTION      ===> █                (SUBmit or GENERate)

SAVE generated job in this file:

FILE NAME     ===>
OVERWRITE?    ===> NO                (Y/N)

JOB CONTROL information:

BATCH MACHINE to do copy : ===>
JOB CLASS      : ===>
Your 'A' disk READ PASSWORD: ===>

===>
===>
===>
===>
```

LKG-8295-93R

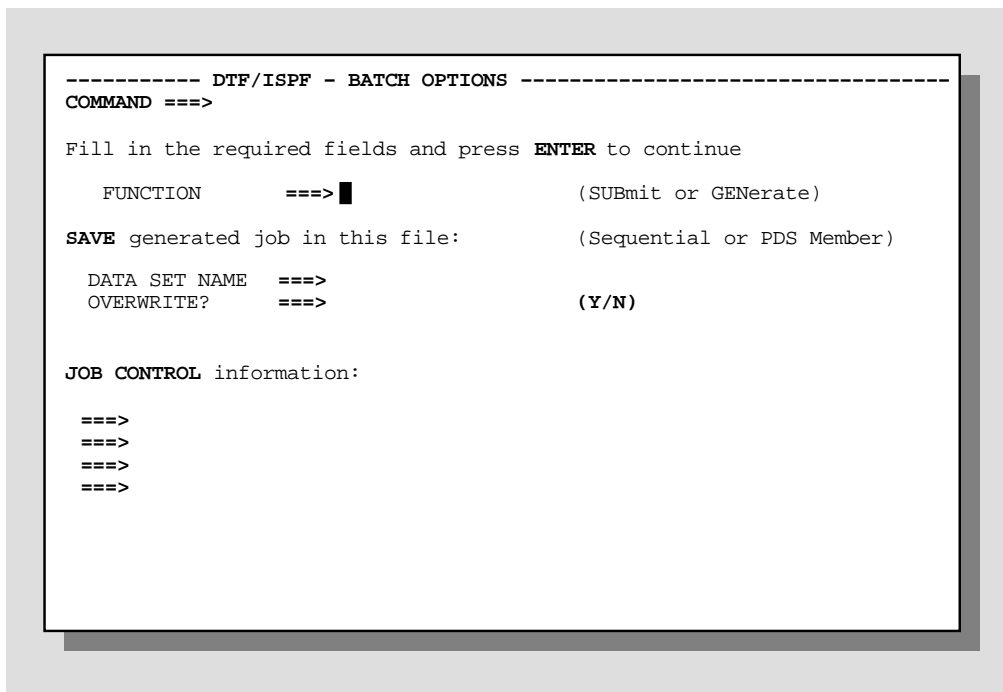
DTF displays the Batch Options panel (see Figure 7–15 and Figure 7–16) if you type Y in the BATCH SUB field in any of the three operation panels.

The Batch Options panel differs slightly between the VM and MVS version. The panel prompts you for job control information and DTF builds a default job card. The panel allows you modify the default job card and to enter additional JCL as needed.

You can press **PF3** to return to the operation panel that invoked this panel without initiating the operation.

The fields on this panel are described following Figure 7-16.

Figure 7-16 Batch Options Panel (MVS Systems)



LKG-8296-93R

FUNCTION

Enter SUB to submit the job immediately to the batch subsystem. Enter GEN to generate the JCL required to submit the batch transfer operation and store the job in a file. You may then submit this file at any time.

SAVE generated job in this file:

FILENAME or DATA SET NAME

If you chose the GENerate function, specify the file or data set where DTF should save the generated job.

OVERWRITE?

Specify Y if you wish to overwrite an existing file.

JOB CONTROL information

BATCH MACHINE to do copy:

This is the name of the virtual machine you wish to perform the copy.

JOB CLASS

Specify a job class valid for the indicated batch machine.

Your 'A' disk READ PASSWORD

This is the READ password for your A minidisk. For more information on minidisk passwords, see the /MDxPASSWORD qualifier description in Section 2.3.3.

JCL COMMANDS (==>)

Enter up to four lines of JCL commands to be stored in the file you indicated.

Note

You must provide the minidisk address, owner, and password for the local file involved in the transfer. In addition, when accessing a local file for write access, you must log off before the batch job runs or the batch job will not be able to obtain write access.

8

Using the DTF for IBM File Transfer Commands

In addition to the DTF for IBM panel interface, you can use single-line DTF for IBM commands to transfer files. This chapter explains how to use the DTF file transfer commands to copy files from one system to another. You can use these commands from TSO (Time Sharing Option) on MVS systems (TSO/E with REXX support) or from CMS (Conversational Monitor System) on VM systems.

There are three interactive DTF file transfer commands that you can use:

1. DTFSEND
2. DTFRECV
3. DTFRESUM

The following sections describe these commands and their parameters.

You can use options with all of the DTF file transfer commands. You must separate the command options from the rest of the command line with a single left parenthesis - (- as in the following example:

```
DTFSEND PROFILE EXEC A TO JONES AT NYC AS ACCOUNTING.DAT ( PASS SECRET
```

All command options can be abbreviated as long as they remain unique.

VM If you type in one of the DTF file transfer commands without specifying any further information, DTF enters the DTF non-ISPF panel interface (see Chapter 7). The panels prompt you for the rest of the information DTF needs to complete the file transfer.

MVS The panel interface cannot be entered by using DTF single-line commands.

VM You can cancel a transfer at any time by typing the CANCEL command and pressing **Enter**. You can also find out the status of a transfer by typing the QUERY command and pressing **Enter**.

[MVS] The method you use to cancel or query a transfer depends on the type of terminal you are using. For non-SNA terminals, you must press **[Reset]** followed by **[PA1]**. For SNA terminals, you must first press **[Attn]**. Then type **C** to cancel a transfer or type **Q** to query the status of a transfer and then press **[Enter]**. To exit the automatically updated query screen, type **W** and then press **[Enter]**.

8.1 DTFSEND

You use the DTFSEND command to transfer a file on either the local MVS or VM system to a node on a DECnet network, or to another IBM system.

```
DTFSEND input-file TO user-name AT node-name AS output-file [( options)]
```

input-file

This is the name of the file you want to copy as it exists on the local system. You must specify the correct file syntax for the type of system you are using. VM system file specifications have a file name, a file type, and, optionally, a file mode, and may be VM maclib members. MVS files may be Partitioned Data Set members. You should refer to manuals that describe your operating system's file system for the correct file syntax.

user-name

This is the user ID on the remote system to which you want to copy the file.

node-name

This is the node name of the remote system. In most cases, this is a node on a DECnet network. You can, however, specify node name abbreviations for IBM systems.

output-file

This parameter identifies the output file on the remote system. You must know how to specify the name of the file on this system correctly (see Chapter 2). This is a required parameter.

Options

CHECKPOINT

Sets the number of logical records between checkpoints. The number of logical records can be from 1 to 65535.

Note

If you do not specify a value for this option, the TRANSFER/DTF utility uses a default value of 1000. The CHECKPOINT option is valid only if you specify the RECOVERABLE option as YES. If you specify the RECOVERABLE option as NO, DTF does not permit you to enter a value in the CHECKPOINT option.

DENSITY

The density of the tape containing the input file. The value must be either 1600 or 6250.

DISP

This option specifies the disposition of the new file. You can choose one of the following three values:

1. NEW
2. OLD
3. REP

NEW creates a new output file. If a file with the same name already exists, DTF does not copy the file. This is the default.

OLD creates a new output file that overlays an existing file of the same name. If the file does not exist, DTF issues an error message.

REP creates a new output file even if a file with the same name already exists.

LABEL

The type of tape labels used on the input file's tape. For MVS systems, specify SL or AL. For VM systems, specify SL.

MDADDR

The output file's 1- to 4-character hexadecimal minidisk address. This option is valid only when transferring files to a VM system.

MOWNER

The output file's 1- to 8-character minidisk owner. This option is valid only when transferring files to a VM system.

MDPASS

The output file's multiwrite minidisk password. This option is valid only when transferring files to a VM system. For more information about minidisk passwords, see the /MDxPASSWORD qualifier in Section 2.3.3.

NETMGR

Specifies the name of the DTF Network Manager component. If you specify a name, the value overrides the name in the DTFUSER parameter file. On VM systems, the name is the 1- to 8-character virtual machine name. On MVS systems, the name is the 1- to 4-character subsystem name as it appears in the subsystem name table.

NULL

The action that DTF should take when it encounters a null record during a transfer to the local system.

SPACE	DTF inserts a record containing a single space (x'40') in place of the null record.
REJECT	DTF gives an error message if transferring an MVS record.
IGNORE	DTF ignores all null records. Note that the output file may have fewer records and that the null records lost will not be restored if DTF is used to transfer the file from the VM system.
NONE	DTF accepts the records as is if transferring to an MVS system. Transfers to a VM system will result in an error message.

NUMBER

The file number for input files residing on tape. The number must be in the range of 1 to 9999.

PASS

This is the user's password on the remote system. If you do not specify a password, DTF will prompt you for one.

RECOVERY

Controls whether the transfer is recoverable or nonrecoverable. Chapter 1 provides additional information about recoverable copy.

You can select one of the following options:

- Y** The transfer operation will execute in recoverable mode.
- N** The transfer operation will execute in nonrecoverable mode.

Notes

- DTF uses a default that is set up when DTF is installed at your site.
 - If you specify the RECOVERY as Y but the client node does not have the DTF utilities software installed (OpenVMS client node without the DTF utilities software, or DECnet client nodes running MS-DOS, RSX-11M/M-PLUS, OS/2, Digital UNIX, or ULTRIX), the transfer request will be rejected.
-

SERVER

This name specifies the OpenVMS/DTF server to which DTF sends the transfer request. SERVER is the name of a DECnet node.

TRANSLATE

If you specify Y for this option, DTF performs the default standard translation of EBCDIC to DMCS. If you type N, DTF does not translate the file from EBCDIC to DMCS format.

UNIT

This parameter describes a generic device that holds the IBM input file. The name can be up to eight characters long.

MVS If the input file you are trying to copy is not cataloged, or you want to avoid a catalog search to find the file, you should specify both the volume name and the unit name.

VM If the input file is on tape, you must indicate this by specifying a tape unit in this field. You can specify the tape unit as TAPE, T3420 or T3480. An IBM site can use additional or different unit names. Check with the system programmer of the site where the input file resides to get the unit names in use at the site.

VOLUME

Specifies the 1- to 6-character volume serial number for the volume containing the input file.

MVS If the input file you are trying to copy is not cataloged, or you want to avoid a catalog search to find the file, you should specify both the volume name and the unit name.

VM This option is valid only for input files on tape.

VSAMDISK

The 1- to 4-character minidisk address for the input file. This option is valid only when the VSAM input file is on a minidisk.

Examples

MVS

```
DTFSEND GIBSON.TESTS(TEST5) TO JONES AT BOSTON AS TEST.DAT
```

This example shows how to copy the PDS member called TEST5 in the PDS GIBSON.TESTS on the local MVS system to a user named JONES at node BOSTON on a DECnet network. The new filename on the DECnet node is TEST.DAT.

MVS

```
DTFSEND JONES.TESTFILE TO RICHARDS AT BOSTON AS TEST.DAT ( PASS SECRET
```

This example shows how to copy the file JONES.TESTFILE from the local MVS system to a user named RICHARDS on node BOSTON on a DECnet network. The new filename on the DECnet node is TEST.DAT. The password for user RICHARDS on the DECnet node is SECRET.

VM

```
DTFSEND 'TEST.DSDS/VSAM' TO SMITH AT BOSTON AS TEST.DAT ( NETMGR DTFGCS  
VSAMDISK 193
```

This example shows how to copy a VSAM file called TEST.DSDS (located on minidisk 193) from the local VM system to a user named SMITH at node BOSTON on a DECnet network. The new filename on the DECnet node is TEST.DAT. This example requests that the subsystem DTFGCS handle the transfer and specifies that the input file is a VSAM file.

VM

```
DTFSEND DMSSP MACLIB A(LINERO) TO GIBSON AT BOSTON AS TEST.DAT
```

This example shows how to copy the maclib member called LINERO in the maclib DMSSP on the local VM system to a user named GIBSON at node BOSTON on a DECnet network. The new filename on the DECnet node is TEST.DAT.

8.2 DTFRECV

You use this command to transfer a file from a remote system to the local MVS or VM system you are on.

`DTFRECV input-file FROM user-name AT node-name AS output-file [(options)]`

input-file

This parameter identifies the input file on the remote system. You must know how to specify the name of the file on this system correctly (see Chapter 2). This is a required parameter.

user-name

This parameter identifies the user ID name on the remote system.

node-name

This parameter identifies the node name of the remote system.

output-file

This parameter identifies the name of the output file on the local system. You must specify the correct file syntax for the type of system you are on. VM system file specifications have a file name, a file type, and, optionally, a file mode, and may be VM maclib members. MVS files may be Partitioned Data Set members. Refer to your local operating system file system manuals for information on how to specify the correct file syntax.

Options

CHECKPOINT

Sets the number of logical records between checkpoints. The number of logical records can be from 1 to 65535.

Note

If you do not specify a value for this option, the TRANSFER/DTF utility uses a default value of 1000. The CHECKPOINT option is valid only if you specify the RECOVERABLE option as YES. If you specify the RECOVERABLE option as NO, DTF does not permit you to enter a value in the CHECKPOINT field.

DENSITY

The density of the tape containing the output file. The value must be either 1600 or 6250.

DISP

This option specifies the disposition of the new file. You can choose one of the following three values:

1. NEW
2. OLD
3. REP

NEW creates a new output file. If a file with the same name already exists, DTF does not copy the file. This is the default.

OLD creates a new output file that overlays an existing file of the same name. If the file does not exist, DTF issues an error message.

REP creates a new output file even if a file with the same name already exists.

LABEL

The type of tape labels used on the output file's tape. For MVS systems, specify SL, AL, BLP, or NL. For VM systems, specify SL, BLP, or NL.

MDADDR

The input file's 1- to 4-character hexadecimal minidisk address. This option is valid only when transferring files from a VM system.

MOWNER

The input file's 1- to 8-character minidisk owner. This option is valid only when transferring files from a VM system.

MPASS

The input file's multiwrite minidisk password. This option is valid only when transferring files from a VM system. For more information about minidisk passwords, see the /MDxPASSWORD qualifier in Section 2.3.3.

NETMGR

Specifies the name of the DTF Network Manager component. If you specify a name, the value overrides the name in the DTFUSER parameter file. On VM systems, the name is the 1- to 8-character virtual machine name. On MVS systems, the name is the 4-character subsystem name as it appears in the subsystem name table..

NULL

The action that DTF should take when it encounters a null record during a transfer to a VM system.

SPACE	DTF inserts a record containing a single space (x'40') in place of the null record.
REJECT	DTF gives an error message if transferring an MVS record.
IGNORE	DTF ignores all null records. Note that the output file may have fewer records and that the null records lost will not be restored if DTF is used to transfer the file from the VM system.
NONE	DTF accepts the records as is if transferring to an MVS system. Transfers to a VM system will result in an error message.

NUMBER

The file number for output files being written to tape. The number must be in the range of 1 to 9999.

PASS

This is the user's password on the remote system. If you do not specify a password DTF, will prompt you for one.

RECOVERY

Controls whether the transfer is recoverable or nonrecoverable. Chapter 1 provides additional information about recoverable copy.

You can select one of the following options:

- Y** The transfer operation will execute in recoverable mode.
- N** The transfer operation will execute in nonrecoverable mode.

Notes

- DTF uses a default that is set up when DTF is installed at your site.
 - If you specify the RECOVERY as Y but the client node does not have the DTF utilities software installed (OpenVMS client node without the DTF utilities software, or DECnet client nodes running MS-DOS, RSX-11M/M-PLUS, OS/2, Digital UNIX, or ULTRIX), the transfer request will be rejected.
-

SERVER

This name specifies the server to which DTF sends the transfer request. SERVER is the name of a DECnet node.

TRANSLATE

If you specify Y for this option, DTF performs the default standard translation of EBCDIC to DMCS. If you type N, DTF does not translate the file from EBCDIC to DMCS format.

UNIT

This parameter describes a generic device that holds the IBM output file. The name can be up to eight characters long.

VM If the output file is on tape, you must indicate this by specifying a tape unit in this field. You can specify the tape unit as TAPE, T3420 or T3480. An IBM site can use additional or different unit names. Check with the system programmer of the site where the output file resides to get the unit names in use at the site.

VOLUME

Specifies the 1- to 6-character volume serial number for the volume containing the output file.

VM This option is valid only for output files on tape.

VSAMDISK

The 1- to 4-character minidisk address for the VSAM output file. This option is valid only when issued on a VM system and only when the output file is on a VSAM minidisk.

Examples

MVS

```
DTFRECVC LOGIN.COM FROM SMITH AT MIAMI AS FRED.LOGIN.COM ( PASS ZEWOJUDE
```

This example shows how to copy a file called LOGIN.COM from a DECnet node named MIAMI to the local file FRED.LOGIN.COM on an MVS system. The file on the DECnet node is accessed through the SMITH account using the password ZEWOJUDE.

MVS

```
DTFRECVC TEXT.LIS FROM JONES AT SMAUG AS JONES.TEXT.LIST ( PASS SECRET DISP NEW
```

This example shows how to copy a file called TEXT.LIS from a DECnet node named SMAUG to the local file JONES.TEXT.LIST on the local MVS system. The file on the DECnet node is accessed through the JONES account using the password SECRET.

VM

```
DTFRCV MONDAY.DAT FROM SIMPSON AT WMASS AS MONDAY DATA ( PASS SECRET  
SERVER TIAMAT NETMGR DTFGCS
```

This example shows how to copy a file called MONDAY.DAT from a DECnet node named WMASS to the file MONDAY DATA on the local VM system. The file on the DECnet node is accessed through the SIMPSON account using the password SECRET. The operation is to be done using the server TIAMAT and the Network Manager subsystem DTFGCS.

VM

```
DTFRCV PROGRAM1.BIN FROM JONES AT SMAUG AS PROGRAM1 EXEC ( PASS SECRET  
RECOVERY Y TRANSLATE N
```

This example shows how to copy a binary file called PROGRAM1.BIN from a DECnet node named SMAUG to the file PROGRAM1 EXEC on the local VM system. The file on the DECnet node is accessed through the JONES account using the password SECRET. The file operation is done using the recovery feature. Note that all binary files must be transferred with TRANSLATE set to N.

8.3 DTFRESUM

You use this command to resume a failed recoverable transfer. During the transfer you can obtain status information by typing the QUERY command.

```
DTFRESUM RETNUM resume-number [( options)]
```

resume-number

Specifies a number up to five digits long that appears in messages beginning with DTF0062I when you submit the request.

Options

NODE

Specifies the name of the client system which is cooperating in the recoverable transfer.

PASS

The user's password on the client node.

USER

This is the user ID of the user on the client node.

NETMGR

This parameter is the name of the DTF Network Manager component. On VM systems, the name is the 1- to 8-character virtual machine name. On MVS systems, the name is the 4-character subsystem ID.

Examples

```
DTFRESUM RETNUM 5 ( NODE HARPO
```

In this example, the interrupted file operation with the retry number 5 is resumed. The DECnet node involved in the transfer is HARPO.

```
DTFRESUM RETNUM 4 ( NODE SMOG USER JONES PASS SECRET
```

In this example, the interrupted file operation with the retry number 4 is resumed. The DECnet node involved in the transfer is SMOG, the user ID is JONES and the user password is SECRET.

```
DTFRESUM RETNUM 222 ( NODE TOOK USER BARNEY PASS FRED
```

In this example, the interrupted file operation with the retry number 222 is resumed. The DECnet node involved in the transfer is TOOK, the user ID is BARNEY and the user password is FRED.

9

Using the DTF for IBM Command Processor

This chapter describes how to use the DTF command processor to transfer files. It provides examples that show you how to perform the most common command processor tasks. The chapter is divided into three sections:

- How to invoke the DTF command processor.
- A detailed description of the DTF command and all of the supported subcommands, keywords, and keyvalues that you can use to transfer files.
- A description of DTF command processor batch mode operations and DTF command processor operations that you can initiate from procedure languages.

9.1 Invoking the DTF Command Processor

Users of both TSO on MVS systems and CMS on VM can use the DTF command processor.

Type the DTF command after the READY prompt to start the DTF command processor and initiate DTF mode, as shown:

```
DTF
```

You can now use the DTF subcommands to transfer files between IBM and DECnet systems. The following sections explain the DTF command and subcommands.

9.2 The DTF Command

The DTF command allows you to use DTF to transfer files between an IBM system and DECnet nodes. After you enter DTF, you use the DEFINE subcommand to set up input and output file formats. Then, use the TRANSFER subcommand to transfer files.

The DTF command does not allow keyword parameters and cannot be abbreviated. The syntax of the DTF command is:

DTF

9.3 DTF Subcommands

Through the use of the following subcommands, the DTF command allows you to perform the following functions:

If you want to do this:	Use this DTF subcommand:
List all of the DTF subcommands and briefly explain the function, syntax, and parameters of each subcommand.	HELP
Define the input file, the output file, and the environment that the transfer will execute in.	DEFINE
Display all the DEFINE subcommands that you entered but have not invoked with the TRANSFER subcommand.	LIST
Initiate the file transfer and erase all of the DEFINE subcommands you entered.	TRANSFER
Recall all the DEFINE subcommands you entered immediately before a TRANSFER subcommand.	RECALL
Restart a previously-failed transfer from the last successful checkpoint.	RESUME
Change the operation of the DTF command processor.	SET
Show DTF defaults.	SHOW
Remove input and output file specifications set up with the DEFINE subcommand and reset default values.	CLEAR
Exit DTF.	END

Syntax for Keywords

Keywords are separated from the DTF subcommands by one or more spaces:

dtf-subcommand keyword(keyvalue)

Multiple keywords can be specified for the DTF subcommands by separating each keyword by one or more spaces:

dtf-subcommand keyword(keyvalue) keyword(keyvalue)

All keywords can be abbreviated to the shortest possible unique string.

Syntax for Keyvalues

Keywords that allow keyvalues must use parentheses to enclose the keyvalue. The opening parenthesis must immediately follow the keyword and the closing parenthesis must immediately follow the keyvalue. Spaces cannot be embedded within the parentheses (except for CMS file specifications):

dtf-subcommand keyword(keyvalue) keyword(keyvalue)

Keyword and Keyvalue Examples

The following examples illustrate how to enter keywords and keyvalues.

```
DEFINE NETMGR(DTF)
```

The DEFINE command was entered with the keyword NETMGR and the keyvalue DTF.

```
DEFINE SERVER(NODE1) NETMGR(DTF2)
```

The DEFINE command was entered with the two keywords SERVER and NETMGR, which have the keyvalues NODE1 and DTF2, respectively.

```
DEFINE SERVER()
```

The DEFINE command was entered with a keyword and a null keyvalue.

Entering an IBM File Specification

An IBM file specification does not include a node name or the double colon (::) node syntax. An IBM file specification can include IBM file specification qualifiers if the entire file specification is enclosed within single quotation marks (' '). Chapter 2 explains the syntax of an IBM file specification and describes the qualifiers you can include.

Entering a DECnet File Specification

A DECnet file specification includes a OpenVMS/DTF client node name and the double colon (::) node syntax as shown in the following format:

node::filename

Chapter 2 explains the syntax of OpenVMS, RSX-11M/M-PLUS, MS-DOS, OS/2, Digital UNIX, and ULTRIX file specifications.

You can enter optional access control information within the node portion of the specification by enclosing that information in quotation marks, as shown in the following format:

node" access-control-information"::filename

Although the access control information can include user IDs and passwords associated with the input and output files, this is not recommended. When this information is entered within the node portion of the file specification, the password is visible on the screen. A better alternative is to use the INUSER/OUTUSER and INPASS/OUTPASS keywords when you define the input and output files.

Chapter 2 explains the syntax you must use when entering IBM and DECnet file specifications and explains the defaulting that is available. Chapter 3 describes any limitations that must be observed when specifying input and output file specifications, and lists the types of file organizations, record formats, and file attributes that are supported for input and output files. You should read these chapters carefully before using the DTF command processor interface.

9.3.1 CLEAR Subcommand

The CLEAR subcommand allows you to remove the input and output file specifications set up with the DEFINE subcommand.

The syntax of the CLEAR subcommand is:

```
{ CLEAR }  
{ C }
```

CLEAR Example

The following example shows how to use the the CLEAR subcommand to erase the values set up with the DEFINE subcommand.

```
LIST [1]  
DEF INFILE(CLUSTR::LOGIN.COM) INUSER(USER1) INPASS(****)  
DEF OUTFILE(IBMUSER.LOGIN.COM) DISP(OLD)  
DEF SERVER(NODE1) NETMGR(DTF2) RECOVERY(N)  
DTF0047I DTF  
CLEAR [2]  
DTF0047I DTF  
LIST [3]  
DEF NETMGR(DTF) RECOVERY(Y)  
DTF0047I DTF
```

1. This LIST subcommand displays the values that were defined before the CLEAR subcommand was entered.

2. The CLEAR subcommand clears or resets all previously defined values.
3. This LIST subcommand displays all currently defined values.

9.3.2 DEFINE Subcommand

The DEFINE subcommand defines input and output file specifications and the environment associated with a file transfer.

You can define an input and an output file specification with one or more DEFINE subcommands, then start the file transfer with a TRANSFER subcommand.

The syntax of the DEFINE subcommand is:

	CHKPT	
	DISP	{ OLD NEW REP }
	INCASE	
	INDEN	{ 1600 6250 }
	INFDEF	
	INFILE	
	INLABEL	
	INMDADDR	
	INMDOWNER	
	INMDPASS	
	INNUM	
	INPASS	
	INRMT	
	INUNIT	
	INUSER	
	INVOL	
	OUTCASE	
{ DEFINE }	OUTDEN	{ 1600 6250 }
{ DEF }	OUTPRIMARY	
	OUTSECONDARY	
	OUTFDEF	
	OUTFILE	
	OUTLABEL	
	OUTMDADDR	
	OUTMDOWNER	
	OUTMDPASS	
	OUTNUM	
	OUTPASS	
	OUTRMT	
	OUTUNIT	
	OUTUSER	
	OUTVOL	
	NETMGR	
	NULL	
	RECDEF	
	RECOVERY	
	SERVER	
	TRANSLATE	

Note

All transfer requests must specify an INFILE and an OUTFILE. Most of the other keywords have defaults available.

The DEFINE subcommand is associated with keywords and keyvalues that define the following:

- The input file used in the transfer.
- The output file created or overlaid during the transfer.
- Parameters that define the environment in which the transfer is performed.

The following sections explain these keywords and keyvalues.

9.3.2.1 Keywords and Keyvalues for Specifying the Input File

The following sections explain the DEFINE subcommand keywords and keyvalues that describe the input file used in the transfer.

INCASE

This field specifies whether DTF should accept the case in which the file name was entered. A value of Y will cause DTF to open the file in a case sensitive fashion. N or blank will default to uppercase.

Note

This option has no effect unless NODEPARMS LOWERCASECREATE is set to ON.

See the /CASE qualifier in Chapter 2 for information on MIXED CASE.

INFILE(*file-specification*)

Specifies the input file used in the transfer request. The file transfer must include at least one IBM file as either an input file or an output file. A file transfer between two DECnet files is not allowed.

If you define an input file as an IBM file specification, then change that file specification to a Digital file specification by entering another DEFINE subcommand, the DTF command processor will clear the IBM-related parameters automatically.

file-specification

The type of input file will determine the syntax of the *file-specification* keyvalue:

IBM file specification Does not include a node name or the double colon (::) node syntax.

DECnet file specification Includes an OpenVMS/DTF client node name and the double colon (::) node syntax as shown in the following format:

node"access-control-information"::filename

Chapter 2 provides detailed information that describes the proper syntax to use for OpenVMS, RSX-11M/M-PLUS, MS-DOS, OS/2, Digital UNIX, and ULTRIX file specifications.

Usage Notes:

- Client nodes running Digital UNIX or ULTRIX use case-sensitive file names. Make sure that you use the proper case when the input file resides on an Digital UNIX or ULTRIX systems.
- The node portion of the DECnet file specification can contain access control information. However, for security reasons this is not recommended, because the password you enter is visible. A better way to enter access control information is by using the INUSER keyword and having the system prompt you for the INPASS in a nondisplay field.
- If you specify an IBM file specification, you can include IBM file specification qualifiers by enclosing the entire file specification within single quotation marks (' '). Appendix G lists the qualifiers that you can use.

INDEN { (1600) }
{ (6250) }

Specifies the density of the tape volume from which the input file is read.

1600

1600 bits per inch (bpi).

6250

6250 bits per inch (bpi).

Usage Notes:

- Tape restrictions may apply. Section 9.4.1 describes these restrictions.
- If you do not specify a keyword value and the input file is on tape, the DTF command processor will select the highest density allowed for that tape volume.

- If the input file is neither on tape nor an IBM file specification, the DTF command processor will ignore this parameter.

INFDEF(*file-specification*)

Specifies reading a file definition record in the OpenVMS/DTF server's file definition database. For a complete discussion of file definition records, refer to Chapter 2.

file-specification

A file definition record in the file definition database. Length: up to 32 characters.

Usage Note:

- The *file-specification* keyvalue is valid only if the INFILE is an IBM file specification. If the input is not an IBM file specification, the DTF command processor will ignore this parameter.

INLABEL(*label*)

Specifies the format of the tape labels on the tape that the input file is read from.

label

One of the following directory labels:

- AL** ANSI V1 labels or ISO/ANSI/FIPS V3 labels. Not supported for VM input files.
- SL** Standard labels

Usage Notes:

- Tape restrictions may apply. Section 9.4.1 describes any restrictions.
- You should specify the INLABEL keyword only if the input file resides on a tape.
- If the input file resides on disk, the DTF command processor will ignore this keyword.
- If you do not specify this keyword and the input file resides on tape, the DTF command processor will assume that standard tape labels are used.

INMDADDR(*file-number*) VM

If the minidisk where the input file resides is not currently linked and accessed, use this keyword to supply the address of the minidisk. This may refer to a minidisk you own or the one owned by the user in the INMDOWNER field.

INMDOWNER(*user-id*) VM

The owner of the minidisk where the input file is located. If you own the minidisk you do not have to use this keyword.

INMDPASS(*password*) VM

If the minidisk with the input file is not currently linked and accessed, then use this parameter to supply the read minidisk password for a link to the minidisk identified in the INMDADDR keyword. For more information about minidisk passwords, see the /MDxPASSWORD qualifier in Section 2.3.3.

INNUM(*file-number*)

The INNUM keyword specifies a sequential file number on tape for an IBM input file.

file-number

A sequential file number on tape. Range: 1 to 9999.

Usage Notes:

- Tape restrictions may apply. Section 9.4.1 describes these restrictions.
- If the input file resides on disk, the DTF command processor will ignore the INNUM keyword.
- If you specify the INNUM keyword but do not specify a value for *file-number*, the DTF processor will default to INNUM(1) if the input file is on tape.

INPASS(*password*)

Specifies the password associated with the INUSER account.

password

The password associated with the INUSER account. The syntax of the input file will determine the format of the *password* keyvalue.

DECnet The password that corresponds to the user ID to be used when contacting the client node. Length: 1 to 31 characters.

IBM The password that corresponds to the user ID that the DTF Network Manager subsystem uses when it verifies access to the input file. Length: 1 to 8 characters.

Usage Notes:

- If the remote client node is running Digital UNIX or ULTRIX, you must enter the *password* keyvalue with the proper case.
- If you enter the INFILE keyword with access control information that includes a password, the DTF command processor will ignore this keyword.

- If you do not specify a user ID, either in the access control information or with the INUSER keyword, the DTF command processor will ignore this keyword.

INRMT(*node*"*server-account-name*"::)

Specifies an OpenVMS/DTF server node name and server account name or a DECnet-Internet Gateway.

***node*"*server-account-name*"::**

A DECnet OpenVMS/DTF server node or TCP/IP host name and optional access control information. If a DECnet-Internet Gateway node is specified, do not specify the access control information.

INUNIT(*unit-specification*)

Specifies a generic device name that describes the device type that contains the IBM input file.

unit-specification

The unit specification. Length: 1 to 8 characters.

Usage Notes:

- If the input file is not cataloged, you should specify this keyword.
- To avoid a catalog search to find the file, you should specify this keyword.
- For VM systems, if the input file is on tape, you must indicate this by specifying a tape unit in this field. You can specify the tape unit as TAPE, T3420 or T3480. An IBM site can use additional or different unit names. Check with the system programmer of the site where the input file resides to get the unit names in use at the site.

INUSER(*user-id*)

Specifies the user ID used to access the input file.

user-id

The user ID used to access the input file.

The syntax of the input file will determine the format of the *user-id* keyvalue.

DECnet user ID The logon ID to use when contacting the client node. Length: 1 to 40 characters.

IBM user ID The logon ID to use when the DTF Network Manager verifies access to the input file. Length: 1 to 8 characters.

Usage Notes:

- If the remote client node is running Digital UNIX or ULTRIX, you must enter the user ID with the proper case.
- If you specified the INFILE keyword with access control information, the DTF command processor will ignore this keyword.

INVOL(volume-serial-number)

Specifies a volume serial number that contains the IBM input file.

volume-serial-number

A volume serial number that contains the IBM input file. Length: 1 to 6 characters.

Usage Notes:

- If the input file is not cataloged, you should specify this keyword.
- To avoid a catalog search to find the file, you should specify this keyword.

9.3.2.2 Keywords and Keyvalues for Specifying the Output File

The following sections explain the keywords and keyvalues that describe the output file created or overlaid during the transfer.

OUTCASE

This field specifies whether DTF should accept the case in which the file name was entered. A value of Y will cause DTF to open the file in a case sensitive fashion. N or blank will default to uppercase.

Note

This option has no effect unless NODEPARMS LOWERCASECREATE is set to ON.

See the /CASE qualifier in Chapter 2 for information on MIXED CASE.

OUTFILE(file-specification)

Specifies the output file that will be either created or overlaid as a result of the transfer request. The file transfer must include at least one IBM file specification as either an input file or an output file. Two DECnet file specifications are not allowed.

file-specification

The name of the output file that will be either created or overlaid during the transfer.

The type of output file will determine the syntax of the *file-specification* keyvalue:

IBM file specification Does not include a node name or the double colon (::) node syntax.

DECnet file specification Includes an OpenVMS/DTF client node name and the double colon (::) node syntax as shown in the following format:

node"access-control-information"::filename

Chapter 2 provides detailed information that describes the proper syntax to use for OpenVMS, RSX-11M/M-PLUS, MS-DOS, OS/2, Digital UNIX, and ULTRIX file specifications.

Usage Notes:

- If the transfer request does not include at least one IBM file specification, the transfer will fail.
- If the *file-specification* keyvalue specifies a DECnet file, the only applicable output file-related keywords are DISP, OUTUSER, and OUTPASS.
- The DECnet file specification can contain access control information within the node portion of the file name. However, this is not recommended because the password will remain visible on the screen. The preferred method of entering access control information is through the OUTUSER and OUTPASS keywords.
- If you specify an IBM file specification for the OUTFILE, and then change the OUTFILE to a DECnet file specification by entering another DEFINE subcommand, the DTF processor will clear the IBM-related parameters automatically.

DISP { (NEW)
(OLD)
(REP) }

A required keyword that specifies whether the output file is a new or existing file.

NEW

NEW creates a new output file. If a file with the same name already exists, DTF does not copy the file. This is the default.

OLD

OLD creates a new output file that overlays an existing file of the same name. If the file does not exist, DTF issues an error message.

REP

REP creates a new output file even if a file with the same name already exists.

OUTDEN { 1600 }
 { 6250 }

Specifies the density of the tape volume to which the DTF processor will write the output file.

1600

1600 bits per inch (bpi).

6250

6250 bits per inch (bpi).

Usage Notes:

- Tape restrictions may apply. Section 9.4.1 describes these restrictions.
- The DTF command processor will ignore the OUTDEN keyword if you direct the processor to write the OUTFILE to a disk or if the output file is not an IBM file specification.
- If you do not specify this field and the output file is written to tape, the DTF command processor will specify the highest density allowed for that device.

OUTFDEF(*file-specification*)

Directs the DTF processor to read a file definition record in the file definition database. Chapter 2 explains file definition records.

file-specification

A file specification in the file definition database. Length:1 to 32 characters.

Usage Note:

- The *file-specification* keyvalue is valid only if the output file is an IBM file specification.

OUTLABEL(*label*)

Specifies the format of the tape label on the output tape.

label

The format of the tape labels on the tape that will be written.

You can specify one of the following tape labels:

- AL** ANSI V1 labels or ISO/ANSI/FIPS V3 labels. Not supported for VM output files.
- BLP** Bypass label processing
- NL** Non-labeled
- SL** Standard labels

Usage Notes:

- If you specify OUTLABEL(BLP) or OUTLABEL(NL), you must specify RECOVERY(NO). You can specify RECOVERY(YES) only if you specify either OUTLABEL(AL) or OUTLABEL(SL).
- Additional tape restrictions may apply. Section 9.4.1 describes any additional restrictions.
- Specify this keyword only if the output file is an IBM file.
- Specify this keyword only if the output file will reside on a tape.
- If the output file resides on disk, this keyword will be ignored.
- If you do not specify this field and the output file resides on tape, the DTF command processor will assume that standard tape labels are used.

OUTMDADDR(*owner-name*) VM

If the minidisk where the output file will reside is not currently linked and accessed, then use this keyword to supply the address of the minidisk. This may refer to a minidisk you own or one owned by the user ID specified in the OUTMDOWNER keyword.

OUTMDOWNER(*owner-name*) VM

The owner of the minidisk which you would like to have own the output file. If you own the minidisk then you do not need to use this keyword.

OUTMDPASS(*owner-name*) VM

If the minidisk where the output file will reside is not currently linked and accessed, use this keyword to supply the multiwrite minidisk password to the minidisk identified with the OUTMDADDR keyword. For more information about minidisk passwords, see the /MDxPASSWORD qualifier in Section 2.3.3.

OUTNUM(*file-number*)

Specifies a sequential file number on tape. It is valid only with an IBM output file.

file-number

A file number on a tape volume number from 1 to 9999.

Usage Notes:

- Tape restrictions may apply. Section 9.4.1 describes these restrictions.
- If the output file resides on disk, the DTF command processor will ignore the OUTNUM keyword.
- If you do not specify this keyword and the output file will be written to tape, the DTF command processor will default to OUTNUM(1).

OUTPASS(*password*)

Specifies the password associated with the OUTUSER account.

password

The password associated with the OUTUSER account.

The syntax of the input file specified with the OUTFILE keyword will determine the format of the *password* keyvalue:

DECnet The password that corresponds to the user ID to be used when contacting
password the client node. Length: 1 to 31 characters.

IBM The password that corresponds to the user ID to use when the DTF
password Network Manager verifies access to the file. Length: 1 to 8 characters.

Usage Notes:

- If the remote client node is running Digital UNIX or ULTRIX, you must enter the password with the proper case.
- If the node portion of the file specification that you specified with the OUTFILE keyword contains access control information that includes a password, the command processor ignores this keyword.
- If you do not specify a user ID, either in the access control information or with the OUTUSER keyword, the command processor ignores this keyword.

OUTPRIMARY

A 5-digit number specifying the number of 512-byte blocks that DTF will initially allocate when creating a file on the IBM system.

OUTRMT(*node*"*server-account-name*":::)

Specifies an OpenVMS/DTF server node and server account name or a DECnet-Internet Gateway node.

node*"*server-account-name*":::*server-account-name

An OpenVMS/DTF server node name and a server account name or a DECnet-Internet Gateway node. If a DECnet-Internet Gateway node is specified, do not specify the account name.

OUTSECONDARY

A 5-digit number specifying the number of 512-byte blocks that DTF will extend the file if the initial allocation is insufficient to hold the transferred file.

OUTUNIT(*unit-specification*)

Specifies a generic device name that describes the device type that will hold the IBM output file.

unit-specification

A generic device name that describes the device type that will hold the IBM output file. Length: 1 to 8 characters.

Usage Note:

- The OUTUNIT keyword is valid only with an IBM file specification.
- For VM systems, if the output file is on tape, you must indicate this by specifying a tape unit in this field. You can specify the tape unit as TAPE, T3420, or T3480. An IBM site can use additional or different unit names. Check with the system programmer of the site where the output file resides to get the unit names in use at the site.

OUTUSER(*user-id*)

Specifies the user ID associated with the file name specified with the OUTFILE keyword.

user-id

The user ID associated with the output file. Length: 1 to 40 characters.

The syntax of the input file specified with the OUTFILE keyword will determine the format of the *user-id* keyvalue:

**DECnet
user ID** The logon ID to use when contacting the client node. Length: 1 to 40 characters.

**IBM
user ID** The logon ID to use when the DTF Network Manager verifies access to the file. Length: 1 to 8 characters.

Usage Notes:

- If the remote client node is running Digital UNIX or ULTRIX, you must enter the user ID with the proper case.
- If you are using a batch dialog, you must include your user ID.
- If you specify the OUTFILE with access control information, the DTF command processor will ignore this keyword.

OUTVOL(*volume-serial-number*)

Specifies a volume serial number that will contain the IBM output file.

volume-serial-number

A volume serial number that will contain the IBM output file. Length: 1 to 6 characters.

Usage Note:

- The OUTVOL keyword is valid only with an IBM file specification.

9.3.2.3 Keywords and Keyvalues for Specifying Transfer Parameters

The following sections describe the keywords and keyvalues that you use to specify the transfer parameters that define the environment that the transfer will be executed in.

CHKPT(*checkpoint-interval*)

Specifies the number of logical records that can be transferred between checkpoints during a recoverable file transfer.

checkpoint-interval

The number of logical records that can be transferred between checkpoints. Range: 1 to 65535.

Usage Notes:

- If you do not specify a keyvalue for the CHKPT keyword, the TRANSFER /DTF on the client node uses a default value.
- The CHKPT keyword is valid only if you specify the RECOVERY keyword as YES. If you specify RECOVERY(NO), the command processor does not permit you to specify a keyvalue for the CHKPT keyword.

NULL { (SPACE)
(REJECT)
(IGNORE)
(NONE) }

This keyword is used to indicate how DTF processes null records (length = 0) to an IBM file. Because the CMS file system does not support null length records, DTF cannot write them to the indicated device. For more information about the NULL option see the /NULL qualifier description in Chapter 2.

SPACE

A 1-byte record is inserted in place of the zero length record. The one byte is set to an EBCDIC space character (x'40').

REJECT MVS

An error is returned if a null record is encountered.

IGNORE

The null record is simply ignored when writing the file. The result is that the output file has fewer records than the input file.

NONE

VM No special processing will occur. An error will be returned if a null record is encountered.

MVS No special processing will occur. The record will be accepted as is.

RECDEF(*path-name*)

Specifies the path name that is used to access a record definition in the VAX Common Data Dictionary (CDD). Appendix F provides additional information about field level translation with the CDD.

path-name

A CDD path name on the server node. Length: 1 to 255 characters.

RECOVERY { (YES) } { (NO) }

A required keyword that specifies whether the transfer is executed in recoverable mode or nonrecoverable mode. Chapter 1 provides additional information about recoverable copy.

YES

The transfer executes in recoverable mode.

NO

The transfer executes in nonrecoverable mode.

Usage Notes:

- If you specify RECOVERY(YES) but the client node does not have the OpenVMS/DTF utilities software installed (OpenVMS client node without the OpenVMS/DTF utilities software, or DECnet client nodes running MS-DOS, RSX-11M/M-PLUS, OS/2, Digital UNIX, or ULTRIX), the transfer request will fail.
- If you specify RECOVERY(NO) and the server or client nodes involved in the transfer request are running DTF for MVS V1.0, the command processor ignores the NO keyvalue and the transfer operation will proceed in recoverable mode.

SERVER(*node*)

Selects the server session to which the transfer request is sent for execution.

node

The node name of the node running the OpenVMS/DTF server software.

Usage Notes:

- If you specify a keyvalue for the SERVER keyword, there must be an active server session with the DTF Network Manager subsystem that uses the same server name. If there is no such server session, the transfer request will be rejected.
- If you do not specify a keyvalue for the SERVER keyword, by default DTF selects an OpenVMS/DTF server node from the pool of active server sessions. DTF first attempts to select a server node with the same name as the client node from the request. If the request does not specify a client node or if the client node is not also a server node, DTF selects the first available OpenVMS/DTF server node.

NETMGR(*dtf-name*)

A required keyword that specifies which DTF Network Manager subsystem should receive the transfer request.

dtf-name

For MVS systems, this is the 1- to 4-character name defined for DTF in the subsystems name table. For VM systems, this is the 1- to 8-character name of the Network Manager virtual machine.

Usage Note:

- If you do not specify a keyvalue for this keyword, the DTF command processor will use the site-defined default name. The *Digital SNA Data Transfer Facility Installation (VM)* or *Digital SNA Data Transfer Facility Installation (MVS)* manual explains how to establish the default subsystem name.

TRANSLATE { (YES)
(NO)
(OpenVMS-file-specification) }

An optional keyword that specifies whether you want your files translated. For more information about data translation, refer to the section in Appendix F that describes translation tables. Also see Appendix G.

YES

Data is translated using standard translation tables.

NO

Data is not translated.

OpenVMS-file-specification

An OpenVMS file specification for loading a nonstandard translation table.

Length: 1 to 255 characters.

Usage Notes:

- If the transfer involves an indexed file that contains binary data, specify TRANSLATE(NO), because data translation could generate duplicate keys. VSAM does not allow duplicate keys. If this occurs on the primary key, the transfer will terminate with errors. This restriction applies to both OpenVMS- and IBM-initiated transfers.
- If you specify an OpenVMS file specification as the keyvalue, do not include a node name, user ID, or password as part of that file specification.
- TRANSLATE(NO) is the default value if you are initiating a transfer that has IBM file specifications for both the input and output file specifications.
- TRANSLATE(YES) is the default value if the transfer has a DECnet file specification as either the input or output file specification.

This concludes the discussion of the keywords you can use with the DEFINE subcommand.

DEFINE Example

The following example shows you how to define the values necessary to set up a file transfer.

```
DEFINE INFILE(CLUSTER::LOGIN.COM) INUSER(VMSUSER) INPASS(VMSPASS) [1]
DTF0047I DTF
DEFINE OUTFILE(IBMUSER.LOGIN.COM) [2]
DTF0047I DTF
DEFINE SERVER(NODE1) NETMGR(DTF1) [3]
DTF0047I DTF
```

1. This DEFINE subcommand sets up an input file specification. The input file definition is an OpenVMS file specification because it contains a node name followed by a double colon (::).
2. This DEFINE subcommand sets up an output file specification. The output file definition is an IBM file specification because it does not contain a node name or a double colon (::).
3. This DEFINE subcommand sets up the transfer parameters.

9.3.3 END Subcommand

The END subcommand allows you to remove your terminal from DTF mode.

The syntax of the END subcommand is:

```
{ END }
```

END Example

The following example shows you how to use the END subcommand to exit DTF.

```
DEFINE INFILE(CLUSTER::LOGIN.COM) [1]
DTF0047I DTF
DEFINE OUTFILE(IBMUSER.LOGIN.COM) [2]
DTF0047I DTF
DEF SERV(NODE1) [3]
DTF0047I DTF
END [4]
READY
```

1. This DEFINE subcommand specifies the input file.
2. This DEFINE subcommand specifies the output file.
3. This DEFINE subcommand specifies the server session.
4. This END subcommand exits DTF; all previously-defined values are lost.

If a TRANSFER subcommand had been issued instead of the END subcommand, the transfer request would have been submitted.

9.3.4 TRANSFER Subcommand

The TRANSFER subcommand passes the information supplied by the DEFINE subcommands to the DTF Network Manager subsystem. The subsystem selects an OpenVMS/DTF server node to process the file transfer request and forwards the information from the DEFINE subcommands to that node.

The syntax of the TRANSFER subcommand is:

```
{ TRANSFER } [POST (file-name)]  
{ TRANS
```

POST *file-name*

Specifies a file that DTF should submit if an operation completes successfully. The file can be an OpenVMS command procedure, an MVS JCL file, or a VM proc file.

If you specify a DECnet file specification, the file is submitted on the indicated DECnet node.

If you specify an IBM file specification using the qualifiers described in Chapter 2 or a local file specification, the file is submitted to the batch subsystem on the indicated IBM system or to the local batch subsystem.

Several messages are written to the terminal to indicate the status of the transfer request. The following messages are the most important:

```
DTFM037I THIS REQUEST ASSIGNED REQUEST NUMBER xxxxx
```

The *xxxxx* value indicates the internal request number assigned by the DTF Network Manager subsystem. Use this number when you report problems to the IBM system support staff. They use the number to locate a variety of trace information.

```
DTFM062I THIS REQUEST ASSIGNED RETRY NUMBER yyyyy
```

The *yyyyy* value indicates the retry number assigned by the DECnet client node. Use this number if the transfer request fails and you want to issue a retry request later. The *yyyyy* value is used as the *retry-number* keyvalue of the RETNUM keyword with the RESUME subcommand.

Note

The DTF processor displays this message only if you specified RECOVERY(YES).

DTFM055I SUBSYSTEM zzzz REPORTS COMPLETION OF REQUEST

Processing has completed. This is followed by messages from the server node that report the outcome of the transfer request. The *Digital SNA Data Transfer Facility for OpenVMS Problem Solving and Messages* provides more information about messages from the server node.

After displaying the DTFM055I message and the related lines, DTF displays the DTF prompt. All previous defined values have been reset to their default values. The RECALL subcommand can be used to reset all defined values to the values that were set at execution time.

Canceling a TRANSFER Subcommand

[MVS] You can cancel a transfer that is in progress at any time. The method used to cancel a transfer depends on the terminal you are using. On non-SNA terminals, press **[Reset]**, then **[PA1]**. On SNA terminals, press **[Attn]**. Both terminals will display the following:

```
C=CANCEL,Q=QUERY,W=WAIT,ELSE IGNORED
```

Take one of the following actions:

- Type C and press **[Enter]** to cancel the request and free the terminal.
- Type Q and press **[Enter]** to receive status for the request.
- Type W and press **[Enter]** to continue processing.

If you respond with C, the DTF processor halts the transfer at the current position. You must then clean up the partial files. You cannot issue a RESUME subcommand to restart a canceled request.

On VM systems, you can cancel a transfer that is in progress at any time by typing the command CANCEL and pressing **[Enter]**. You can request a status report for a transfer in progress by typing the command QUERY and pressing **[Enter]**.

TRANSFER Subcommand Prompts

The DTF command processor may prompt you to enter passwords after you issue the TRANSFER subcommand. You will enter the information you were prompted for in nondisplayable fields.

The DTF command processor prompts you to enter the password for the input file if the following requirements are met:

- You specified the INUSER keyword.

Note

If the INFILE specifies a DECnet file specification, you can specify the INUSER value within the access control information portion of the INFILE node specification. This is not recommended for security reasons.

- You did not specify the INPASS keyword.

Note

If the INFILE specifies a DECnet file specification, you can specify the INPASS value in the access control information portion of the INFILE node specification. This is not recommended for security reasons.

- MVS The TSO profile is set to PROMPT.

The DTF command processor prompts you to enter the password for the output file if the following requirements are met:

- You specified the OUTUSER keyword.

Note

If the OUTFILE specifies a DECnet file specification, you can specify the OUTUSER value in the access control information portion of the OUTFILE node specification. This is not recommended for security reasons.

- You did not specify the OUTPASS keyword.

Note

If the OUTFILE specifies a DECnet file specification, you can specify the OUTPASS value in the access control information portion of the OUTFILE node specification. This is not recommended for security reasons.

- MVS The TSO profile is set to PROMPT.

TRANSFER Subcommand Examples

The following example shows you how to use the TRANSFER subcommand to transfer an input and an output file specification set up with the DEFINE subcommand. The DEFINE subcommands are omitted from this example. They are entered prior to the TRANSFER subcommand.

```
TRANSFER
DTF0030I REQUEST SUBMITTED TO DTF1 [1]
DTF0037I THIS REQUEST ASSIGNED REQUEST NUMBER 00022
DTF0066I THE RECOVERABLE COPY REQUEST IS ABOUT TO BEGIN
DTF0062I THIS REQUEST ASSIGNED RETRY NUMBER 00151 [2]
DTF0067I SERVER NODE HANDLING THE TRANSFER IS NODE1 VERSION 3.1.0 [3]
DTF0068I CLIENT NODE INVOLVED IN TRANSFER IS NODE1 VERSION 3.1.0 [4]
DTF0055I SUBSYSTEM DTF1 REPORTS COMPLETION OF REQUEST
===> %SNADTF-S-COPIED, USER3$:<VMSUSER>LOGIN.COM;127 COPIED TO
===> NODE1"SNADTF"::"IBMUSER.LOGIN.COM/USER:IBMUSER/PASS:PASSWORD
===> /CHEC:1000" (73 RECORDS)
DTF0048I ENTER CLEAR, DEFINE, END, LIST, RECALL, RESUME, SET, SHOW, OR TRANS
END

READY
```

1. This message displays the name of the DTF subsystem that receives the transfer request.
2. This message is displayed after the OpenVMS client node accepts the request and assigns a retry number. This is written for a recoverable copy request only.
3. This message displays the name of the OpenVMS/DTF server node.
4. This message displays the name of the client node. The server node and client node are the same in this example.

9.3.5 HELP MVS

The HELP subcommand provides you with a list of DTF subcommands and a brief explanation of the function, syntax, and parameters of each subcommand.

The syntax of the HELP subcommand is:

{ HELP } { H }	{ CLEAR DEFINE END TRANSFER LIST RECALL RESUME SET }
-------------------	---

HELP Subcommand Examples

The following examples show common uses of the HELP subcommand.

Example 1: display a list of all DTF subcommands.

HELP

SUBCOMMANDS -

SUBCOMMANDS FOR INITIATING A NEW TRANSFER REQUEST:

CLEAR/C
DEFINE/DEF
TRANSFER/TRANS
LIST/L
RECALL/REC

SUBCOMMAND FOR RESTARTING A PREVIOUSLY FAILED TRANSFER:

RESUME/RES

SUBCOMMANDS FOR CONTROLLING DTF/TSO OPERATION:

END
HELP/H
SET

FUNCTION -

THE DTF/TSO COMMAND PROCESSOR ENABLES YOU TO TRANSFER FILES BETWEEN AN IBM PROCESSOR RUNNING MVS AND NODES ON THE DECNET NETWORK.

THE DTF/TSO COMMAND PROCESSOR USES SUBCOMMANDS TO INITIATE NEW TRANSFERS, RESUME PREVIOUSLY FAILED TRANSFERS, AND CONTROL HOW THE COMMAND PROCESSOR OPERATES.

INITIATING A NEW TRANSFER:

TO INITIATE A NEW TRANSFER, USE THE DEFINE SUBCOMMAND TO IDENTIFY THE FOLLOWING:

- 1) THE INPUT FILE FOR THE TRANSFER
- 2) THE OUTPUT FILE FOR THE TRANSFER
- 3) PARAMETERS THAT CONTROL HOW AND WHERE THE TRANSFER WILL BE HANDLED

THE DEFINE SUBCOMMAND USES KEYWORDS TO DESCRIBE EACH INPUT PARAMETER. TYPE 'HELP DEFINE' TO SEE MORE INFORMATION ABOUT THESE KEYWORDS.

AFTER YOU USE ONE OR MORE DEFINE SUBCOMMANDS TO ENTER ALL OF THE INFORMATION THAT PERTAINS TO THE TRANSFER, USE THE TRANSFER SUBCOMMAND TO INITIATE THE TRANSFER. TYPE 'HELP TRANSFER' TO SEE MORE INFORMATION ABOUT THE PROCESSING INVOLVED IN A TRANSFER SUBCOMMAND.

YOU CAN USE THE LIST, CLEAR, AND RECALL SUBCOMMANDS TO KEEP TRACK OF THE KEYWORDS YOU ENTERED WITH THE DEFINE SUBCOMMAND. HOWEVER, YOU DO NOT NEED TO USE THESE SUBCOMMANDS BEFORE YOU USE THE TRANSFER SUBCOMMAND. TYPE 'HELP' FOLLOWED BY LIST, CLEAR, OR RECALL TO SEE MORE INFORMATION ABOUT THESE SUBCOMMANDS.

RESTARTING A PREVIOUSLY FAILED TRANSFER:

IF A PREVIOUSLY INITIATED TRANSFER FAILED WITH A RECOVERABLE ERROR, YOU CAN USE THE RESUME SUBCOMMAND TO RESTART THAT TRANSFER FROM THE CHECKPOINT WHERE THE FAILURE OCCURRED. THE RESUME SUBCOMMAND USES A SET OF KEYWORDS TO IDENTIFY WHICH TRANSFER FAILED AND WHERE TO RESTART THE TRANSFER.

BECAUSE THE SERVER NODE RETAINS MOST OF THE INFORMATION FROM THE ORIGINAL TRANSFER, THE RESUME SUBCOMMAND DOES NOT USE ANY OF THE INFORMATION SUPPLIED WITH THE DEFINE SUBCOMMANDS. TYPE 'HELP RESUME' TO SEE MORE INFORMATION ABOUT RETRYING A FAILED TRANSFER.

SYNTAX -
DTF
REQUIRED - NONE
DEFAULTS - NONE

OPERANDS -
NONE

IKJ56804I FOR MORE INFORMATION ENTER THE HELP SUBCOMMAND NAME OR HELP HELP
DTF0047I DTF

Example 2: display information about the LIST subcommand.

HELP LIST

FUNCTION -
THE LIST COMMAND PROCESSOR DISPLAYS ALL DEFINED PARAMETERS

SYNTAX -
LIST
REQUIRED - NONE
DEFAULTS - NONE

OPERANDS -
NONE

DTF0047I DTF

9.3.6 LIST Subcommand

The LIST subcommand allows you to display all the DEFINE subcommands you have set up but have not transferred with the TRANSFER subcommand.

The syntax of the LIST subcommand is:

```
{ LIST }  
{ L }
```

The LIST subcommand will not display passwords. Instead, a representative number of asterisks (*) will be displayed to show the presence of a password.

LIST Subcommand Example

The following example shows how you use the LIST subcommand to check the values that were defined for a file transfer.

```
DEFINE INFILE(NODE::LOGIN.COM) INUSER(VMSUSER) INPASS(VMSPASS)
DTF0047I DTF
DEFINE OUTFILE(IBMUSER.LOGIN.COM)
DTF0047I DTF
DEF SERVER(NODE1)
DTF0047I DTF
LIST
DEF INFILE(NODE::LOGIN.COM) INUSER(VMSUSER) INPASS(*****)
DEF OUTFILE(IBMUSER.LOGIN.COM)
DEF SERVER(NODE1) NETMGR(DTF1)
DTF0047I DTF
```

The output of the LIST subcommand also allows you to type over minor errors in the file specification and enter corrections by pressing . You are limited, however, to one typeover for each DEFINE list.

9.3.7 RECALL Subcommand

The RECALL subcommand allows you to reuse all of the DEFINE subcommands you created immediately before you entered a TRANSFER subcommand. After you enter a TRANSFER subcommand, all DEFINE subcommands are erased and can no longer be displayed with the LIST subcommand.

The syntax of the RECALL subcommand is:

```
{ RECALL }
{ REC }
```

RECALL Subcommand Example

The following example shows how you use the RECALL subcommand if you start a transfer with the TRANSFER subcommand and the transfer fails because of the parameters you defined.

```
DEFINE INFILE(NODE::LOGIN.COM) INUSER(VMSUSER) INPASS(VMSPASS)
DTF0047I DTF
DEFINE OUTFILE(IBMUSER.LOGIN.COM)
DTF0047I DTF
DEFINE SERVER(NODE1) NETMGR(DTF1) RECOVERY(Y)
DTF0047I DTF
TRANSFER [1]
DTF0030I REQUEST SUBMITTED TO DTF1
====> DTF0612I A SERVER SESSION IS NOT AVAILABLE FOR THIS REQUEST [2]
====>
DTF0047I DTF
```

```

LIST [3]
  DEF NETMGR(DTF1) RECOVERY (Y)
  DTF0047I DTF
RECALL [4]
  DTF0047I DTF
LIST [5]
  DEF INFILE(NODE::LOGIN.COM) INUSER(VMSUSER) INPASS(*****)
  DEF OUTFILE(IBMUSER.LOGIN.COM)
  DEF SERVER(NODE1) NETMGR(DTF1) RECOVERY(Y)
  DTF0047I DTF
DEFINE SERVER(NODE2) [6]
TRANSFER [7]

```

1. This TRANSFER subcommand started the transfer.
2. The transfer failed because an active server session was not available.
3. This LIST subcommand shows the default values only.
When a transfer request fails, DTF automatically issues a CLEAR command to erase the input and output file specifications and to reset the default values.
4. This RECALL subcommand calls back all the values defined prior to the last TRANSFER subcommand.
5. This LIST subcommand displays the input and output file specifications and the transfer parameters.
6. The DEFINE command corrects the error (in this case, by specifying SERVER and a keyvalue that specifies an active server session).
7. Another TRANSFER subcommand resubmits the transfer.

9.3.8 RESUME Subcommand

The RESUME subcommand allows you to restart a previously-failed recoverable transfer from the point of failure. Its function is similar to the TRANSFER subcommand. However, the RESUME subcommand does not use information provided by the DEFINE subcommands. Instead, all information must be provided as part of the RESUME subcommand.

The syntax of the RESUME subcommand is:

```

{ RESUME }   RETNUM (retry-number)   { CLIENT
RES         }                         { CLIENTPASS
}                         { CLIENTUSER
}                         { NETMGR
}                         { SERVER
}

```

The minimum entry for the RESUME subcommand includes the RETNUM, NETMGR, and SERVER keywords.

The RESUME subcommand passes the information supplied by the keywords to the DTF Network Manager subsystem. The subsystem selects an OpenVMS /DTF server node to process the resume request and forwards the information from the keywords to that node. For the duration of the retry, the terminal keyboard is locked.

Several messages are written to the terminal to indicate the status of the retry request. The following messages are the most important:

```
DTFM037I THIS REQUEST ASSIGNED REQUEST NUMBER xxxxx
```

The xxxxx value indicates the internal request number assigned by the DTF Network Manager subsystem. You use this number when you report problems to the IBM system support staff. They use this number to locate a variety of trace information.

```
DTFM055I SUBSYSTEM yyyy REPORTS COMPLETION OF REQUEST
```

Processing is complete. Messages from the server node report the outcome of the retry request. *Digital SNA Data Transfer Facility for OpenVMS Problem Solving and Messages* provides additional information about OpenVMS/DTF messages from the server node.

Canceling a RESUME Subcommand

[MVS] You can cancel a resumed transfer that is in progress at any time. The method used to cancel the transfer depends on the terminal you are using. On non-SNA terminals, press **[Reset]**, then **[PA1]**. On SNA terminals, press **[Attn]**. Both terminals will display the following:

```
CANCEL REQUEST (YES/NO)
```

Take one of the following actions:

- Type YES and press **[Enter]** to cancel the request and free the terminal.
- Type NO and press **[Enter]** to continue processing.

If you respond with YES, the DTF processor halts the transfer at the current position. You must then clean up the partial files. You cannot issue a RESUME subcommand to restart a canceled request.

On VM systems, you can cancel a transfer that is in progress at any time by typing the command CANCEL and pressing . You can request a status report for a transfer in progress by typing the command QUERY and pressing .

The following sections describe the required and optional keywords associated with the RESUME subcommand.

RETNUM(*retry-number*)

The number returned in message DTF0062I when the original request was submitted.

retry-number

The number returned by the DTF processor in the DTF0062I message. Length: 5 digits. Range: 1 to 32767.

CLIENTUSER(*decnet-user-id*)

Specifies the user ID for the DECnet file specification involved in the original TRANSFER request.

decnet-user-id

The user ID for the DECnet file specification involved in the original TRANSFER request. Length: 1 to 40 characters.

Usage Notes:

- The CLIENTUSER keyword should not be used in the following situations:
 - If the original file transfer involved two IBM file specifications
 - If the original request involved a DECnet file that had no user ID associated with it
- The CLIENTUSER keyword will be ignored in the following situations:
 - If the CLIENT keyword is not used
 - If the CLIENT keyword value has access control information within it

CLIENTPASS(*password*)

Specifies the password for the DECnet file specification involved in the original transfer request.

password

The password for the DECnet file specification involved in the original transfer request. Length: 1 to 31 characters.

Usage Notes:

- If you specify the CLIENTPASS keyword, the password will be visible on the screen. A better alternative is to let DTF prompt you for the password, which you then enter in a nondisplay field.
- The DTF processor may prompt you to enter a password after you issue the RESUME subcommand.
- The DTF processor prompts you to enter a client password under the following circumstances:
 - If you specified the CLIENT keyword
 - If you specified the CLIENTUSER keyword or specified the client user ID in the access control information portion of the CLIENT keyword
 - If you did not specify the CLIENTPASS keyword or the client password in the access control information portion of the CLIENT keyword
 - If the TSO profile is set to PROMPT
- The CLIENTPASS keyword should not be used in the following situations:
 - If the original request involved two IBM file specifications
 - If the original request involved a DECnet file specification that had no password associated with it
- The CLIENTPASS keyword will be ignored in the following situations:
 - If you did not specify the CLIENT keyword
 - If you specified the CLIENT keyword with access control information that contains a password
 - If you did not specify the CLIENTUSER keyword and the CLIENT keyword contains no access control information

CLIENT(*node*)

Identifies the client node that was specified for the DECnet file involved in the original TRANSFER subcommand.

node

The node portion of the DECnet file specification that was involved in the original TRANSFER subcommand. Length: 1 to 64 characters.

Usage Notes:

- The CLIENT keyword should not be used if the original request involved two IBM files.
- If you specify the CLIENT keyword but do not specify the node keyvalue, the DTF command processor will use the SERVER value for the CLIENT keyvalue. In this case, no CLIENTUSER or CLIENTPASS keyvalues will be associated with the CLIENT keyword.
- The node specification can contain OpenVMS access control information; however, this is not recommended because the DECnet password will be visible on the screen. The recommended method of specifying access control information is to use the CLIENTUSER keyword and let DTF prompt you for the password.

NETMGR(component-name)

Specifies the name of the DTF Network Manager component. If you specify a name, the value overrides the name in the DTFUSER parameter file.

component-name

On VM systems, the name is the 1- to 8-character virtual machine name. On MVS systems, the name is the 1- to 4-character subsystem name as it appears in the subsystem name table.

SERVER(server-name)

This name specifies the OpenVMS/DTF server to which DTF sends the transfer request. SERVER is the name of a DECnet node or TCP/IP host.

server-name

The 1- to 6-character node name of the OpenVMS/DTF server node you wish to handle the transfer request.

RESUME Subcommand Example

The following example shows you how to use the RESUME subcommand to restart a failed transfer request.

```
DEFINE INFILE(NODE1:<JONES.DTF.TEST>TEST123.RES) INUSER(JONES) INPASS(*****) [1]
DEFINE OUTFILE(JONES.DTFTEST(MEMBER)) [2]
DEFINE SERVER(NODE1) NETMGR(DTF2) RECOVERY(Y) [3]
DTF0048I ENTER CLEAR, DEFINE, END, LIST, RECALL, RESUME, SET, SHOW, OR TRANS
```

```

TRANSFER
DTF0030I REQUEST SUBMITTED TO DTF2
DTF0037I THIS REQUEST ASSIGNED REQUEST NUMBER 00014
DTF0066I THE RECOVERABLE COPY REQUEST IS ABOUT TO BEGIN
DTF0062I THIS REQUEST ASSIGNED RETRY NUMBER 00051 [4]
DTF0067I SERVER NODE HANDLING THE TRANSFER IS NODE1 VERSION 3.1.0 [5]
DTF0068I CLIENT NODE INVOLVED IN TRANSFER IS NODE1 VERSION 3.1.0 [6]

===> DTF0630E RECOVERABLE ERROR REPORTED FROM SERVER NODE1 [7]
===>
===> %SNADTF-E-RCOPYCOM, error communicating with TRANSFER/DTF process
===> -SYSTEM-F-THIRDPARTY, network logical link disconnected by a third party
DTF0048I ENTER CLEAR, DEFINE, END, LIST, RECALL, RESUME, SET, SHOW, OR TRANS

RESUME RETNUM(00051) CLIENT(NODE1) CLIENTUSER(JONES) CLIENTPAS(BOBJONES)
SERVER(NODE1) NETMGR(DTF2) [8]
DTF0030I REQUEST SUBMITTED TO DTF2
DTF0037I THIS REQUEST ASSIGNED REQUEST NUMBER 00016
DTF0066I THE RECOVERABLE COPY REQUEST IS ABOUT TO BEGIN
DTF0062I THIS REQUEST ASSIGNED RETRY NUMBER 00054
DTF0067I SERVER NODE HANDLING THE TRANSFER IS NODE1 VERSION 3.1.0
DTF0068I CLIENT NODE INVOLVED IN TRANSFER IS NODE1 VERSION 3.1.0
DTF0055I SUBSYSTEM DTF2 REPORTS COMPLETION OF REQUEST [9]
===> %SNADTF-S-COPIED, NODE1::<JONES.DTF.TEST>TEST123.RES;17 copied to
===> NODE1"SNADTF"::"JONES.DTFTEST(MEMBER)/USER:JONES/CORR:00047/PASS:password
===> /CHEC:1000" (4583 records)
DTF0048I ENTER CLEAR, DEFINE, END, LIST, RECALL, RESUME, SET, SHOW, OR TRANS

```

1. This DEFINE statement specifies the input file.
2. This DEFINE statement specifies the output file.
3. This DEFINE statement specifies the transfer parameters.
4. This message reports that the transfer was assigned retry number 00051.
5. This message reports that the server node handling the transfer is NODE1.
6. This message reports that the client node involved in the transfer is NODE1.
7. This message reports that a recoverable error was reported from server NODE1. Additional OpenVMS/DTF error messages report what caused the recoverable error.
8. This RESUME command restarts the transfer.
9. This message reports that the transfer was successfully completed.

9.3.9 SET Subcommand

The SET subcommand allows you to change the way the DTF command processor operates. SET alters the parameters described in the following sections for the duration of the DTF command processor session, or until you alter the parameter with another SET command. These same parameters that you change with the DEFINE command stay in effect for one transfer.

The default value for this subcommand is set during installation of the DTF for IBM software. The *Digital SNA Data Transfer Facility Installation (VM)* or *Digital SNA Data Transfer Facility Installation (MVS)* manual provides additional information about how to establish the default.

CONVERSE { (LONG)
(SHORT) }

Gives you the option to display full or abbreviated messages.

LONG

Full messages are displayed.

SHORT

Abbreviated messages are displayed.

DEBUG { (ON)
(OFF) }

Gives you the option to display and mask DTF processor messages.

ON

All DTF command processor messages are displayed.

OFF

Non-essential DTF command processor messages are masked.

NULL { (SPACE)
(REJECT)
(IGNORE)
(NONE) }

This keyword is used to indicate how DTF processes null records (length = 0) in an IBM environment. Because the CMS file system does not support null length records, DTF cannot write them to the indicated device. For more

information about the NULL option see the /NULL qualifier description in Chapter 2.

SPACE

A 1-byte record is inserted in place of the zero length record. The one byte is set to an EBCDIC space character (x'40').

REJECT

The file transfer is rejected with an invalid record length message.

IGNORE

The null record is simply ignored when writing the file. The result is that the output file has fewer records than the input file.

NONE

- VM Null records cannot be written to CMS files. If a null record is encountered, an error will be returned.
- MVS The record will be accepted as is.

RECOVERY { (YES) } { (NO) }

A required keyword that specifies whether the transfer is executed in recoverable mode or nonrecoverable mode. Chapter 1 provides additional information about recoverable copy.

YES

The transfer executes in recoverable mode.

NO

The transfer executes in nonrecoverable mode.

SERVER(*node*)::

Selects the server session to which the transfer request is sent for execution.

node

The node name of the node running the OpenVMS/DTF server software.

NETMGR(*dtf-name*)

A required keyword that specifies which DTF Network Manager subsystem should receive the transfer request.

The syntax of the SET subcommand is:

```
SET    CONVERSE  { LONG  }
                   { SHORT }
```

SET Subcommand Examples

The following examples show common uses of the SET subcommand.

Example 1: display the short form of the DTF prompt.

```
SET CONVERSE(SHORT)
DTF0047I DTF
```

Example 2: display the long form of the DTF prompt.

```
SET CONVERSE(LONG)
DTF0048I ENTER CLEAR, DEFINE, END, LIST, RECALL, RESUME, SET, SHOW, OR TRANS
```

9.3.10 SHOW Subcommand

The SHOW subcommand is used to show the defaults used by the command processor and any values which may have been changed through use of the SET subcommand.

No options are available on the SHOW subcommand.

This is the syntax of the SHOW subcommand:

```
SHOW
```

SHOW Subcommand Example

```
DTF0048I ENTER CLEAR, DEFINE, END, LIST, RECALL, RESUME, SET, SHOW, OR TRANS
SHOW
SET NETMGR(DTF4) DEBUG(OFF) RECOVERY(N) CONVERSE(L) NULL(REJECT)
SET VERSION(3.1.0)
DTF0048I ENTER CLEAR, DEFINE, END, LIST, RECALL, RESUME, SET, SHOW, OR TRANS
```

9.4 DTF Command Processor Usage Considerations

As you use the DTF command processor, you should be aware of certain considerations which involve using tapes and overriding the server node and server account.

The following sections address these considerations.

9.4.1 Using Tapes (MVS Systems)

The DTF Network Manager subsystem may not allow unmounted tapes. This is an installation-dependent option. A tape can be mounted using the following MVS command at an MVS console:

```
M unit,VOL=(SL,volser),USE=PUBLIC
```

Only standard label tapes can be mounted.

9.4.2 Using Tapes (VM Systems)

When you copy a file to or from a tape, DTF prompts the user ID specified in the TAPEOPER parameter to attach a tape drive to that ID. The transfer continues once the tape drive is attached.

9.4.3 Overriding the Server and Server Account

If the INRMT, the OTRMT, or both keywords are used to override the server node or server account, you should be aware of the following:

- The DTF Network Manager subsystem that accepts the transfer request may not be the one that executes the file transfer. If this happens, there will be no logging information about the file transfer in the DTF Network Manager subsystem that did not execute the transfer.
- The OpenVMS/DTF proxy database is stored on a server node. If the server node is overridden to a different node, the proxy database on the different node will be used.
- The OpenVMS/DTF file definition database is stored on a server node. If the server node is overridden, the file definition database on the different node will be used.

9.5 Executing the DTF Command Processor in Batch Mode (VM Systems)

There are two ways to initiate batch transfers in CMS:

- Execute an exec file that stacks DTF subcommands before invoking DTF. For example:

```
Queue 'def netmgr(dtfgcs) server(dtfsrv)'  
Queue 'def infile(profile exec a)'  
Queue 'def outfile(vmssys::a.b)'  
Queue 'def outuser(jones) outpass(secret)'  
Queue 'transfer '  
Queue 'end'  
  
r = dtf()
```

```
return r
```

- Submit the job to a CMS batch machine through the panel interfaces.

The batch machine must have the authority required to link and access the minidisks needed during the transfer. For more information about minidisk passwords, see the /MDxPASSWORD qualifier in Section 2.3.3.

9.6 Executing the DTF Command Processor in Batch Mode (MVS Systems)

To execute the DTF command processor in batch mode, invoke the TSO Terminal Monitor Program through JCL and DTF command processor subcommands, which are supplied in a sequential file.

Batch mode operations are useful for the following types of situations:

- For transferring large files that lock the TSO terminal for a long period of time
- In production environments where the same files are regularly moved and can be controlled by a scheduling system through the use of JCL procedures
- For transfers that require a log (a JES job log can retain transfer information)

The IBM MVS manuals *SPL: TSO User's Guide* or *SPL: TSO/E User's Guide* describe how to invoke the TSO Terminal Monitor Program and explain any TSO-enforced restrictions.

9.6.1 The SYSTSIN DD Card

The SYSTSIN DD card points to a sequential file that supplies the DTF subcommands. The sequential file must adhere to the following guidelines:

- Any record in the file can contain the DTF command. However, that record must precede the record containing the first DTF subcommand.
- The record that contains the DTF command cannot contain any DTF subcommands. All subsequent DTF subcommands must occupy separate records.
- A DTF subcommand (such as DEFINE) could have several keywords, thereby using more characters than a single JCL card can accept (assuming 80 characters as the norm). If the subcommand and its keywords and keyvalues do not fit on a single JCL card, you can use standard TSO line continuation rules and continue the subcommand string on the next card.

The IBM manuals *TSO Command Language Reference* or the *TSO/E Command Language Reference* explain how to use line continuation. The manuals also explain how to enter comments in the SYSTSIN file records.

- The SYSTSIN file can contain other TSO commands or command processors. However, these must either precede the record containing the DTF command or follow the record containing the DTF END subcommand.

The following example shows three files, each with records that contain the DTF command, DTF subcommands, and a TSO command. The first file is acceptable; the last two are not.

	File 1	File 2	File 3
Record 1	DTF	DTF DEFINE	DTF
Record 2	DEFINE		Any TSO command
Record 3	TRANSFER		DEFINE

File 1 is acceptable. The first record contains the DTF command, the DTF command precedes the DTF subcommands, and the DTF subcommands are in separate records.

File 2 is not acceptable. Record 1 contains both the DTF command and a DTF subcommand.

File 3 is not acceptable. Although Record 1 contains the DTF command, it is followed by a TSO command. The TSO command must be either in a record that precedes the record containing the DTF command or in a record that follows the record containing the DTF END subcommand.

Example 9–1 Sample REXX EXEC to Perform Post-Processing

```
/* DTF for VM Batch Job with post processing */
Queue 'def netmgr(dtf4) server(boston)'
Queue 'def infile(payroll data a)'
Queue 'def outfile(boston::a.a)'
Queue 'def outuser(smith) outpass(secret)'
Queue 'def recovery(y)'
Queue 'transfer post(post.com)'
Queue 'end'
rc = dtf(batc)

return rc
```

This is an example REXX EXEC that shows how the post-processing feature may be used. The local file PAYROLL DATA on the user's A disk is being transferred to the remote OpenVMS node named BOSTON. When the transfer completes successfully, the DCL command procedure named POST.COM will be submitted on the remote node.

Example 9–2 Sample JCL to Perform a Post-Processing Function

```
//DTFTESTB JOB 1,'DTFTEST',CLASS=A,MSGCLASS=X
//          EXEC PGM=IKJEFT01
//STEPLIB DD DISP=SHR,DSN=DTFTLOAD
//DTFUSER DD DISP=SHR,DSN=DTFIBM.DTFUSER.PARMS
//SYSTSPRT DD SYSOUT=*
//SYSLBL DD DSN=SYS1.BROADCAST,DISP=SHR
//SYSHELP DD DSN=IP01.HELP,DISP=SHR
//SYSTSIN DD *
DTF
/* Start the recoverable transfer */
DEF NETM(DTF) SERVER(BOSTON)
DEF RECOV(Y)
DEF INFILE(PAYROLL.DATA)
DEF OUTFILE(BOSTON"SMITH SECRET"::A.A)
LIST
TRANSFER POST(POST.COM)
END
//
```

This is a sample batch job that demonstrates how the post-processing feature may be used. In this case the local file PAYROLL.DATA is being transferred to the remote OpenVMS node named BOSTON. When the transfer completes successfully, the DCL command procedure named POST.COM will be submitted on the remote node.

Example 9–3 Sample REXX EXEC to Automatically Recover a Transfer

```

/* VM/DTF Batch Job */
Queue 'def netmgr(dtf4) server(vmsland)'
Queue 'def infile(profile EXEC a)'
Queue 'def outfile(vmsland::a.a)'
Queue 'def outuser(lindsey) outpass(secret)'
Queue 'def recovery(y)'
Queue 'transfer '
Queue 'end'

rc = dtf(batc)

/* If we had a recoverable error, try to resume the transfer */
if rc = 2 then
do
  /* Retrieve values saved by DTF */
  'globalv select dtfresum get $retnum $client $clientuser $clientpass'

  /* Try again in 10 minutes      */
  'CP SLEEP 10 MIN'

  /* Resume the transfer          */
  Queue 'set netmgr(dtf4) server(vmsland)'
  Queue 'resume retnum('$retnum') client('$client') ',
        'clientuser('$clientuser') clientpass('$clientpass')'
  Queue 'end'
  rc = dtf(batc)
end

return rc

```

This is an example of REXX EXEC that demonstrates how a failed recoverable transfer may be resumed.

Usage Notes:

- If the transfer fails with a recoverable error, the retry number, client node, user, and password are all stored using CMS's GLOBALV facility.

- The return code from the transfer may be interrogated to determine if the transfer failed and if it may be recovered.

SYSTSIN File:

- The example shows a set of DTF subcommands used to copy an IBM file to two different OpenVMS client nodes.
- The DEFINE subcommands are written with continuation characters and comments to make the procedure more readable.

9.6.2 Notes and Restrictions for DTF Batch Mode Operations

The following restrictions apply when DTF is used in batch mode:

- **Multiple error messages returned:**
Whenever possible, DTF continues processing subcommands even after an error occurs. If multiple errors occur, the return code passed back to TSO indicates the most severe error that has occurred.
- **Information missing from TRANSFER and RESUME subcommands:**
If a TRANSFER or RESUME subcommand is rejected because that subcommand is missing parameters (return code 4), DTF ceases operations in batch mode and immediately returns to TSO. TSO then processes any remaining records within the SYSTSIN file. If these records contain DTF subcommands, TSO issues an error message for each record and then rejects the subcommand.
- **Internal error conditions:**
An internal error condition in DTF (return code 12) causes DTF to cease operations in batch mode and immediately return to TSO. TSO then processes any remaining records within the SYSTSIN file. If these records contain DTF subcommands, TSO issues an error message for each record and then rejects the subcommand.
- **Missing passwords:**
DTF does not prompt for missing passwords in batch mode.
- **Lower case Digital UNIX or ULTRIX file specifications, user IDs, and passwords:**
Be careful when you use DTF in batch mode to access Digital UNIX or ULTRIX nodes. Digital UNIX or ULTRIX file specifications, user IDs, and passwords must all be entered in lowercase. Since IBM JCL requires uppercase for control cards, files are usually edited with the uppercase option (in the ISPF editor this is CAPS ON in the edit profile). The editor option must be reset to allow the use of lowercase characters when you enter the DTF DEFINE subcommands.

9.7 DTF Return Codes in DTF Batch Mode Operations

The DTF command processor returns with a completion code that reflects the most severe error encountered.

[MVS] TSO uses this completion code as the job step completion code. If other TSO commands are included within the SYSTSIN file, TSO uses the most severe error encountered in all TSO commands, including the DTF completion code, as the job step completion code.

The following DTF subcommands change the completion code:

- **DEFINE:** sets only a return code of 1 or 0
- **TRANSFER:** sets all return codes except 1
- **RESUME:** sets all return codes except 1

Note

If an internal error occurs during the processing of any subcommand, a return code of 12 is generated.

Return codes from the TRANSFER subcommand are grouped into the following ranges:

- 0—3** The transfer was partially successful.
- 4—7** The transfer could not begin. Cleanup of partial files is not necessary.
- 8—12** The outcome of the transfer request is in doubt. Some investigation must be done to determine the state of the transfer. Cleanup of partial files is necessary.

A complete explanation of all return codes and suggested user action is contained in an appendix of the *Digital SNA Data Transfer Facility for OpenVMS Problem Solving and Messages* manual.

9.8 Initiating DTF Command Processor Operations from Command Procedures

The DTF command processor can be invoked from a TSO CLIST or a CMS REXX exec. The DTF subcommands should be entered immediately following the DTF command. All records following the DTF command will be treated as input records to DTF until an END subcommand is found.

When the DTF command processor encounters an END subcommand, it returns control to the command procedure with the condition code set. A list of return codes is included in Section 9.7.

9.8.1 Variables Set by DTF

DTF sets the values of CLIST and CMS global variables by taking the appropriate values from the DEFINE subcommands that are in effect when a TRANSFER subcommand is processed. These values are the key pieces of information that DTF requires to resume transfers that fail with a recoverable error.

DTF will set the variables under the following conditions:

- TSO/E or CMS was used to invoke the DTF command processor
- A TRANSFER subcommand was issued to initiate the transfer request
- The transfer was executed in recoverable mode
- The OpenVMS client node accepted the transfer request

Under these conditions DTF will set the following variables:

- \$RETNUM
- \$CLIENT
- \$CLIENTUSER
- \$CLIENTPASS

The following sections describe these variables.

\$RETNUM

Contains a value that can later be used with the RESUME subcommand and the RETNUM keyword to restart the transfer from the last successful checkpoint before the transfer failed.

\$CLIENT

Contains a value that can later be used with the RESUME subcommand and the CLIENT keyword to restart the transfer from the last successful checkpoint before the transfer failed. Cleared for IBM-to-IBM transfers.

\$CLIENTUSER

Contains a value that can later be used with the RESUME subcommand and the CLIENTUSER keyword to restart the transfer from the last successful checkpoint before the transfer failed. Cleared for IBM-to-IBM transfers.

\$CLIENTPASS

Contains a value that can later be used with the RESUME subcommand and the CLIENTPASS keyword to restart the transfer from the last successful checkpoint before the transfer failed. Cleared for IBM-to-IBM transfers.

10

ULTRIX and UNIX File Transfer Operations

This chapter describes how to use the DECnet-ULTRIX and DECnet/OSI for Digital UNIX file transfer commands with DTF software to transfer files between ULTRIX or Digital UNIX DTF clients and IBM DTF clients. You can use the DECnet-ULTRIX and DECnet/OSI for Digital UNIX commands supported by DTF at any ULTRIX or Digital UNIX node in a DECnet network so long as that node can access the OpenVMS/DTF server node. You simply include the OpenVMS/DTF server node as part of the file specification. No DTF software is required at the ULTRIX or Digital UNIX client node.

10.1 Transferring Files Between ULTRIX, UNIX, and IBM DTF Clients

You can use the following DTF-supported DECnet-ULTRIX or UNIX commands with an IBM file specification:

- **dcat** - to display the contents of IBM data sets
- **dcp** - to copy, append, or submit files
- **dls** - to display the contents of an IBM directory
- **drm** - to remove an IBM data set

10.2 dcat Command

To display the contents of one or more files on an IBM DTF client, use the DECnet-ULTRIX or DECnet/OSI for Digital UNIX **dcat** command:

dcat [-option] *file-spec-1 file-spec-2 file-spec-n*

Table 10-1 shows the level of support DTF provides for the DECnet-ULTRIX **dcat** command options. For more detailed information about the **dcat** command, refer to the *DECnet-ULTRIX Use* manual or the *DECnet/OSI for Digital UNIX Introductory User's Guide*.

Table 10–1 DTF Level of Support for dcat Command Options

Option	For IBM File Specifications	Comments
-v	supported	

You cannot specify wildcard characters in an IBM file specification.

Examples

MVS

```
% dcat veena/snadtff::"mike.dtf(test)"
```

This command displays the member **test** from the PDS called **mike.dtf** located on an IBM DTF client. The DTF server node is **veena** and the server account is **snadtff**. Note that no server account password is given. The **dcat** command will prompt you for a password. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

MVS

```
% dcat veena/dfmvs::"ward.dtf/vol:tsov2/user:ward/pass:hunter" >output.txt
```

This command displays the data set **ward.dtf** located on the IBM DTF client on volume **tsov2**. The OpenVMS/DTF server node is **veena** and the server account is **dfmvs**. The display output is directed to the file **output.txt**. The MVS user ID is **ward** and the password is **hunter**.

MVS

```
% dcat veena/dfmvs//::"mike.basin.txt/userid:mike/password:secret"
```

This command displays the data set **mike.basin.txt**, located on the IBM DTF client under the user ID **mike**, to the standard output. Note that in the IBM file specification the options **userid** and **password** are given. This serves to override the default account set up for the account **dfmvs** and the user **mike** on the server **veena**. Also, the password for the server account **dfmvs** is null as indicated by the double slashes (/).

VM

```
% dcat dtfsrv/dfvm//::"tuesday orders"
```

This command displays the file **tuesday orders**, to the standard output. The VM user ID and password are taken from the OpenVMS/DTF server's proxy database. The password for the server account **dtfvm** on OpenVMS/DTF server node **dtfsrv** is null as indicated by the double slashes (/).

10.3 dcp Command

To transfer ASCII or binary files between a DECnet-ULTRIX or DECnet/OSI for Digital UNIX node and an IBM DTF client, use the DECnet-ULTRIX or DECnet/OSI for Digital UNIX **dcp** command:

dcp [-option] *input-file-spec output-file-spec*

The **dcp** command allows an ULTRIX or UNIX user to do the following:

- Copy an ASCII or binary file residing on an IBM DTF client to any location in a DECnet network.
- Copy an ASCII or binary file on a DECnet network to any IBM DTF client.
- Copy an ASCII or binary file residing on an IBM DTF client to any other IBM DTF client.
- Append the specified input file to the specified output file. Be sure that the output file you specify already exists.

Note

DTF does not support append operations to files with RECFM=FBS. The request will be rejected.

- Copy a file to an IBM DTF client and submit that file to the batch system on that IBM system.

Note

Files submitted to IBM batch subsystems must not contain any records that exceed 80 bytes.

If a remote file has additional attributes besides ASCII or binary, the file will lose those attributes when transferred to an ULTRIX or UNIX system.

Table 10–2 shows the level of support DTF provides for the DECnet–ULTRIX or DECnet/OSI for Digital UNIX **dcp** command options. For more detailed information about the **dcp** command, refer to the *DECnet–ULTRIX Use manual* or the *DECnet/OSI for Digital UNIX Introductory User's Guide*.

Table 10–2 DTF Level of Support for dcp Command Options

Option	For IBM File Specifications	Comments
-i	supported	
-a	supported	
-A	supported	
-S	supported	File should not contain any records that exceed 80 bytes. The file will be deleted after the submitted job completes.
-v	supported	

Examples

MVS

```
% dcp veena/dtfmvs::"gary.dtf.file.transfer" /usr/users/gary/dtf.file
```

This command transfers the file **gary.dtf.file.transfer**, located on the IBM DTF client to the output file **/usr/users/gary/dtf.file** located on the local DECnet–ULTRIX or DECnet/OSI for Digital UNIX node. The OpenVMS /DTF server name is **veena** and the server account is **dtfmvs**. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

MVS

```
% dcp -A veena/dtfmvs//::"mike.dtf" veena/snadt//::"lendl.dtf.append"
```

This command appends the file **mike.dtf** located on the IBM DTF client to the file **lendl.dtf.append** on the IBM DTF client. The input file is accessed through the server account **dtfmvs** on the OpenVMS/DTF server node **veena**. The output file is accessed through server account **snadt** on the OpenVMS /DTF server node **veena**. Server accounts **snadt** and **dtfmvs** can point to different IBM DTF clients or to the same IBM DTF client. No password prompt will appear after this command is entered because two slashes (//) have replaced the password.

VM

```
% dcp -i binary.a dtfsrv/spot//::"a exec/user:richards/pass:guessit/notrans"
```

This command copies the file **binary.a** on the ULTRIX system to the file **a exec** on the VM system. The OpenVMS/DTF server node is **dtfsrv** and the server account is **spot**. The two slashes indicate that no password prompt will be generated. The VM user ID is **richards** and the user password is **guessit**. Note the use of the **/notrans** qualifier to indicate that no data translation should occur because the file is an image file.

VM

```
% dcp -S vmlogin.com dtfsrv/spot//::"login exec/user:richards/pass:guessit"
```

This command copies the file **vmlogin.com** on the ULTRIX system to the file **login exec** on the VM system. The OpenVMS/DTF server node is **dtfsrv** and the server account is **spot**. The two slashes indicate that no password prompt will be generated. The VM user ID is **richards** and the user password is **guessit**. Note the use of the **-S** option. This causes the file to be submitted to the VM batch machine for execution after the file is closed. Note that the file **login exec** should not contain any records exceeding 80 bytes. After the file is executed it will be deleted from the VM system.

10.4 Remote Backup and Restore Operations

To backup and restore files on an IBM DTF client system, use the DECnet-ULTRIX or DECnet/OSI for Digital UNIX **dcp -i** command:

dcp -i [-option] file-spec

Examples

MVS

```
% dls -l veena/dtfmvs::"gary.dtf.file(*)"
```

```
% tar cvf - filenames | dcp -i - boston/SNADTF//::'dtftest.tar/notrans/
```

```
% dcp -i boston/SNADTF//::'dtftest.tar/notrans/- | tar tf -
```

These examples show how to back up and restore ULTRIX or UNIX files on IBM disks. If you would like to do a backup on IBM tape devices, use the appropriate **/UNIT** and **/VOLUME** qualifiers in the IBM file specification. The examples show how to pipe the output from **tar** to **dcp**, which routes the transfer to DTF for IBM systems. Similarly using **dcp** you can pipe the output to **tar**, which can display or restore files archived using **tar**. Note that you must use the **/NOTRANSLATE** qualifier in the IBM file specification.

10.5 dls Command

To display a file list on an IBM DTF client, use the DECnet-ULTRIX or DECnet/OSI for Digital UNIX **dls** command:

dls [-option] file-spec

Table 10-3 shows the level of support DTF provides for the DECnet-ULTRIX or DECnet/OSI for Digital UNIX **dls** command options. For more detailed information about the **dls** command, refer to the *DECnet-ULTRIX Use manual* or the *DECnet/OSI for Digital UNIX Introductory User's Guide*.

Note

You can use the **dls** command to obtain directory listings of IBM tape-resident files.

Table 10-3 DTF Level of Support for dls Command Options

Option	For IBM File Specifications	Comments
-1	supported	
-a	supported	
-C	supported	
-l	supported	

Examples

MVS

```
% dls -1 veena/dtfmvs::"gary.dtf.file(*)"
```

This command displays the names of all the members of the PDS called **gary.dtf.file** located on an IBM DTF client. The OpenVMS/DTF server node is **veena** and the server account is **dtfmvs**. The display is in a single column format (-1). Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

MVS

```
% dls -1 veena/snadtf::"betty.dtf.*/vol:tsov2"
```

This command displays all data sets beginning with **betty.dtf**, located on the IBM DTF client, on volume **tsov2**. The OpenVMS/DTF server node is **veena** and the server account is **snadt**. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

VM

```
% dls -l dtfsrv/DTFVM//::"a* */user:white/pass:gray"
```

```
Directory dtfsrv::
ACCOUNTS DATA 1/USER:WHITE/PASS:GRAY          4096
AMERICA DATA 1/USER:WHITE/PASS:GRAY           512
AMOUNTS DATA 1/USER:WHITE/PASS:GRAY          3584
ANIMALS DATA 1/USER:WHITE/PASS:GRAY           512
4 files in 1 directory
```

This command displays all files beginning with **a**, located on the VM system. The OpenVMS/DTF server node is **dtfsrv** and the server account is **dtfvm**.

10.6 drm Command

To remove a file on an IBM DTF client, use the DECnet-ULTRIX or DECnet/OSI for Digital UNIX **drm** command:

drm [-option] file-spec

DTF does not provide support for any DECnet-ULTRIX or DECnet/OSI for Digital UNIX **drm** command options.

Notes

If you attempt to delete an MVS file that was created with an expiration date and that expiration date has not been reached, MVS will prompt the MVS console operator for permission to delete the file. The file operation will wait until the console operator replies.

You cannot delete IBM tape-resident files.

You cannot use wildcard characters in IBM file specifications when using the **drm** command.

Examples

MVS

```
% drm veena/dtfmvs::"richards.dtf.file/user:richards/pass:snowy"
```

This command removes the data set **richards.dtf.file** on the IBM DTF client. The OpenVMS/DTF server node is **veena**, and the server account is **dtfmvs**. The MVS user ID is **richards** and the password is **snowy**.

MVS

```
% drm veena/dtfmvs::"eric.dtf/vol:tsov2"
```

This command removes the data set **eric.dtf** on the IBM DTF client. The file is on volume **tsov2**. The OpenVMS/DTF server node is **veena**, and the server account is **dtfmvs**. Account verification on the MVS system is installation dependent. Contact your systems administrator for information on the /USERID and /PASSWORD qualifiers.

VM

```
% drm dtfsrv/dtfvm//::"test dtf/user:james"
```

This command removes the data set **test dtf** on the VM system. The OpenVMS/DTF server node is **dtfsrv**, and the server account is **dtfvm**. The VM user ID is **james** and the IBM password is taken from the OpenVMS/DTF server's proxy database.

10.7 Error Reporting

Most error messages consist of a primary message and a secondary message followed by a secondary status value (STV). The primary and secondary messages are usually self-explanatory. STV codes are reported as hexadecimal numbers.

Messages appear in a format similar to the following:

```
command: primary-text  
secondary-text  
Secondary Status = 0X"hex-number"
```

where *command* is the command that caused the error, *primary-text* is the primary error message, *secondary-text* is the secondary error message, and *hex-number* is a hexadecimal number.

Refer to the *DECnet-ULTRIX Use* manual for help in understanding the primary and secondary error messages. Refer to the *Digital SNA Data Transfer Facility for OpenVMS Problem Solving and Messages* manual for the corresponding abbreviated ASCII message text of a particular STV code, an explanation of the abbreviated message text, and a suggested course of action to correct the problem. If you receive any messages that are not listed, see the OpenVMS/DTF server manager. Some user actions (for example, those that require you to use the SNADTFCFG commands) may require the assistance of the OpenVMS/DTF server manager.

11

RSX-11M/M-PLUS File Transfer Operations

This chapter describes how to use the three RSX-11M/M-PLUS file transfer interfaces (DCL, NFT, and FTS) with the OpenVMS/DTF server software to transfer files. You can use the RSX-11M commands supported by DTF at any RSX-11M/M-PLUS node in a DECnet network so long as that node can access the OpenVMS/DTF server. You simply include the OpenVMS/DTF server node as part of the file specification. No DTF software is required at the client node.

11.1 Transferring Files Between RSX-11M/M-PLUS DTF Clients and IBM DTF Clients

You can use the DTF-supported RSX-11M/M-PLUS file transfer commands described in Table 11-1 with an IBM file specification.

Table 11-1 Supported RSX-11M/M-PLUS File Transfer Commands

DCL Command	NFT Primary Switch	FTS Primary Switch
APPEND	/AP	/AP
COPY	(no switch)	(no switch)
CREATE	use TI: as input	¹
DELETE	/DE	/DE
DIRECTORY	/LI, /FU, /AT, or /BR	¹
SUBMIT	/SB or /EX	/SB or /EX
TYPE	(use TI: as output)	¹

¹Operation not available

11.2 APPEND Command

To add the contents of one or more specified input files to the end of a specified output file, use the DCL APPEND command:

```
APPEND input-file-spec[...] output-file-spec
```

the NFT command with the /AP switch:

```
output-file-spec[switches]= input-file-spec[switches]/AP
```

or the FTS command with the /AP switch:

```
output-file-spec[switches]= input-file-spec[switches]/AP
```

See Chapter 2 for the IBM file specification syntax.

Table 11–2 shows the level of support DTF provides for the DCL APPEND command qualifiers. For more detailed information about the APPEND command, refer to the *DECnet-RSX Guide to User Utilities*.

Notes

When DTF for MVS creates an MVS file, it calculates a maximum record length that is greater than the maximum record length of the input file (record length is the next greater multiple of 256 minus 1). Any attempt to append a file with records larger than the calculated maximum record length of the existing file will result in an error.

DTF does not support append operations to files with RECFM=FBS. The request will be rejected.

Table 11–2 DTF Level of Support for APPEND Command Qualifying Switches

DCL APPEND Qualifier	NFT Switch	FTS Switch	For IBM File Specifications
Command Qualifiers			
/LOG	/LO[: <i>filespec</i>]	/LO[: <i>filespec</i>]	supported
/NOWARNINGS	/NM	¹	supported

¹Operation not available

(continued on next page)

Table 11–2 (Cont.) DTF Level of Support for APPEND Command Qualifying Switches

DCL APPEND Qualifier	NFT Switch	FTS Switch	For IBM File Specifications
File Qualifiers			
/DATA_TYPE= ASCII	/AS	/AS	supported
/DATA_TYPE= IMAGE	/IM	/IM	supported
/MACY11	/RAT:MACY11	¹	not supported
/OWNER[= <i>uic</i>]	/FO: <i>owner</i>	¹	not supported
/PRINT	/SP	¹	not supported
/PROTECTION= (<i>code</i>)	/PR: <i>pr-info</i>	¹	not supported
/TRANSFER_MODE= AUTO	/AX	¹	supported
/TRANSFER_MODE= BLOCK	/BK	¹	supported
/TRANSFER_MODE= RECORD	/RC	¹	supported

¹Operation not available

Examples

MVS

```
$APPEND PUFF.TEST SEA"SNADTF"::"DRAGONS.MAGIC(PUFF)/USER:JACKIE"
```

This command appends the latest version of the file PUFF.TEST in the current RSX default directory to the PDS member PUFF of the DRAGONS.MAGIC data set. The server account is SNADTF on the OpenVMS/DTF server node SEA. The MVS user ID is JACKIE. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

MVS

```
FTS>INVENTORY.DAT=FINGER"SNADTF"::"GIBSON.FILTERS/USER:GIBSON/PASS:SMALL,-
FTS>FINGER"SNADTF"::"GIBSON.TIRES/USER:GIBSON/PASS:SMALL"/AS/AP
```

This command appends the files GIBSON.FILTERS and GIBSON.TIRES on the MVS system to the local file INVENTORY.DAT. The OpenVMS/DTF server node is FINGER and the server account is SNADTF. The MVS files are accessed using the MVS user ID GIBSON and the password SMALL. Note the use of the /AS (ASCII) switch and the /AP (append) switch.

VM

```
$APPEND ASHFLD"DTFVM"::"PROFILE EXEC A1/USER:GIBSON" LOGIN.CMD
```

This command appends the file PROFILE EXEC A1 to the RSX file LOGIN.CMD located in default local directory. By default, file access on the VM system is done using mode 1. The server account is DTFVM, the OpenVMS/DTF server node is ASHFLD, and the VM user ID is GIBSON. The VM password is retrieved from the server's proxy database.

VM

```
NFT>EAST"DTFVM"::"WEEK DATA/USER:JONES"=MONDAY.DATA/AP
```

This command appends the file MONDAY.DATA to the VM file WEEK DATA. File access on the VM system is done using mode 1. The server account is DTFVM, the OpenVMS/DTF server node is EAST, and the VM user ID is JONES. The VM password is retrieved from the server's proxy database.

11.3 COPY Command

To transfer sequential files between a DECnet-RSX DTF client and an IBM DTF client use the DCL COPY command:

```
COPY input-file-spec[...] output-file-spec
```

the standard NFT command:

```
output-file-spec[switches]=input-file-spec[switches]
```

or the standard FTS command:

```
output-file-spec[switches]=input-file-spec[switches]
```

These commands allow you to do the following:

- Copy a sequential file from any IBM DTF client to any location in a DECnet network.
- Copy a sequential file from any location in a DECnet network to any IBM DTF client.
- Copy a sequential file from any IBM DTF client to any other IBM client.

By using an IBM file specification as either the input file, the output file, or both, you can initiate a transfer to or from that IBM file. See Chapter 2 for the IBM file specification syntax.

Table 11–3 shows the level of support DTF provides for the COPY command qualifying switches of the various file transfer interfaces. For more detailed information about the COPY commands, refer to the *DECnet-RSX Guide to User Utilities*.

Table 11–3 DTF Level of Support for COPY Command Qualifying Switches

DCL COPY Qualifier	NFT Switch	FTS Switch	For IBM File Specifications
/CONCATENATE	/ME	1	supported
/CONTIGUOUS	/CO	1	not supported
/DATA_TYPE= ASCII	/AS	/AS	supported
/DATA_TYPE= IMAGE	/IM	/IM	supported
/LOG	/LO[: <i>filespec</i>]	/LO[: <i>filespec</i>]	supported
/MACY11	/RAT:MACY11	1	not supported
/NEW_VERSION	/NV	1	not supported
/NO_SPAN	/RAT:NOSPAN	1	ignored
/OWNER[= <i>uic</i>]	/FO: <i>owner</i>	1	not supported
/PRINT	/SP	1	not supported
/PROTECTION= (<i>code</i>)	/PR: <i>pr-info</i>	1	not supported
/REPLACE	/SU	1	supported
/SUBMIT	/SB	1	supported
/TRANSFER_MODE= AUTO	/AX	1	supported
/TRANSFER_MODE= BLOCK	/BK	1	supported
/TRANSFER_MODE= RECORD	/RC	1	supported
/NOWARNINGS	/NM	1	supported

¹Operation not available

Note

When jobs are submitted to IBM batch subsystems, the file records must not exceed 80 bytes. The file will be deleted from the IBM system after the job completes.

Examples

MVS

```
$COPY SNPY"DTF"::"COMICS.SUNDAY(GAR)/USER:SMITH/PASS:CHUCKLES" CAT.TXT
```

This command copies the PDS member GAR of the COMICS.SUNDAY data set located on the IBM system under the MVS user ID SMITH to the CAT.TXT file located in the current RSX directory. The OpenVMS/DTF server node is SNPY and the server account is DTF.

MVS

```
FTS>SAMPLE.MVS=HORSES"SNADTF"::"LIBBY.SAMPLE/USER:LIBBY/PASS:GOBBLED"
```

This command copies the file LIBBY.SAMPLE on the MVS system to the file SAMPLE.MVS file located in the current RSX directory. The MVS user ID is LIBBY and the password is GOBBLED. The OpenVMS/DTF server node is HORSES and the server account is SNADTF.

VM

```
$COPY CHESS"DTFVM"::"PAWN EXEC/USER:ROOK/PASS:BISHOP" KNIGHT.TXT
```

This command copies the file PAWN EXEC to the local file KNIGHT.TXT. The server account is DTFVM, the OpenVMS/DTF server node is CHESS, the VM user ID is ROOK, and the VM password is BISHOP.

VM

```
$COPY MAPLE.TXT BOSTON"DTFVM"::"TREES MACLIB(MAPLE)/USER:HARRIS/PASS:LEAVES"
```

This command copies the MAPLE.TXT file to the TREES maclib on the IBM VM system. The VM user ID is HARRIS and the VM user password is LEAVES. The OpenVMS/DTF server node is BOSTON and the server account is DTFVM.

VM

```
NFT>BOSTON"DTFVM"::"TREES MACLIB(MAPLE)/USER:HARRIS/PASS:LEAVES"  
=MAPLE.TXT/AP
```

(This command must be entered with no continuators within double quotation marks. Because of line-length restrictions, in this manual the command is represented as a wrapped command line.) This command performs the same operation as the previous example except it uses the NFT utility. The command copies the MAPLE.TXT file to the TREES maclib on the IBM VM system. The VM user ID is HARRIS and the VM user password is LEAVES. The OpenVMS/DTF server node is BOSTON and the server account is DTFVM.

11.4 CREATE Command

To create a file on an IBM DTF client, use the DCL CREATE command:

```
CREATE file-spec[,...]
```

or the NFT command:

```
output-file-spec[switches]= TI:[switches]
```

See Chapter 2 for the IBM file specification syntax.

Table 11–4 shows the level of support DTF provides for the DCL CREATE command qualifiers. For more detailed information about the CREATE command, refer to the *DECnet-RSX Guide to User Utilities*.

Notes

VM You cannot create an empty CMS file. The CMS file system does not support the allocation of an empty file. The CREATE operation will succeed but the VM system will delete the empty file when it is closed. Note that you can use the CREATE command with VM DTF clients as long as you enter data before ending the file data with **CTRL/Z**.

The FTS utility does not support a file creation command.

Table 11–4 DTF Level of Support for CREATE Command Qualifying Switches

DCL CREATE Qualifier	NFT Switch	For IBM File Specifications
/CONTIGUOUS	/CO	not supported
/LOG	/LO[: <i>filespec</i>]	supported
/MACY11	/RAT:MACY11	not supported
/NEW_VERSION	/NV	not supported
/NO_SPAN	/RAT:NOSPAN	ignored
/OWNER[= <i>uic</i>]	/FO: <i>owner</i>	not supported
/PRINT	/SP	not supported
/PROTECTION= (<i>code</i>)	/PR: <i>pr-info</i>	not supported
/REPLACE	/SU	supported

(continued on next page)

Table 11–4 (Cont.) DTF Level of Support for CREATE Command Qualifying Switches

DCL CREATE Qualifier	NFT Switch	For IBM File Specifications
/SUBMIT	/SB	supported

Examples

MVS

```
$CREATE BIGVAX"MVSDTF" :: "GARY.DTF(TUESDAY)/USER:GARY/PASS:SECRET"
```

This command creates a new member TUESDAY in the GARY.DTF data set on the IBM system. You can now enter data or press **CTRL/Z** to exit. The OpenVMS/DTF server node is BIGVAX, the server account is MVSDTF, the MVS account is GARY, and the MVS password is SECRET.

VM

```
$CREATE DTFSRV"VMDTF" :: "NEWDATA EXEC/USER:GARY/PASS:GUESSME"
```

This command creates a new file NEWDATA EXEC on the IBM system. You must now enter data (you cannot create empty VM files) and then press **CTRL/Z** to exit. The OpenVMS/DTF server node is DTFSRV, the server account is VMDTF, the VM account is GARY, and the VM password is GUESSME.

11.5 DELETE Command

To delete a file on an IBM DTF client, use the DCL DELETE command:

```
DELETE file-spec[...]
```

the NFT command with the /DE switch:

```
input-file-spec[switches]/DE
```

or the FTS command with the /DE switch:

```
[output-file-spec=]input-file-spec[switches]/DE
```

See Chapter 2 for the IBM file specification syntax.

Table 11–5 shows the level of support DTF provides for the DELETE command qualifying switches of the various file transfer interfaces. For more detailed information about the DELETE commands, refer to the *DECnet-RSX Guide to User Utilities*.

You can use FTS to move a file, that is, copy the file to another system and then delete the original. To move a file, specify both a source and a destination along with the /DE switch (for instance, A=B/DE copies file B to file A and then deletes B).

Notes

If you attempt to delete an MVS file that was created with an expiration date and that expiration date has not been reached, MVS will prompt the MVS console operator for permission to delete the file. The file operation will wait until the console operator replies.

You cannot delete IBM tape-resident files.

You cannot use wildcard characters in an IBM file specification when deleting IBM files.

Table 11–5 DTF Level of Support for DELETE Command Qualifying Switches

DCL DELETE Qualifier	NFT Switch	FTS Switch	For IBM File Specifications
Command Qualifiers			
/LOG	/LO[: <i>filespec</i>]	/LO[: <i>filespec</i>]	supported
/NOWARNINGS	/NM	¹	supported

¹Operation not available

Examples

MVS

```
$ DELETE MYFAIR"DTFMVS"::"NICK.ACCOUNTS(PAYABLE)/USER:NICK"
```

This command deletes the PDS member PAYABLE of the NICK.ACCOUNTS data set from the account under the user ID NICK. The OpenVMS/DTF server node is MYFAIR and the server account is DTFMVS. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

MVS

```
$ DELETE MYFAIR"DTFV200"::"NICK.DTF*/USER:NICK"
```

This command deletes all the data sets beginning with NICK.DTF. Caution is recommended when using wild cards. The OpenVMS/DTF server node is MYFAIR and the server account is DTFV200. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

VM

```
$ DELETE/LOG PARIS"DTFVM"::"ASHFLD * 1/USER:ROBERT"
Files deleted:
PARIS:DTFVM C 1/USER:ROBERT
PARIS:DTFVM M 1/USER:ROBERT
PARIS:DTFVM Z 1/USER:ROBERT
```

This command deletes all files named ASHFLD with mode 1. The server account is DTFVM, the OpenVMS/DTF server node is PARIS, and the user account on the VM system is ROBERT. The password for the ROBERT account is supplied from the OpenVMS/DTF server's proxy database.

VM

```
FTS> BANGOR"MUSIC"::"BLUES EXEC/USER:BROTHERS"/DE
```

This command deletes the file named BLUES EXEC. The server account is MUSIC, the OpenVMS/DTF server node is BANGOR, and the user account on the VM system is BROTHERS. The password for the BROTHERS account is supplied from the OpenVMS/DTF server's proxy database.

VM

```
NFT> CONWAY"DTFVM" :: "COUNTRY EXEC/USER:ROBERT" /DE
```

This command deletes the file named COUNTRY EXEC. The server account is DTFVM, the OpenVMS/DTF server node is CONWAY, and the user account on the VM system is ROBERT. The password for the ROBERT account is supplied from the OpenVMS/DTF server's proxy database.

11.6 DIRECTORY Command

To list one or more files on an IBM client, use the DCL DIRECTORY command:

```
DIRECTORY file-spec[...]
```

or the NFT command with the /LI, /FU, /AT, or /BR switch:

```
output-file[switches]= input-file[switches]/LI[width] | /FU[:width] | /AT | /BR
```

Refer to the sample directory shown at the end of this command description for information about directory contents.

Notes

The DIRECTORY command is not supported for IBM tape volumes.

The FTS utility does not support a file directory command.

See Chapter 2 for the IBM file specification syntax.

Table 11–6 shows the level of support DTF provides for the DIRECTORY command qualifying switches of the various file transfer interfaces. For more detailed information about the DIRECTORY commands, refer to the *DECnet-RSX Guide to User Utilities*.

Table 11–6 DTF Level of Support for DIRECTORY Command Qualifying Switches

DCL DIRECTORY Qualifier	NFT Switch	For IBM File Specifications
/NOWARNINGS	/NM	supported
/OUTPUT= <i>filespec</i>	²	supported
/WIDTH	¹	supported

¹Operation not available
²Specify an output file in the command line

Examples

MVS

```
$DIR/ATTRIBUTES ONE"SNADTF"::"BILL.RSX.*/USER:BILL/PASS:DOLLAR"
```

```
Directory ONE::ETSOV1:
13-JUN-91 15:47:53
```

```
BILL.RSX.TEST/USER:BILL
                                Size: 94./94.      Created: <none_specified>
                                Owner:              Revised: <none_specified>
                                                Expires: <none_specified>
File protection:   System:RWED, Owner:RWED, Group:RWED, World:RWED
File organization: Sequential
File attributes:   Allocation=94.
Record format:    Variable length, maximum 251 bytes
Record attributes: Carriage return
```

```
Total of 94./94. blocks in 1. File
```

This command displays all the fields for all files containing BILL.RSX. in the beginning of their names for the user BILL. The password for the user account is DOLLAR. The OpenVMS/DTF server node is ONE and the server account is SNADTF.

VM

```
$ DIR/BRIEF CANDY"DTFVM"::"FILE1 */USER:ROGERS/PASS:SHADOW"
FILE1 EXEC 1/USER:ROGERS
FILE1 SCRPT 1/USER:ROGERS
FILE1 SUB 1/USER:ROGERS
```

This command displays a brief listing of all files of mode 1 for the user ROGERS. The server account is DTFVM, the OpenVMS/DTF server node is CANDY, and the user password is SHADOW. Note that this command assumes there is an entry in the proxy database that will supply the /MDRPASSWORD, /MDWPASSWORD, and /MDMPASSWORD qualifier information.

VM

```
NFT> MIAMI "DTFVM" : : "RSX */USER:MATTIS/PASS:GLOIN" /BR
FILE1 EXEC 1/USER:MATTIS
FILE1 SCRPT 1/USER:MATTIS
FILE1 SUB 1/USER:MATTIS
```

This command displays a brief listing of all files with the name RSX for the user MATTIS. The server account is DTFVM, the OpenVMS/DTF server node is MIAMI, and the user password is GLOIN. Note that this command assumes there is an entry in the proxy database that will supply the /MDRPASSWORD, /MDWPASSWORD, and /MDMPASSWORD qualifier information.

Additional Information

The DIRECTORY command does not accurately display information about the following fields:

- **Size**—Always contains two numbers in the format *nn/nn*. The first number is always the same as the second number and does not indicate the amount of space currently in use as it should. These numbers are only approximations of the number of 512-byte blocks in a file.
These numbers are always 0 for partitioned data set (PDS) members.
- **Owner**—Always displayed as [0,0].
- **Creation date**—Always displayed as <none_specified>.
- **Expiration date**—Always displayed as <none_specified>.
- **File protection**—Always indicates that system, owner, group, and world do not have access. Access is really determined by the access control information (user name and password) specified in the IBM file specification.

MVS If a file is uncataloged, the IBM file specification qualifier /VOLUME is required with the DIRECTORY command. This qualifier indicates the volumes on which the files reside on the IBM system. If /VOLUME is not specified, then an MVS catalog search will be performed.

11.7 SUBMIT/REMOTE Command

To submit selected files for remote execution on IBM DTF clients, use the DCL SUBMIT/REMOTE command:

```
SUBMIT/REMOTE file-spec[...]
```

the standard NFT command with the /SB or /EX switch:

```
output-file-spec[switches]= input-file-spec[switches]/SB | /EX
```

or the standard FTS command with the /SB or /EX switch:

```
output-file-spec[switches]= input-file-spec[switches]/SB | /EX
```

See Chapter 2 for the IBM file specification syntax.

Table 11–7 shows the level of support DTF provides for the SUBMIT/REMOTE command qualifying switches of the various file transfer interfaces. For more detailed information about the SUBMIT/REMOTE commands, refer to the *DECnet-RSX Guide to User Utilities*.

Note

Files submitted to IBM batch subsystems must not contain any records that exceed 80 bytes.

Table 11–7 DTF Level of Support for SUBMIT/REMOTE Command Qualifying Switches

DCL SUBMIT Qualifier	NFT Switch	FTS Switch	For IBM File Specifications
¹	/AS ²	/AS ²	supported
¹	/IM ²	/IM ²	supported
/LOG	/LO[: <i>filespec</i>]	/LO[: <i>filespec</i>]	supported
/NOWARNINGS	/NM	¹	supported
¹	/NV ²	¹	not supported

¹Operation not available

²Switch available for /SB operations only

(continued on next page)

Table 11–7 (Cont.) DTF Level of Support for SUBMIT/REMOTE Command Qualifying Switches

DCL SUBMIT Qualifier	NFT Switch	FTS Switch	For IBM File Specifications
1	/FO:owner ²	1	not supported
1	/AX ²	1	supported
1	/BK ²	1	supported
1	/RC ²	1	supported

¹Operation not available

²Switch available for /SB operations only

Examples

MVS

```
$SUBMIT/REMOTE WINTER"SNADTF"::"KRINGLE.LIST.JCL/USER:KRINGLE/PASS:SNOW"
```

This command submits the file KRINGLE.LIST.JCL to the batch system on the MVS system. The OpenVMS/DTF server node is WINTER and the server account is SNADTF. The IBM user ID is KRINGLE and the user password is SNOW. The file KRINGLE.LIST.JCL should not contain any records that exceed 80 bytes.

MVS

```
FTS>WINTER"SNADTF"::"KRINGLE.LIST.JCL/USER:KRINGLE/PASS:SNOW"/EX
```

This command is identical to the preceding except it uses the FTS interface.

MVS

```
FTS>WINTER"SNADTF"::"KRINGLE.LIST.JCL/USER:KRINGLE/PASS:SNOW"=JOB.MVS/SB
```

As in the previous two examples this command submits the file KRINGLE.LIST.JCL to the batch system on the MVS system. However, this command first copies the file from the RSX file JOB.MVS. Also this command deletes the MVS file after execution is complete.

VM

```
$SUBMIT/REMOTE -  
_>CASTLE"DTFVM"::"TESTBAT EXEC/USER:GIBSON/BATCH:CMSBATCH/CLASS:A"
```

This command submits the file TESTBAT EXEC to the batch machine CMSBATCH in the class A. The OpenVMS/DTF server node is CASTLE and the server account is DTFVM. The IBM user ID is GIBSON and the user password is taken from the proxy database. The file TESTBAT EXEC should not contain any records that exceed 80 bytes.

VM

```
NFT>HOMER"DTFVM"::"GUIDE EXEC/USER:NEWELL/PASS:SNOWY"/EX
```

This command submits the file GUIDE EXEC to the batch machine CMSBATCH in the class A. The OpenVMS/DTF server node is HOMER and the server account is DTFVM. The IBM user ID is NEWELL and the user password is SNOWY. The file GUIDE EXEC should not contain any records that exceed 80 bytes.

11.8 TYPE Command

To display on a terminal a file residing on an IBM DTF client, use the DCL TYPE command:

```
TYPE file-spec[,...]
```

Table 11–8 shows the level of support DTF provides for the DCL TYPE command qualifiers. For more detailed information about the TYPE command, refer to the *DECnet-RSX Guide to User Utilities*.

Note

To simulate a TYPE command NFT users can copy a file to the TI: device.

Table 11–8 DTF Level of Support for TYPE Command Qualifiers

Qualifier	For IBM File Specifications
/LOG	supported
/NOWARNINGS	supported
/MACY11	not supported

Example

MVS

```
$ TYPE TENNIS"PROS"::"GARY.DTF(TYPE)"
```

This command types the contents of the PDS member TYPE of the GARY.DTF data set located on the IBM system. The OpenVMS/DTF server node is TENNIS and the server account is PROS. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

VM

```
$ TYPE CITIES"DTFVM"::"PROFILE EXEC A1/USER:SAM"
```

This command types the contents of the file PROFILE EXEC A1 on the IBM system. The OpenVMS/DTF server node is CITIES, the server account is DTFVM, and the IBM user is SAM. The VM password, the /MDADDRESS value, and the /MDRPASSWORD value are taken from the OpenVMS/DTF server's proxy database.

11.9 Setting Default File Information

The DECnet-RSX NFT and FTS utilities allow you to set up default file specifications for both input and output files. For DTF file transfers this allows you to specify the server node and password as a partial default file specification. For example,

```
FTS>CHIEF/SNADTF/::/DF:OUT
FTS>/DF
Destination defaults = CHIEF/SNADTF/::
Source defaults =
FTS>GRANGER.DTF2=DTFSAMPLE.TXT
```

In this example the first line sets the output file default to be the OpenVMS /DTF server node name. The second line causes the third and fourth lines to be displayed. The fifth line copies the file DTFSAMPLE.TXT on the RSX system to the file GRANGER.DTF2 on the MVS system using the OpenVMS/DTF server node CHIEF and the server account SNADTF. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

11.10 Error Reporting

Most error messages for the DCL and NFT interfaces are returned as normal DCL or NFT error messages (such as "File already exists"). In some cases the DCL or NFT interface may return a secondary message indicating a DAP error code in the form *nn:nnn*. Usually the DAP error code is returned when the error is DTF-specific. Table 11-9 provides the corresponding abbreviated ASCII message text for all DTF-specific DAP codes. The *Digital SNA Data Transfer Facility for OpenVMS Problem Solving and Messages* manual contains explanations and user actions for these error messages.

The FTS utility creates log files where it records transfer information and any error messages. The log file errors are recorded using the DAP code in the form *nnnnnn*. Table 11-9 provides the corresponding abbreviated ASCII message text for all DTF-specific DAP codes. Before using this table for the FTS DAP codes parse the value as *nn:nnnn*. The *Digital SNA Data Transfer Facility for OpenVMS Problem Solving and Messages* manual contains explanations and user actions for these error messages.

Note

The DECnet-RSX interfaces do not return the DAP secondary status codes. These codes are an important aid in determining the solution to any DTF problems. You can see the error messages generated by these codes by accessing the NETSERVER.LOG files in the SNADTF\$MANAGER account on the OpenVMS/DTF server node. Because these files are not available for world access, you will have to ask the OpenVMS/DTF server manager for assistance.

Table 11-9 DAP Error Codes

Status Code	ASCII Text
4:471	DTFCDDREC
4:472	DTFCFGFIL
4:473	DTFCVT
4:474	DTFDEFFIL
4:475	DTFFALSHT

(continued on next page)

Table 11–9 (Cont.) DAP Error Codes

Status Code	ASCII Text
4:476	DTFNOFIL
4:477	DTFNOREC
4:501	DTFOPRABO
4:502	DTFQUASYN
4:503	DTFQUAVAL
4:506	DTFSESEST
4:507	DTFSESTER
4:510	DTFTRATBL
4:511	DTFUNSTYP
4:512	DTFVERMIS
4:513	DTFACC
4:514	DTFCRE

12

MS-DOS and OS/2 File Transfer Operations

This chapter describes how to use the PATHWORKS for DOS and PATHWORKS for OS/2 NFT file transfer commands with DTF software to transfer files between MS-DOS and OS/2 DTF clients and IBM DTF clients. You can use the NFT commands supported by DTF at any MS-DOS or OS/2 node in a DECnet network so long as that node can access the OpenVMS/DTF server node. You simply include the OpenVMS/DTF server node as part of the file specification. No DTF software is required at the client node.

12.1 Transferring Files Between MS-DOS or OS/2 DTF Clients and IBM DTF Clients

You can use the following DTF-supported PATHWORKS for DOS and PATHWORKS for OS/2 NFT commands with an IBM file specification:

- APPEND - appends a file to a file on an MS-DOS, OS/2, or IBM system
- COPY - transfers a file between an MS-DOS or OS/2 DTF client and an IBM DTF client
- DELETE - deletes a data set on an IBM DTF client
- DIRECTORY - lists data sets stored on an IBM DTF client
- SUBMIT - submits a remote file for execution on an IBM DTF client
- TYPE - displays the contents of a data set on an IBM DTF client

12.2 APPEND Command

To append the contents of one or more input files to the end of a specified output file, use the NFT APPEND command:

```
APPEND [/switch] input-file-spec[...] output-file-spec
```

Table 12–1 shows the level of support DTF provides for the NFT APPEND command switches. For more detailed information about the APPEND command, refer to the *PATHWORKS for DOS DECnet User's Guide* or *PATHWORKS for OS/2 Utilities Guide* manuals.

Note

DTF does not support append operations to files with RECFM=FBS. The request will be rejected.

Table 12–1 DTF Level of Support for APPEND Command Switches

Switch	For IBM File Specifications
/ASCII	supported
/DELETE	not supported
/IMAGE	supported
/NOLOG	supported
/PRINT	not supported
/STREAM	ignored
/SUBMIT	supported
/UNDEFINED	not supported

Examples

MVS

```
C: NFT  
NFT> APPEND FILE1.DAT IBM1 "DTFMVS"::"WHITE.DTF(TALL) "
```

This command appends the local file FILE1.DAT to the member TALL of the PDS WHITE.DTF located on the MVS system. The OpenVMS/DTF server node is IBM1 and the server account is DTFMVS. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

MVS

```
C: NFT
NFT> APPEND RABBIT"DTFMVS"::"JONES.DTF(MAIL)" FILE2.DAT
```

This command appends the member MAIL of the data set JONES.DTF, located on the MVS system, to the local file FILE2.DAT. The OpenVMS/DTF server node is RABBIT and the server account is DTFMVS. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

VM

```
C: NFT
NFT> APPEND STATS1.DAT TENNIS"DTFVM"::"STATS MACLIB(STATS1)/MDWPASS:SECRET"
```

This command appends the local file STATS1.DAT to the maclib member STATS1 in the maclib STATS on the VM system. The OpenVMS/DTF server node is TENNIS and the server account is DTFVM. Note that no IBM user ID or IBM password is given. DTF accesses the files on the IBM system by using information in the OpenVMS/DTF server's proxy database. The minidisk write password is SECRET.

12.3 COPY Command

To transfer ASCII or binary files between an MS-DOS or OS/2 DTF client and an IBM DTF client, use the NFT COPY command:

```
COPY [/switch] input-file-spec[,...] output-file-spec
```

The COPY command allows an MS-DOS or OS/2 user to do the following:

- Copy an ASCII or binary file residing on an IBM DTF client to the local system.
- Copy an ASCII or binary file residing on the local system to any IBM DTF client.

If a remote file has additional attributes besides ASCII or binary, the file will lose those attributes when transferred to an MS-DOS or OS/2 system.

Table 12-2 shows the level of support OpenVMS/DTF provides for the NFT COPY command switches. For more detailed information about the COPY command, refer to the *PATHWORKS for DOS DECnet User's Guide* or *PATHWORKS for OS/2 Utilities Guide* manuals.

Notes

All files submitted for execution on IBM systems using the /SUBMIT switch must contain records that do not exceed 80 bytes.

Files copied and submitted for execution on IBM systems are not deleted after the job is submitted.

Table 12–2 DTF Level of Support for COPY Command Switches

Switch	For IBM File Specifications
/ALLOCATION	supported
/ASCII	supported
/BLOCK	supported
/CC= <i>xxx</i>	supported
/DELETE	not supported
/FIXED	supported
/IMAGE	supported
/LSA	not supported
/MACY11	not supported
/MRS= <i>nnn</i>	supported
/NOCONVERT	not supported
/NOLOG	supported
/NOSPAN	supported
/PRINT	not supported
/STREAM	ignored
/SUBMIT	supported
/UNDEFINED	not supported
/VARIABLE	supported
/VFC= <i>nnn</i>	not supported

Examples

MVS

```
C: NFT
NFT> COPY FILE1.DAT RABBIT"DTFMVS"::"WHITE.DTF(TALL)"
```

This command transfers the local file FILE1.DAT to the member TALL of the PDS called WHITE.DTF located on the IBM system. The OpenVMS /DTF server node is RABBIT and the server account is DTFMVS. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

MVS

```
C: NFT
NFT> COPY RABBIT"DTFMVS"::"WHITE.DTF(SMALL)" FILE2.DAT
```

This command copies the member SMALL of the data set WHITE.DTF, located on the IBM system, to the local file FILE2.DAT. The OpenVMS/DTF server node is RABBIT and the server account is DTFMVS. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

VM

```
C: NFT
NFT> COPY FILE1.DAT PARIS"DTFVM"::"VMSFILE DAT/USER:DTFMAN/PASS:CURIOUS"
```

This command copies the local file FILE1.DAT to the file VMSFILE DAT on the IBM system. The OpenVMS/DTF server node is PARIS and the server account is DTFVM. The VM user ID is DTFMAN and the VM password is CURIOUS.

VM

```
C: NFT
NFT> COPY/SUBMIT EXEC_FILE.DAT OCEAN"DTFVM"::"DTFPROC EXEC
/USER:ROBERTS/PASS:SHADOW/BATCH:CMSBATCH/CLASS:A"
```

This command copies the local file EXEC_FILE.DAT to the file DTFPROC EXEC on the VM system. The OpenVMS/DTF server node is OCEAN and the server account is DTFVM. The VM user ID is ROBERTS and the VM password is SHADOW. After the file is successfully copied it will be submitted to the CMSBATCH batch machine with a class value of A. The file EXEC_FILE.DAT should not contain any records that exceed 80 bytes.

12.4 DELETE Command

To delete a file on an IBM DTF client, use the NFT DELETE command:

```
DELETE [/switch] file-spec
```

Table 12-3 shows the level of support DTF provides for the NFT DELETE command switch. For more detailed information about the DELETE command, refer to the *PATHWORKS for DOS DECnet User's Guide* or *PATHWORKS for OS/2 Utilities Guide* manuals.

Notes

If you attempt to delete an MVS file that was created with an expiration date and that expiration date has not been reached, MVS will prompt the MVS console operator for permission to delete the file. The file operation will wait until the console operator replies. You may terminate the operation by using a `CTRL/Z`.

You cannot delete IBM tape-resident files.

You cannot use wildcard characters in the IBM file specification when deleting files.

Table 12-3 OpenVMS/DTF Level of Support for DELETE Command Switch

Switch	For IBM File Specifications
/NOLOG	supported

Examples

MVS

```
C: NFT
NFT> DELETE PARIS"DTFMVS"::"WHITE.DTF(SPECS)"
```

This command deletes the member SPECS from the PDS called WHITE.DTF located on an IBM system. The OpenVMS/DTF server node is PARIS and the server account is DTFMVS. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

MVS

```
C: NFT
NFT> DELETE BOMBAY"DTFMVS"::"JONES.DTF(TEST)/VOL:DTFV1/USE:JOE/PAS:SECRET"
```

This command deletes the member TEST from the data set JONES.DTF, located on the IBM system, on volume DTFV1. The DTF server is BOMBAY and the server account is DTFMVS. The IBM user ID JOE and IBM password SECRET are used to access the data set. Assuming that the DTF server has a proxy record for IBM user ID JOE and that the operation is successful, the password in the OpenVMS/DTF server's proxy database is set to SECRET.

VM

```
C: NFT
NFT> DELETE NEWYRK"DTFVM"::"FLOWERS IRIS 1/USER:NORMAN/PASS:IDUNNO"
```

This command deletes the file FLOWERS IRIS 1. The OpenVMS/DTF server is NEWYRK and the server account is DTFVM. The VM user ID is NORMAN and the VM password is IDUNNO. Assuming that the OpenVMS/DTF server node has a proxy record for IBM user ID NORMAN and that the operation is successful, the password in the OpenVMS/DTF server's proxy database is set to IDUNNO.

12.5 DIRECTORY Command

To display a file list on an IBM DTF client, use the NFT DIRECTORY command:

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

Table 12-4 shows the level of support DTF provides for the NFT DIRECTORY command switches. For more detailed information about the DIRECTORY command, refer to the *PATHWORKS for DOS DECnet User's Guide* or *PATHWORKS for OS/2 Utilities Guide* manuals.

Table 12–4 DTF Level of Support for DIRECTORY Command Switches

Switch	For IBM File Specifications
/BRIEF	supported
/FULL	supported

Examples

MVS

```
C: NFT
NFT> DIR/BRIEF RABBIT"DTFMVS//"::"WHITE.DTF(*)/USER:WHITE/PASSWORD:TCHDWN"
Directory of RABBIT"DTFMVS":ETSOV2:
WHITE.DTF(MONDAY)/USER:WHITE/PASS:TCHDWN
WHITE.DTF(TUESDAY)/USER:WHITE/PASS:TCHDWN
WHITE.DTF(FRIDAY)/USER:WHITE/PASS:TCHDWN
WHITE.DTF(WEEKEND)/USER:WHITE/PASS:TCHDWN
```

This command displays the names of all the members of the PDS called WHITE.DTF located on an IBM system. The OpenVMS/DTF server node is RABBIT and the server account is DTFMVS. The MVS user ID is WHITE and the user password is TCHDWN.

MVS

```
C: NFT
NFT> DIRECTORY RABBIT"DTFMVS"::"JONES.DTF.T*/VOL:DTFV1"
```

This command displays all data sets beginning with JONES.DTF.T, located on the IBM system, on volume DTFV1. The OpenVMS/DTF server node is RABBIT and the server account is DTFMVS. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

VM

```
C: NFT
NFT> DIRECTORY PARIS"DTFVM"::"HOURS MACLIB(*)/USER:WARNER/PASS:XXXABC"
```

This command displays a list of all files in the maclib HOURS on the VM system. The OpenVMS/DTF server node is PARIS and the server account is DTFVM. The VM user ID is WARNER and the VM password is XXXABC.

Additional Information

The DIRECTORY command does not accurately display information about the following fields:

- **Size**—Always contains two numbers in the format *nn/nn*. The first number is always the same as the second number and does not indicate the amount of space currently in use as it should. These numbers are only approximations of the number of 512-byte blocks in a file.
These numbers are always 0 for partitioned data set (PDS) members.
- **Owner**—Never displayed.
- **Created**—Never displayed.
- **File protection**—Always indicates that system, owner, group, and world have complete access. Access is really determined by the access control information (user name and password) specified in the IBM file specification.

[MVS] If a file is uncataloged, the IBM file specification qualifier /VOLUME is required with the DIRECTORY command. This qualifier indicates the volumes on which the files reside on the IBM system. If /VOLUME is not specified, then an MVS catalog search will be performed.

12.6 SUBMIT Command

To submit a file for remote execution on an IBM DTF client, use the NFT SUBMIT command:

SUBMIT [/switch] *file-spec*

Table 12–5 shows the level of support DTF provides for the NFT SUBMIT command switch. For more detailed information about the SUBMIT command, refer to the *PATHWORKS for DOS DECnet User's Guide* or *PATHWORKS for OS/2 Utilities Guide* manuals.

Table 12–5 DTF Level of Support for SUBMIT Command Switch

Switch	For IBM File Specifications
/[NO]LOG	supported

Examples

MVS

```
C: NFT
NFT> SUBMIT DOCTOR"DTFVM"::"SPOON.TEST.JCL/USER:SPOON/PASS:DTFPASS"
```

This command submits the file SPOON.TEST.JCL, located on the MVS system, to an MVS batch queue. The OpenVMS/DTF server node is DOCTOR and the server account is DTFVM. The MVS user ID is SPOON and the password is DTFPASS. The file SPOON.TEST.JCL should not contain any records that exceed 80 bytes.

VM

```
C: NFT
NFT> SUBMIT MIAMI"HUMOR"::"PRANKS EXEC/USER:JOKER/PASS:LAUGH"
```

This command submits the file PRANKS EXEC, located on the VM system, to the batch machine CMSBATCH with a class of A. The OpenVMS/DTF server node is MIAMI and the server account is HUMOR. The VM user ID is JOKER and the VM password is LAUGH. The file PRANKS EXEC should not contain any records that exceed 80 bytes.

12.7 TYPE Command

To display a file on an IBM client, use the NFT TYPE command:

```
TYPE [switch] file-spec
```

Table 12–6 shows the level of support DTF provides for the NFT TYPE command switch. For more detailed information about the TYPE command, refer to the *PATHWORKS for DOS DECnet User's Guide* or *PATHWORKS for OS/2 Utilities Guide* manuals.

Note

No wildcards are allowed in IBM file names when using the TYPE command.

Table 12–6 DTF Level of Support for TYPE Command Switch

Switch	For IBM File Specifications
/NOLOG	supported

Examples

MVS

```
C: NFT
NFT> TYPE BOSTON"DTFMVS"::"WHITE.DTF(SPECS) "
```

This command displays the member SPECS from the PDS called WHITE.DTF located on an IBM system. The OpenVMS/DTF server node is BOSTON and the server account is DTFMVS. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

MVS

```
C: NFT
NFT> TYPE RABBIT"DTFMVS"::"JONES.DTF(ACCOUNTS)/VOL:DTFV1 "
```

This command displays the member ACCOUNTS from the data set JONES.DTF, located on the IBM system, on volume DTFV1. The OpenVMS /DTF server node is RABBIT and the server account is DTFMVS. Account verification on the MVS system is installation dependent. Contact your system administrator for information on the /USERID and /PASSWORD qualifiers.

VM

```
C: NFT
NFT> TYPE TROLLS"DTFMUSIC"::"FIFTIES TEXT/USER:MUSIC/PASS:CRUISE "
```

This command displays the file FIFTIES TEXT located on the VM system. The OpenVMS/DTF server node is TROLLS and the server account is DTFMUSIC. The VM user ID is MUSIC and the password is CRUISE.

12.8 Error Reporting

This section lists the top-level status messages you may receive when using DTF. You will usually see a primary and secondary message followed by secondary status value (STV). The top-level messages are usually self-explanatory. The STV codes are reported as hexadecimal numbers. Messages appear in a format similar to the following:

primary-text
secondary-text
STV=*hex-number*

where *primary-text* is the primary error message, *secondary-text* is a secondary message offering additional information, and *hex-number* is a hexadecimal number. The secondary message can contain a DAP error code of the form *nn/nnn*.

The primary and secondary messages are standard NFT error messages. Table 12-7 provides the corresponding abbreviated ASCII message text for any DAP error codes received. For the STV codes refer to the *Digital SNA Data Transfer Facility for OpenVMS Problem Solving and Messages* manual for the corresponding abbreviated ASCII message text of a particular code. An explanation of both sets of abbreviated message text and suggested actions to correct any problems can be found in the *Digital SNA Data Transfer Facility for OpenVMS Problem Solving and Messages* manual. If you receive any messages that are not listed, see your OpenVMS/DTF server manager.

Table 12-7 DAP Error Codes

Status Code	ASCII Text
4/471	DTFCDDREC
4/472	DTFCFGFIL
4/473	DTFCVT
4/474	DTFDEFFIL
4/475	DTFFALSHT
4/476	DTFNOFIL
4/477	DTFNOREC
4/501	DTFOPRABO
4/502	DTFQUASYN
4/503	DTFQUAVAL
4/506	DTFSESEST
4/507	DTFSESTER
4/510	DTFTRATBL
4/511	DTFUNSTYP

(continued on next page)

Table 12–7 (Cont.) DAP Error Codes

Status Code	ASCII Text
4/512	DTFVERMIS
4/513	DTFACC
4/514	DTFCRE

A

Using OpenVMS DATATRIEVE with DTF

DATATRIEVE access to IBM file types is a feature that enables you to define DATATRIEVE domains that reference non-VSAM sequential, VSAM sequential, VSAM relative, and VSAM indexed files that reside on an IBM system. Refer to the *VAX DATATRIEVE User's Guide* for more information about VAX DATATRIEVE.

You can use DATATRIEVE/ADT to define record definitions for your IBM files or to use existing record definitions if they are available.

You can use DATATRIEVE commands to:

- Define a DATATRIEVE domain to access an IBM file.
- Prepare a DATATRIEVE domain for access to an IBM file.
- Manipulate records from a domain on an IBM system.
- End a DATATRIEVE session that is accessing an IBM file.

A.1 Defining a DATATRIEVE Domain to Access an IBM File

A.1.1 Before Defining Your DATATRIEVE Domain

Prior to issuing a DATATRIEVE DEFINE command to define your DATATRIEVE domain, you should review the following considerations that might apply to your files:

1. Alphanumeric data stored on IBM systems is typically in EBCDIC format, while data on OpenVMS systems is usually stored in ASCII format. Although DTF provides data translation between these two data formats, you may need to use the VAX Common Data Dictionary and DTF data translation.
2. If you are accessing a VSAM indexed file from DATATRIEVE, you must specify the IBM file specification qualifier /RFA when defining your domain if you are:
 - modifying the file

- forming collections using the FIND command

The /RFA qualifier ensures that your VSAM relative byte addresses (RBAs) will map to your RMS relative file addresses (RFAs). Refer to Chapter 2 for more information.

A.1.2 Using the DATATRIEVE DEFINE Command

In the DATATRIEVE DEFINE command, an IBM file specification identifies the domain that will be used to access an IBM file.

For example:

```
DTR> DEFINE DOMAIN IBM_DOMAIN USING EMPLOYEE_RECDEF ON
DFN> 0 "SNADTF" :: "EMPLOYEE.VSAM.CLUSTER/USERID:DTFUSER/RFA" ;
```

where

0"SNADTF"	is the node and server account information.
EMPLOYEE.VSAM.CLUSTER	identifies the IBM file that will be opened when a DATATRIEVE READY command is issued.
/USERID /RFA	are IBM file specification qualifiers. Review Chapter 2 for applicable qualifiers.

When a DATATRIEVE READY command is issued, a DTF session will be set up to access the file on the IBM system.

A.2 Preparing a DATATRIEVE Domain for Access to an IBM File

Use the DATATRIEVE READY command to access a domain that references an IBM file. Read and write access is supported for all file types.

For example:

```
DTR> READY IBM_DOMAIN WRITE
```

This READY command creates an SNA session to DTF for IBM and opens the file that was identified in the DATATRIEVE DEFINE command. Using the example from the previous section, this command would open the EMPLOYEE.VSAM.CLUSTER file.

If security is installed on the IBM system, then access rights are checked and enforced before the file is opened.

Note

Shared access to domains opened for write access is supported only for VSAM file types.

If you are accessing VSAM indexed files and you receive a status code indicating that the RFA table space size has been exceeded and the requested operation will not be performed, your file has exceeded the RFA table size specified during DTF for IBM installation. To free your table space, close and open the indexed file by issuing a DATATRIEVE FINISH command followed by a READY command.

A.3 Manipulating Records from a Domain on an IBM System

Use the same commands to retrieve, read, write, modify, and delete records from an IBM file as you would use for an RMS file.

A.4 Ending a DATATRIEVE Session That Is Accessing an IBM File

Issue the DATATRIEVE FINISH or EXIT command to end a session to a domain that is accessing an IBM file.

B

RMS File System and Programming Constraints

This chapter describes the Record Management Services (RMS) file system and programming constraints that you should be aware of when using the RMS programming interface to access files on an IBM system. The information provided here should be used along with the information found in the *OpenVMS Record Management Services* manual.

B.1 RMS File System Constraints

The DTF software causes IBM files to appear to the OpenVMS operating system as remote RMS files. You can access the IBM files with direct RMS calls or with certain utilities (such as COPY) that are layered on RMS. The underlying differences in the file systems used by IBM and OpenVMS impose a number of constraints on accessing IBM files.

The following types of files and access methods are not supported by OpenVMS when communicating with an IBM system:

- Sequential:
 - Stream (STM)
 - Stream_CR (STMCR)
 - Stream_LF (STMLF)
 - Undefined (UDF)
 - VFC—When creating a data set on the IBM system, a VFC format can be specified if the record attribute PRINT CARRIAGE_CONTROL is also specified. When this data set is subsequently opened by RMS, it has a record format of VARIABLE and a record attribute of CARRIAGE_RETURN CARRIAGE_CONTROL. If this data set is copied back to an OpenVMS system, the resulting OpenVMS file has similar attributes. In other words, the FAB\$C_VFC FAB\$V_PRN options are transformed to FAB\$C_VAR and FAB\$V_CR.

- Record attributes—No carriage control. A record attribute of FABSV_CR, FABSV_FTN, or FABSV_PRN must be specified when creating a data set on the IBM system with variable-length records. You do not need to specify carriage control for creating data sets with fixed-length records. Data sets without carriage control have a record attribute of CARRIAGE_RETURN CARRIAGE_CONTROL when opened by RMS. If you copy these data sets to an OpenVMS file, the resulting file contains similar attributes. If the files were originally text files with embedded carriage control, then they appear to be double spaced after copying to the IBM system and then back to the OpenVMS system.
- Record access modes—Block I/O is not supported.

B.2 RMS Programming Restrictions

You can use the standard RMS programming interface when writing an application program to access IBM-resident files. You should, however, be aware of certain programming limitations when using the RMS programming interface with DTF.

For more information about the RMS programming interface, refer to the *OpenVMS Systems Routines Record Management Services* manual.

The following sections are an aid for those who intend to write their own programs. Most of the differences between OpenVMS and IBM are in the area of indexed file support. You should review these differences before performing record level access to IBM VSAM file types. There is also a section listing differences for record level access to IBM non-VSAM files.

B.2.1 Record Level Access to VSAM Files

In general, all accesses (such as \$FIND, \$GET, \$PUT, or \$UPDATE) are supported for VSAM sequential, relative, and indexed file types. However, there are differences that should be considered.

B.2.1.1 Unsupported OpenVMS RMS Service Calls

The following OpenVMS RMS service calls, usually supported between two OpenVMS nodes, are not supported between an OpenVMS node and an IBM node.

- \$ENTER
- \$EXTEND
- \$FLUSH
- \$NXTVOL

- \$READ
- \$REMOVE
- \$RENAME
- \$SPACE
- \$TRUNCATE
- \$WAIT
- \$WRITE

B.2.1.2 Unsupported RMS Extended Attribute Blocks

The following RMS extended attribute blocks, usually supported between two OpenVMS nodes, are not supported between an OpenVMS node and an IBM node.

- Protection XAB (zeroes are returned as values for this XAB.)
- Revision date and time XAB
- Terminal XAB

B.2.1.3 Record Locking

As documented in the sections of this appendix that describe the supported RMS service calls, the record locking RAB\$\$_L_ROP bits are not supported with DTF for IBM. However, if a program has a VSAM file open for read and write, then a record is locked when it is read (or located through \$FIND).

The record can be unlocked by:

- Issuing another read
- Issuing a write
- Issuing a free
- Issuing a release

VSAM record locking is different from RMS record locking. When an RMS record is locked, only that particular record is locked. However, when a VSAM record is locked, the entire control interval is locked because VSAM stores data records in control intervals (similar to RMS buckets). Therefore, a record lock condition may exist, not because another program owns the record you are trying to retrieve, but because the other program owns another record in the control interval.

B.2.1.4 Programming Considerations for VSAM Indexed Files

The following should be considered when accessing VSAM indexed files:

- Null keys

VSAM does not support the null key attribute. If an RMS indexed file has been transferred to an IBM system with DTF, the null key attribute will be ignored. This means that if you are accessing the file through a KRF that has a null key defined, records that have the null key string will not be suppressed on a read. Your program will have to do the checking for the null key string.

- Loading a VSAM file

If you are using your own program to load an empty VSAM file, then the file must be closed and opened after records have been inserted in order to gain read access to the file through the use of alternate keys.

- Primary key restriction

Indexed files cannot support duplicate primary keys.

- Key data types

The keys in an indexed file are not typed as in RMS (for example, bin1). Keys are merely hexadecimal strings and records are retrieved in this collating sequence. Note that ASCII collating sequence differs from EBCDIC collating sequence in that EBCDIC sorts numbers after letters.

- Multikey access

IBM multikeyed index files have alternate indexes associated with them. These alternate indexes are initialized after the primary index has been loaded and closed. Once they are initialized, they can be accessed.

If you attempt to access an indexed file with uninitialized indexes, a message is returned to indicate that the file's index tree is in error. Contact the IBM system programmer to help identify the uninitialized index. You can find the file name in the DTF for IBM console log.

B.2.1.5 Programming Considerations for VSAM Relative Files

The following should be considered when accessing VSAM relative files:

- Record length

Records in relative files are fixed-length only.

- Nonexistent record processing

Nonexistent record processing is not supported with relative files. If your program attempts to read by relative record number and the record is not there, then a record not found condition is raised.

B.2.1.6 Programming Considerations for VSAM Sequential Files

The device type should be considered when accessing VSAM sequential files. VSAM sequential files must reside on disk. Note that non-VSAM sequential files can reside on tape or disk.

B.2.2 Record Level Access to IBM Non-VSAM Files

Use these programming considerations when writing your own program for non-VSAM sequential files:

- File sharing
You cannot share a file for output with another program.
- Update support
Update support is disallowed for IBM sequential files with the spanned record attribute.
- Append
You should not use sequential record access when appending records to an existing sequential file. If you do, multiple I/Os will be required if sequential record access is performed when the IBM file has the blocked record format attribute.

B.3 Supported RMS Service Calls

The following is a list of OpenVMS RMS service calls that are supported with DTF:

- \$CLOSE
- \$CONNECT
- \$CREATE
- \$DELETE
- \$DISCONNECT
- \$DISPLAY
- \$ERASE
- \$FIND
- \$FREE

- \$GET
- \$OPEN
- \$PARSE
- \$PUT
- \$RELEASE
- \$REWIND
- \$SEARCH
- \$UPDATE

The rest of this chapter lists each supported RMS service alphabetically and describes which input and output file parameters require special consideration.

Note

Any parameters that are not supported for DECnet operations are not supported by OpenVMS/DTF.

B.3.1 \$CLOSE

Table B-1 describes the special fields of the \$CLOSE RMS service call.

Table B-1 \$CLOSE FAB and XAB Input Fields

Field Name	Option or XAB Type	Restriction	Description
FAB\$F_FOP			File-processing options
	FAB\$V_RWC	Always in effect	Rewind magnetic tape volume on close (applies to magnetic tape only)
	FAB\$V_TEF	Ignored	Truncate at end of file. Never done for MVS files. To deallocate unused space, use the RELEASE attribute or /RELEASE qualifier in your file definition. Always done for VM files.
FAB\$L_XAB			Next XAB address
	XABPRO	Ignored	Modify file protection and ownership
	XABRDT	Ignored	Modify revision date and number

B.3.2 \$CONNECT

Table B-2 describes the special fields of the \$CONNECT RMS service call.

Table B-2 \$CONNECT RAB Input Fields

Field Name	Option	Restriction	Description
RAB\$L_ROP			Record-processing options
	RAB\$V_BIO	Disallowed	Block I/O

B.3.3 \$CREATE

Table B–3 and Table B–4 describe the special fields of the \$CREATE RMS service call.

Table B–3 \$CREATE FAB Input Fields

Field Name	Option	Restriction	Description
FAB\$B_BKS		Ignored	Bucket size
FAB\$W_BLS		Ignored	Block size
FAB\$B_FAC			File access
	FAB\$V_BIO	Disallowed	Block I/O access to file
	FAB\$V_BRO	Ignored	Block or record I/O access to file
	FAB\$V_DEL	Ignored	Delete access to file
	FAB\$V_TRN	Ignored	Truncate access to file
	FAB\$V_UPD	Ignored	Update access to file and explicit file extension
FAB\$L_FOP			File-processing options
	FAB\$V_CBT	Disallowed	Contiguous best try
	FAB\$V_CIF	Disallowed	Create if
	FAB\$V_CTG	Disallowed	Contiguous
	FAB\$V_MXV	Disallowed	Maximize version
	FAB\$V_NFS	Ignored	Non-file-structured
	FAB\$V_OFF	Disallowed	Output file parse
	FAB\$V_POS	Disallowed	Current position
	FAB\$V_RCK	Disallowed	Read check
	FAB\$V_RWC	Always in effect	Rewind on close (applies to magnetic tape only)
	FAB\$V_RWO	Disallowed	Rewind on open
	FAB\$V_SPL	Disallowed	Spool
	FAB\$V_TEF	Ignored	Truncate at end of file. Never done for MVS files. To deallocate unused space, use the RELEASE attribute or /RELEASE qualifier in your file definition. Always done for VM files.

(continued on next page)

Table B-3 (Cont.) \$CREATE FAB Input Fields

Field Name	Option	Restriction	Description
	FABSV_TMD	Disallowed	Temporary marked for delete
	FABSV_TMP	Disallowed	Temporary
	FABSV_WCK	Disallowed	Write check
FABSB_FSZ		Ignored	Fixed control area size
FABSL_MRN		Ignored	Maximum record number
FABSB_RFM		Record formats not listed are disallowed	Record format: indicates type of record format. (FABSC_FIX and FABSC_VAR are fully supported. FABSC_VFC is supported only if the FABSV_PRN option in the FABSB_RAT field has been specified.)
FABSB_RTV		Ignored	Retrieval window size
FABSB_SHR		Disallowed	File sharing. If FABSV_GET is specified, FABSV_SHRPUT is also implied.

Table B-4 \$CREATE XAB Output Fields

Field Name	Option	Restriction	Description
XABALL			Allocation XAB
	XABSB_ALN	Not filled in	Alignment boundary type
	XABSB_AOP	Not filled in	Allocation options
	XABSW_DEQ	Not filled in	Default extension quality
	XABSL_LOC	Not filled in	Location
	XABSW_RFI	Not filled in	Related file identifier
	XABSW_VOL	Not filled in	Related volume number
XABDAT		Disallowed	Date and time XAB
XABKEY			Key definition XAB
	XABSB_DBS	Not filled in	Data bucket size
	XABSW_DFL	Not returned	Data bucket fill size
	XABSB_DTP	Not filled in	Data type of the key

(continued on next page)

Table B-4 (Cont.) \$CREATE XAB Output Fields

Field Name	Option	Restriction	Description
	XAB\$ <i>L</i> _DVB	Not filled in	First data bucket virtual block number
	XAB\$ <i>B</i> _IBS	Not filled in	Index bucket size
	XAB\$ <i>W</i> _IFL	Not filled in	Index bucket file size
	XAB\$ <i>L</i> _KNM	Not filled in	Key name buffer address
	XAB\$ <i>B</i> _LVL	Not filled in	Level of root bucket
	XAB\$ <i>W</i> _MRL	Not filled in	Minimum record length
	XAB\$ <i>B</i> _NSG	Always 1	Number of key segments
	XAB\$ <i>B</i> _NUL	Not supported	Null key value
	XAB\$ <i>W</i> _POS <i>n</i>	Only POS0 returned	Key position
	XAB\$ <i>B</i> _PROLOG	Always 3	Prolog level
	XAB\$ <i>L</i> _RVB	Not returned	Root bucket virtual block number
	XAB\$ <i>B</i> _SIZ <i>n</i>	Only SIZ0 returned	Key size
XABPRO		Disallowed	Protection XAB
XABRDT		Disallowed	Revision date and time XAB
XABSUM			Summary XAB
	XAB\$ <i>W</i> _PVN	Always 3	Prolog version number

B.3.4 \$DELETE

Table B-5 describes the special fields of the \$DELETE RMS service call.

Table B-5 \$DELETE RAB Input Fields

Field Name	Option	Restriction	Description
RAB\$\$_ROP			Record processing options
	RAB\$\$_FDL	Disallowed	Fast delete

B.3.5 \$DISCONNECT

There are no restrictions associated with the \$DISCONNECT input and output fields.

B.3.6 \$DISPLAY

Table B-6 and Table B-7 describe the special fields of the \$DISPLAY RMS service call.

Table B-6 \$DISPLAY NAM Input Fields

Field Name	Option	Restriction	Description
NAM\$B_NOP			NAM block options
	NAM\$V_NOCONCEAL	Ignored	Do not conceal device name

Table B-7 \$DISPLAY FAB and XAB Output Fields

Field Name	Option	Restriction	Description
FAB\$L_ALQ		Not filled in	Allocation quantity (in blocks)
FAB\$L_BKS		Not filled in	Bucket size
FAB\$L_FSZ		Not filled in	Fixed control area size
FAB\$L_GBC		Not filled in	Global buffer count
FAB\$L_MRN		Not filled in	Maximum record number
FAB\$L_RTV		Not filled in	Retrieval window size
XABALL			Allocation XAB
	XAB\$B_ALN	Not filled in	Alignment boundary type
	XAB\$B_AOP	Not filled in	Allocation options
	XAB\$W_DEQ	Not filled in	Default extension quality
	XAB\$L_LOC	Not filled in	Location
	XAB\$W_RFI	Not filled in	Related file identifier
	XAB\$W_VOL	Not filled in	Related volume number
XABDAT		Disallowed	Date and time XAB
XABKEY			Key definition XAB
	XAB\$B_DBS	Not filled in	Data bucket size
	XAB\$W_DFL	Not returned	Data bucket fill size
	XAB\$B_DTP	Not filled in	Data type of the key
	XAB\$L_DVB	Not filled in	First data bucket virtual block number

(continued on next page)

Table B-7 (Cont.) \$DISPLAY FAB and XAB Output Fields

Field Name	Option	Restriction	Description
	XAB\$B_IBS	Not filled in	Index bucket size
	XAB\$W_IFL	Not filled in	Index bucket file size
	XAB\$SL_KNM	Not filled in	Key name buffer address
	XAB\$B_LVL	Not filled in	Level of root bucket
	XAB\$W_MRL	Not filled in	Minimum record length
	XAB\$B_NS	Always 1	Number of key segments
	XAB\$B_NUL	Not supported	Null key value
	XAB\$W_POS n	Only POS0 returned	Key position
	XAB\$B_PROLOG	Always 3	Prolog level
	XAB\$SL_RVB	Not returned	Root bucket virtual block number
	XAB\$B_SIZ n	Only SIZ0 returned	Key size
XABPRO		Disallowed	Protection XAB
XABRDT		Disallowed	Revision date and time XAB
XABSUM			Summary XAB
	XAB\$W_PVN	Always 3	Prolog version number

B.3.7 \$ERASE

There are no restrictions associated with the \$ERASE input and output fields.

B.3.8 \$FIND

Table B-8 describes the special fields of the \$FIND RMS service call.

Table B-8 \$FIND RAB Input Fields

Field Name	Option	Restriction	Description
RAB\$\$_ROP			Record-processing options
	RAB\$\$_NLK	Ignored	No lock
	RAB\$\$_NXX	Ignored	Nonexistent record processing
	RAB\$\$_REA	Ignored	Lock for read
	RAB\$\$_RLK	Ignored	Read of locked record allowed
	RAB\$\$_RRL	Ignored	Read regardless of lock
	RAB\$\$_ULK	Ignored	Manual unlocking
	RAB\$\$_WAT	Disallowed	Wait

B.3.9 \$FREE

There are no restrictions associated with the \$FREE input and output fields.

Note

When using \$FREE with VSAM files, only the current record or control interval is unlocked because only one record or control interval can be locked at a time. \$FREE is ignored for non-VSAM files.

B.3.10 \$GET

Table B-9 describes the special fields of the \$GET RMS service call.

Table B-9 \$GET RAB Input Fields

Field Name	Option	Restriction	Description
RAB\$\$_ROP			Record-processing options
	RAB\$\$_LIM	Disallowed	Limit
	RAB\$\$_NLK	Disallowed	No lock
	RAB\$\$_NXR	Ignored	Nonexistent record processing
	RAB\$\$_REA	Disallowed	Lock for read
	RAB\$\$_RLK	Disallowed	Read of locked record allowed
	RAB\$\$_RRL	Disallowed	Read regardless of lock
	RAB\$\$_ULK	Disallowed	Manual unlocking
	RAB\$\$_WAT	Disallowed	Wait

B.3.11 \$OPEN

Table B–10 and Table B–11 describe the special fields of the \$OPEN RMS service call.

Table B–10 \$OPEN FAB Input Fields

Field Name	Option	Restriction	Description
FAB\$W_DEQ		Ignored	Default file extension quantity: if a nonzero value is present in this field, it applies only to this open of the file.
FAB\$B_FAC			File access
	FAB\$V_BIO	Disallowed	Block I/O access to file
	FAB\$V_BRO	Disallowed	Block or record I/O access to file
	FAB\$V_DEL	Ignored	Delete access to file
	FAB\$V_TRN	Ignored	Truncate access to a file
	FAB\$V_UPD	Ignored	Update access to file and explicit file extension
FAB\$L_FOP			File-processing options
	FAB\$V_OFP	Disallowed	Output file parse
	FAB\$V_POS	Disallowed	Current position
	FAB\$V_RCK	Disallowed	Read check
	FAB\$V_RWC	Always in effect	Rewind on close (applies to magnetic tape only)
	FAB\$V_RWO	Disallowed	Rewind on open
	FAB\$V_TEF	Ignored	Truncate at end of file. Never done for MVS files. To deallocate unused space, use the RELEASE attribute or /RELEASE qualifier in your file definition. Always done for VM files.
	FAB\$V_WCK	Disallowed	Write check
FAB\$B_FSZ		Ignored	Fixed control area size
FAB\$B_RFM		Record formats not listed are disallowed	Record format: indicates type of record format. (FAB\$C_FIX and FAB\$C_VAR are fully supported. FAB\$C_VFC is supported only if the FAB\$V_PRN option in the FAB\$B_RAT field has been specified.)

(continued on next page)

Table B-10 (Cont.) \$OPEN FAB Input Fields

Field Name	Option	Restriction	Description
FAB\$B_SHR		Ignored	File sharing. (For non-VSAM files: files opened for write are not shareable.)

Table B-11 \$OPEN XAB Output Fields

Field Name	Option	Restriction	Description
XABALL			Allocation XAB
	XAB\$B_ALN	Not filled in	Alignment boundary type
	XAB\$B_AOP	Not filled in	Allocation options
	XAB\$W_DEQ	Not filled in	Default extension quality
	XAB\$L_LOC	Not filled in	Location
	XAB\$W_RFI	Not filled in	Related file identifier
	XAB\$W_VOL	Not filled in	Related volume number
XABDAT		Disallowed	Date and time XAB
XABKEY			Key definition XAB
	XAB\$B_DBS	Not filled in	Data bucket size
	XAB\$W_DFL	Not returned	Data bucket fill size
	XAB\$B_DTP	Not filled in	Data type of the key
	XAB\$L_DVB	Not filled in	First data bucket virtual block number
	XAB\$B_IBS	Not filled in	Index bucket size
	XAB\$W_IFL	Not filled in	Index bucket file size
	XAB\$L_KNM	Not filled in	Key name buffer address
	XAB\$B_LVL	Not filled in	Level of root bucket
	XAB\$W_MRL	Not filled in	Minimum record length
	XAB\$B_NSG	Always 1	Number of key segments
	XAB\$B_NUL	Not supported	Null key value
	XAB\$W_POS n	Only POS0 returned	Key position
	XAB\$B_PROLOG	Always 3	Prolog level

(continued on next page)

Table B-11 (Cont.) \$OPEN XAB Output Fields

Field Name	Option	Restriction	Description
	XAB\$L_RVB	Not returned	Root bucket virtual block number
	XAB\$B_SIZ n	Only SIZ0 returned	Key size
XABPRO		Disallowed	Protection XAB
XABRDT		Disallowed	Revision date and time XAB
XABSUM			Summary XAB
	XAB\$W_PVN	Always 3	Prolog version number

B.3.12 \$PARSE

There are no restrictions associated with the \$PARSE input and output fields.

B.3.13 \$PUT

Table B–12 describes the special fields of the \$PUT RMS service call.

Table B–12 \$PUT RAB Input Fields

Field Name	Option or XAB Type	Restriction	Description
RAB\$\$_ROP			Record-processing options
	RAB\$\$_RLK	Disallowed	Read of locked record allowed
	RAB\$\$_TPT	Disallowed	Truncate on put
	RAB\$\$_UIF	Disallowed	Update if
	RAB\$\$_WBH	Disallowed	Write behind

B.3.14 \$RELEASE

Table B–13 describes the special fields of the \$RELEASE RMS service call.

Table B–13 \$RELEASE RAB Input Fields

Field Name	Restriction	Description
RAB\$\$_RFA	Ignored	Record's file address (ignored for non-VSAM files only)

Note

When using \$RELEASE, only the current record or control interval is unlocked because only one record or control interval can be locked at a time.

B.3.15 \$REWIND

There are no restrictions associated with the \$REWIND input and output fields.

B.3.16 \$SEARCH

There are no restrictions associated with the \$SEARCH input and output fields.

B.3.17 \$UPDATE

Table B-14 describes the special fields of the \$UPDATE RMS service call.

Table B-14 \$UPDATE RAB Input and Output Fields

Field Name	Option	Restriction	Description
RAB\$\$_ROP			Record-processing options
	RAB\$\$_WBH	Disallowed	Write behind

C

Maintaining the OpenVMS/DTF File Definition Database

A file definition database resides on each OpenVMS/DTF server node and contains the file definitions that are used to access files on the IBM system. Use the SNA DTF Configuration utility to perform operations on the file definition database.

C.1 Adding Entries to the File Definition Database

To add a new entry to the file definition database, use the ADD FILE_DEFINITION command:

```
ADD FILE_DEFINITION[/qualifier] [node["access-control"]::]file-definition-name input-file-spec
```

or

```
ADD FILE_DEFINITION[/qualifier] [node["access-control"]::]file-definition-name ibm-file-spec-quals
```

Command Parameters

[node["access-control"]::]

The *node* parameter specifies the OpenVMS/DTF server node where the file definition database resides. If you omit the *node* parameter from the ADD FILE_DEFINITION command, the server node defined by the last USE NODE command is selected by default. If you have not issued a USE NODE command, then your current node is selected.

The *access-control* parameter specifies the account name and password of an account on the server node with read and write access to all the databases. The account name is usually SNADTF\$MGR.

Note

Do not use an account in the OpenVMS/DTF server account database for the *access-control* parameter.

file-definition-name

Specifies the user-assigned name of the file definition in the definition database.

input-file-spec

The name of the file used to define file creation parameters. This file can be used for all supported file types, including VSAM files. See Chapter 2 for information on setting up this file.

ibm-file-spec-quals

Specifies IBM file specification qualifiers. These qualifiers are described in Chapter 2.

Command Qualifier

/[NO]LOG

Provides the option of displaying informational messages about the operation.

ADD FILE_DEFINITION Examples

```
DTFCFG> ADD FILE_DEFINITION PERSONNEL_FILE PERSONNEL
```

This example creates the PERSONNEL_FILE file definition record in the file definition database on the default node (the node defined by the last USE NODE command). The DTF file definition attributes are read from the PERSONNEL.SNADTF\$FDL file in the current directory.

```
DTFCFG> ADD FILE_DEFINITION/LOG RAWDATA TAPE_SPECS.TXT
%SNADTF-I-ADDFIL, file definition RAWDATA added
```

This example creates the RAWDATA file definition record in the file definition database on the default node. The /LOG qualifier indicates that a message should be displayed indicating that the file definition has been created. The DTF file definition attributes are read from the TAPE_SPECS.TXT file in the current directory.

```
DTFCFG> ADD FILE_DEFINITION SWNODE"SNADTF$MGR SECRET"::SWD_DATA-
_DTFCFG> SWD_DATA
```

This example creates the SWD_DATA file definition record in the file definition database on node SWNODE. The DTF file definition attributes are read from the SWD_DATA.SNADTF\$FDL file in the current directory.

```
DTFCFG> ADD FILE_DEFINITION NON_VSAM1/NOSUPERSEDE/MDADDRESS=193
```

This example creates a file definition record NON_VSAM1 in the file definition database on the default node (the node defined by the last USE NODE command). The values for the various file creation parameters are specified on the command line. Files created using the NON_VSAM1 file definition will not be superseded and will be placed on the minidisk with an address of 193.

C.2 Displaying File Definition Attributes

To display the attributes of a single file definition or a group of file definitions in the file definition database on a specified node, use the SHOW FILE_DEFINITION command:

```
SHOW FILE_DEFINITION[/qualifiers] [node["access-control"]:::]file-definition-name
```

Command Parameters

[node["access-control"]:::]

The *node* parameter specifies the OpenVMS/DTF server node where the file definition database resides. If you omit the *node* parameter from the SHOW FILE_DEFINITION command, the server node defined by the last USE NODE command is selected by default. If you have not issued a USE NODE command, then your current node is selected.

The *access-control* parameter specifies the account name and password of an account on the server node with read and write access to all the databases. The account name is usually SNADTF\$MGR.

Note

Do not use an account in the OpenVMS/DTF server account database for the *access-control* parameter.

file-definition-name

Specifies the user-assigned name of the file definition in the definition database.

The file definition you specify can include wildcard characters. If you choose to use wildcards, information about all file definitions with matching wildcards is displayed. A percent sign (%) indicates a single wildcard character, and an asterisk (*) indicates a wildcard string. Refer to the description of the OpenVMS RTL routine STR\$MATCH_WILD in the *DCL Dictionary* for more information about wildcards.

Command Qualifiers

/BRIEF

Displays only the file definition names. By default, the entire contents of a file definition are displayed.

/OUTPUT[=*file-specification*]

The name of the file where the output of the SHOW FILE_DEFINITION command is to be placed. If the /OUTPUT qualifier is used without an argument, SYSS\$OUTPUT is used as the default.

SHOW FILE_DEFINITION Examples

```
DTFCFG> SHOW FILE_DEFINITION/BRIEF *DAT*
```

```
!  
!   File definition:  
!  
RAWDATA   SWD_DATA
```

This example displays the file definition names in the database containing the string "DAT." The /BRIEF qualifier causes only the names to be displayed, rather than the entire file definition.

```
DTFCFG> SHOW FILE_DEFINITION/OUTPUT=VSAM_ESDS VSAM_ESDS_FDL
```


This example displays the VSAM_ESDS_FDL file definition. The /OUTPUT qualifier directs the output from this command to the VSAM_ESDS.SNADTF\$FDL file in the current directory.

```
DTFCFG> SHOW FILE_DEFINITION PERSONNEL_FILE

!
!  File definition: PERSONNEL_FILE
!
BASE
RECORD_DEFINITION      PERSONNEL_RECORD
RELEASE                NO
SINGLE                  NO
SPANNED                NO
SUPERSEDE              NO
TRANSLATE              YES
UNIT                   DISK01
USERID                 PERONL
VOLUMES                VOL1
VSAM_SEQUENTIAL        NO
```

This example displays the PERSONNEL_FILE file definition on the user's terminal.

C.3 Modifying a File Definition

The process for modifying file definitions depends on whether you are modifying the BASE section or an ALTERNATE_INDEX section. The following two subsections describe the modify process for each instance.

C.3.1 Modifying the File Definition BASE Section

To modify entries in the file definition's BASE section, use the MODIFY FILE_DEFINITION command:

```
MODIFY FILE_DEFINITION[/qualifier] [node["access-control"]::file-definition-name ibm-file-spec-quals
```

Command Parameters

[*node*["*access-control*"]::*file-definition-name* *ibm-file-spec-quals*

The *node* parameter specifies the OpenVMS/DTF server node where the file definition database resides. If you omit the *node* parameter from the MODIFY FILE_DEFINITION command, the server node defined by the last USE NODE command is selected by default. If you have not issued a USE NODE command, then your current node is selected.

The *access-control* parameter specifies the account name and password of an account on the server node with read and write access to all the databases. The account name is usually SNADTF\$MGR.

Note

Do not use an account in the OpenVMS/DTF server account database for the *access-control* parameter.

file-definition-name

Specifies the user-assigned name of the file definition in the definition database.

ibm-file-spec-quals

Specifies IBM file specification qualifiers. These qualifiers are described in Chapter 2.

Command Qualifier

/[NO]LOG

Provides the option of displaying informational messages about the operation.

MODIFY FILE_DEFINITION Examples

```
DTFCFG> MODIFY FILE_DEFINITION PERSONNEL_FILE PERSONNEL
```

This example modifies the PERSONNEL_FILE file definition record in the file definition database on the default node (the node defined by the last USE NODE command). The DTF file definition attributes are read from the PERSONNEL.SNADTF\$FDL file in the current directory.

```
DTFCFG> MODIFY FILE_DEFINITION NON_VSAM1/TRANSLATE/NOSUPERSEDE-  
_DTFCFG> /MDADDRESS=195
```

This example updates the NON_VSAM1 file definition. Files created using the NON_VSAM1 file definition will now use DMCS/EBCDIC data translation, will not be superseded, and will be on the minidisk with address 195.

```
DTFCFG> MODIFY FILE_DEFINITION MVS_VSAM/USERID=PAYROLL-  
_DTFCFG> /EXPIRATION_DATE=1-JAN-1999/TRANSLATE
```

This example alters the values associated with the EXPIRATION DATE and USERID attributes and allows data translation for the file definition MVS_VSAM. Note that all of these changes occur only in the BASE section of the file definition.

C.3.2 Modifying File Definition ALTERNATE_INDEX Sections

To modify file definition ALTERNATE_INDEX sections, do the following:

1. Use the SHOW FILE_DEFINITION command with the /OUTPUT qualifier. This creates a copy of the file definition you want to modify.
2. Edit this copy of the file and make your changes.
3. Use the REMOVE FILE_DEFINITION command to remove the file definition you want modified.
4. Use the ADD FILE_DEFINITION command to replace the file definition with your modified file.

Example

```
$ RUN SYS$SYSTEM:SNADTF CFG
DTFCFG> SHOW FILE_DEFINITION/OUTPUT=NEW_CHANGES.DAT PERSONNEL_FILE
DTFCFG> EXIT
$ EDIT NEW_CHANGES.DAT
.
.
.
* EXIT
$ RUN SYS$SYSTEM:SNADTF CFG
DTFCFG> REMOVE FILE_DEFINITION PERSONNEL_FILE
DTFCFG> ADD FILE_DEFINITION PERSONNEL_FILE NEW_CHANGES.DAT
DTFCFG> EXIT
```

In this example, `PERSONNEL_FILE` is the file definition you want to modify; `NEW_CHANGES.DAT` is the copy of the file definition that you use to make your actual changes.

C.4 Removing an Entry from the File Definition Database

To remove a file definition from the file definition database on a specified server node, use the `REMOVE FILE_DEFINITION` command:

```
REMOVE FILE_DEFINITION [node["access-control"]::]file-definition-name
```

Command Parameters

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

The *node* parameter specifies the OpenVMS/DTF server node where the file definition database resides. If you omit the *node* parameter from the `REMOVE FILE_DEFINITION` command, the server node defined by the last `USE NODE` command is selected by default. If you have not issued a `USE NODE` command, then your current node is selected.

The *access-control* parameter specifies the account name and password of an account on the server node with read and write access to all the databases. The account name is usually `SNADTF$MGR`.

Note

Do not use an account in the OpenVMS/DTF server account database for the *access-control* parameter.

file-definition-name

Specifies the user-assigned name of the file definition in the definition database.

REMOVE FILE_DEFINITION Examples

```
DTFCFG> REMOVE FILE_DEFINITION ACCOUNTING_FILE
```

This example deletes the record called `ACCOUNTING_FILE` from the file definition database on the current node.

```
DTFCFG> REMOVE FILE_DEFINITION MOOSE"SNADTF$MGR SECRET":POP_FILE
```

This example deletes the record called `POP_FILE` from the file definition database on node `MOOSE`. The access control information within double quotation marks (" ") consists of an account name (`SNADTF$MGR`) and password (`SECRET`) for an account with read and write access to the file definition database.

D

VSAM File Support

This appendix describes how the Digital SNA Data Transfer Facility software processes Variable Sequential Access Method (VSAM) files.

D.1 A VSAM Primer for Non-IBM Users

DTF supports three VSAM file types:

- Sequential files
- Relative record files
- Indexed files

VSAM file types consist of components. The components are:

- Data component (exists for all VSAM file types)
- Index component (exists for indexed files only)

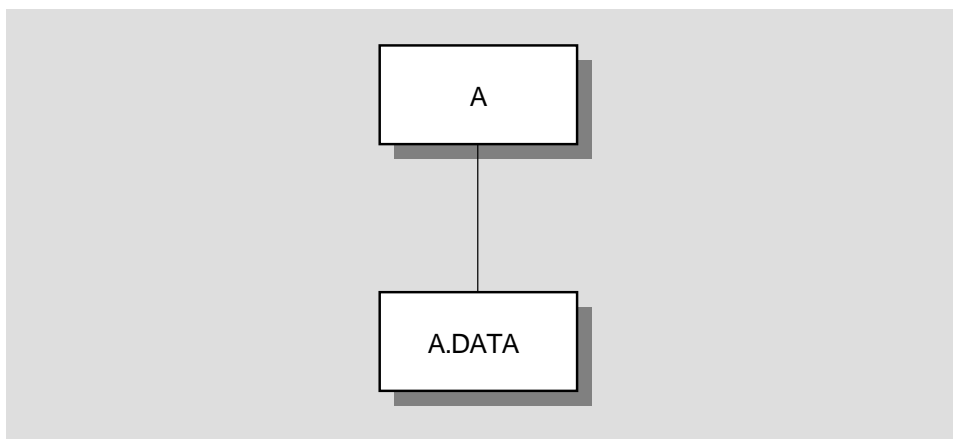
Each component has a name. In addition, the file itself has a name. For information on how to specify the names for VSAM files and their components as file creation parameters, refer to Section 2.4.

D.1.1 Sequential and Relative Files

Figure D-1 illustrates the file structure of VSAM sequential and relative files, where:

- | | |
|--------|---|
| A | is the name of the cluster and is specified on the command line. |
| A.DATA | is the data component name for A. The value of the NAME attribute for the DATA subsection of the BASE section is used here. If this value is not specified, then the default value is used by VSAM. |

Figure D–1 File Structure of VSAM Sequential or Relative Files



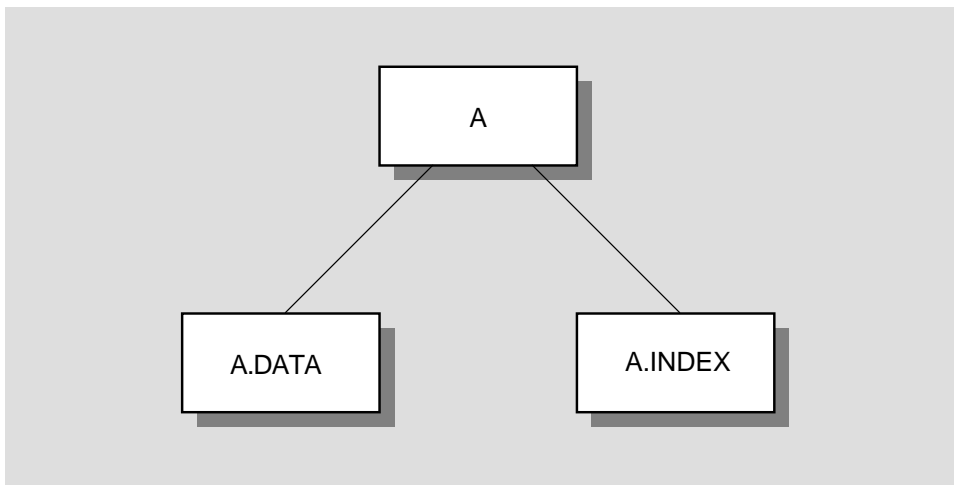
LKG-4261-93R

D.1.2 Indexed Files

Figure D–2 illustrates the file structure of a VSAM indexed file with a single key of reference, where:

- A is the name of the base cluster and is specified on the command line.
- A.DATA is the data component name for A. The value of the NAME attribute for the DATA subsection of the BASE section is used here. If this value is not specified, then the default value is used by VSAM.
- A.INDEX is the index component name for A. The value of the NAME attribute for the INDEX subsection of the BASE section is used here. If this value is not specified, then the default value is used by VSAM.

Figure D-2 File Structure for VSAM Indexed Files



LKG-4262-93R

Like RMS, VSAM can support indexed files with multiple keys of reference. You create multiple keys of reference for an IBM file by creating alternate indexes.

An alternate index is actually an indexed file unto itself (that is, it has data and index components). So, the file structure of an alternate index is identical to the one shown in Figure D-2. However, in order to access a VSAM file through an alternate index, a PATH association must be made between the alternate index and the VSAM indexed file that contains the data records. Unlike RMS, this association is not implicit. Figure D-3 illustrates the file structure of a VSAM multikey indexed file, where:

- A.KRF1 is the name of the first alternate key of reference and is specified by the value of NAME in the ALTERNATE_INDEX section or generated by DTF for IBM.
- A.KRF1.DATA is the data component name for A.KRF1. The value of the NAME attribute for the DATA subsection of the ALTERNATE_INDEX section is used here. If this value is not specified, then the default value is used by VSAM.
- A.KRF1.INDEX is the index component name for A.KRF1. The value of the NAME attribute for the INDEX subsection of the ALTERNATE_INDEX section is used here. If this value is not specified, then the default value is used by VSAM.

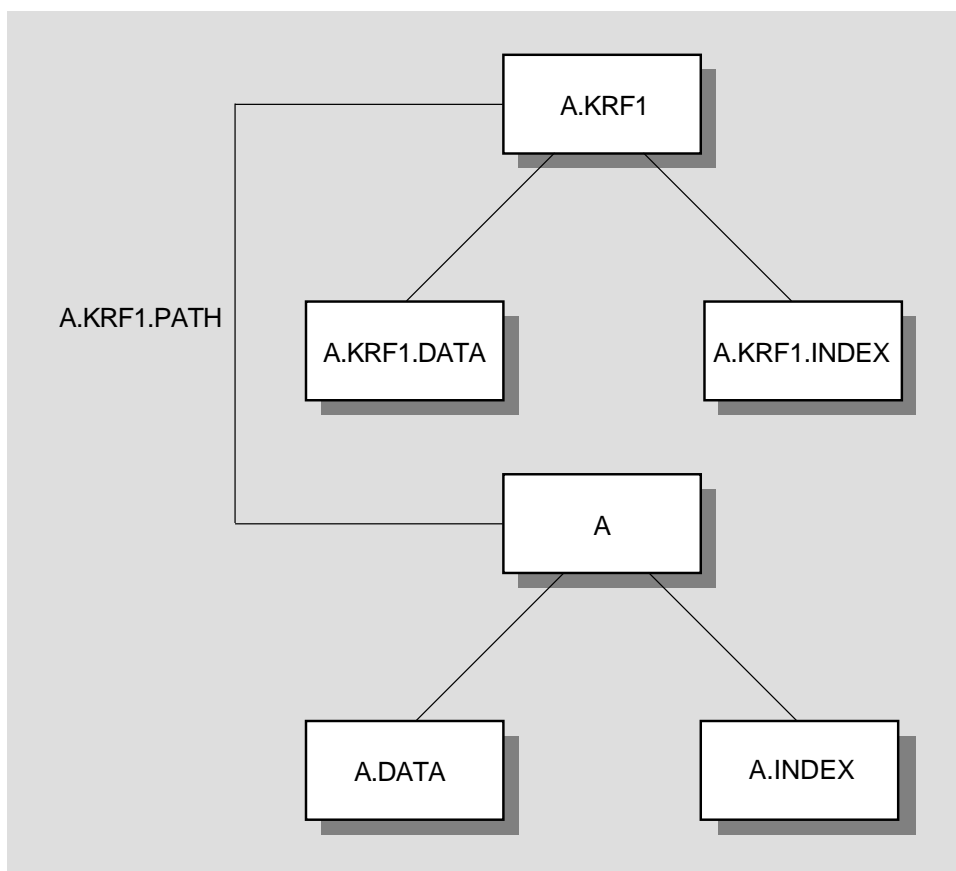
A.KRF1.PATH is the path component name for A.KRF1. The value of the NAME attribute for the PATH section of the ALTERNATE_INDEX section is used here. If this value is not specified, then the default value is generated by DTF for IBM.

A is the name of the base cluster and is specified on the command line.

A.DATA is the data component name for A. The value of the NAME attribute for the DATA subsection of the BASE section is used here. If this value is not specified, then the default value is used by VSAM.

A.INDEX is the index component name for A. The value of the NAME attribute for the INDEX subsection of the BASE section is used here. If this value is not specified, then the default value is used by VSAM.

Figure D-3 File Structure for VSAM Indexed File with Multiple Keys



LKG-4263-93R

D.2 DTF for IBM and VSAM Naming Conventions

Names are assigned to all of the following VSAM entities:

- File
- Data component
- Index component
- Alternate index

- Path

To create any type of VSAM file you need only to specify the name of the cluster on the command line. You can use a file definition to specify any of the names associated with VSAM files. If you do not specify a name for a particular entity, the default values described in the following sections are used.

D.2.1 DTF for IBM Naming Conventions

As noted previously, DTF for IBM will generate the default names for the alternate index and path components. This is because VSAM does not provide defaults for these components. The following conventions are used by DTF for IBM:

- The name of the alternate index cluster component is the name of the base cluster followed by ".*Annnn*" (for example, FILE.NAME.A0001), where *nnnn* corresponds to RMS's key of reference (KRF).
- The name of the path component is the name of the base cluster followed by ".*Pnnnn*" (for example, FILE.NAME.P0001), where *nnnn* corresponds to RMS's KRF.

For each additional key defined for the file, an additional pair of names for the alternate index cluster and path component is generated. The last four digits of those names will correspond to the RMS KRF number. If the six characters appended to the base cluster name cause DTF for IBM to generate a name that is longer than 44 characters, then the base cluster portion of the name is truncated so that the resulting name (truncated base cluster name and appended characters) does not exceed 44 characters.

D.2.2 VSAM Naming Conventions

If names for the data and index components are not specified in a file definition at creation time, then VSAM will generate these names using the following algorithm:

clustername.INDEX for index components

clustername.DATA for data components

where *clustername* is the first portion (qualifier) of the qualified file name used in the TRANSFER/DTF COPY command or the CONVERT command.

D.2.3 Choosing Your Own Name for ALTERNATE_INDEX

An alternate index is IBM's data structure for defining a key of reference. IBM and DTF do not keep track of the KRF sequence. For example, suppose an RMS indexed file has three keys:

<u>Offset of Key in Data Record</u>	<u>Length of Key</u>
0	5 (KRF 0 - primary key)
7	3 (KRF 1 - first alternate)
15	8 (KRF 2 - second alternate)

RMS would store the following data as part of the prolog information: The key field for KRF 0 begins at position 0 and has a length of 5 and KRF 1 begins at offset 7, while KRF 2 begins at offset 15.

Since IBM does not maintain the KRF order, another mechanism is required to preserve this order when an RMS indexed file is copied to the IBM system. DTF for IBM uses the alternate index cluster name to assign the KRF value. When an IBM indexed file is opened by DTF, DTF collects all the alternate index cluster names associated with the file. It then sorts by the full alternate index cluster name and uses the position of the name within the list of names to determine the KRF value.

So, if you wanted to maintain the RMS KRF order when copying the previous file to the DTF for IBM system, you might specify the following alternate index cluster names:

<u>Offset of Key in Data Record</u>	<u>Length of Key</u>	<u>Alternate Index Name</u>
0	5	Since this is the primary key, the name of the file specified on the command line is used. This name is not sorted.
7	3	<i>userid.AIX.KRF1</i>
15	8	<i>userid.AIX.KRF2</i>

Specifying the names in this way allows them to be sorted in the correct KRF order.

D.3 VSAM Cluster Attribute Defaults

DTF uses the standard Access Method Services (IDCAMS) utility to create VSAM clusters. The IDCAMS utility provides defaults for most of the VSAM file attributes. However, DTF does provide defaults for the following VSAM cluster attributes:

- ATTEMPTS
- EXCEPTIONEXIT
- SHAREOPTIONS
- VOLUME

The following sections describe the defaults for these attributes.

ATTEMPTS

DTF provides a value for the IDCAMS ATTEMPTS keyword. The default is not to use this keyword; however, this can be overridden during installation.

EXCEPTIONEXIT

DTF provides a value for the IDCAMS EXCEPTIONEXIT keyword. The default is not to use this keyword; however, this can be overridden during installation.

SHAREOPTIONS

DTF provides a value for the IDCAMS SHAREOPTIONS keyword if the user does not specify a value. The default for this keyword is (1 3); however, this can be overridden during installation.

VOLUME

DTF provides a value for the IDCAMS VOLUME keyword if the user does not specify a volume. A default volume name must be specified during the DTF installation.

E

FTAM File Support

Users in an OSI network can transfer files to and from an IBM system by using OpenVMS FTAM in conjunction with DTF software. IBM users can also transfer files to and from systems in an OSI network by using DTF software and the DAP-FTAM Gateway.

This appendix covers the FTAM product's APPEND, COPY, DELETE, and DIRECTORY facilities and its associated Digital Command Language (DCL) interface.

E.1 Overview of FTAM

The OpenVMS FTAM software is an Open Systems Interconnection (OSI) product that implements the OSI File Transfer Access and Management (FTAM/OSI 8571) standard. When implemented by different computer vendors, the FTAM standard enables the transferring, accessing, and managing of files that reside on other vendors' computer systems. An FTAM system can be defined as any system containing an FTAM implementation that conforms to the FTAM standard and the necessary underlying OSI software. In OpenVMS systems, FTAM software conforms to the FTAM standards.

FTAM software can transfer unstructured files containing either binary or text data and sequential files having a variable record format to and from an FTAM system. (See Table E-3 for information on supported file types.) FTAM using the DAP-FTAM Gateway allows users to transfer files to and from an OSI network.

E.2 FTAM User Facilities

FTAM provides the capability for other OSI, FTAM-compliant systems to use the approved International Organization for Standardization (ISO 8571) protocols. These protocols support file transfer, access, and management through the use of the FTAM user facilities. The facilities supported by DTF software are described in Table E-1. The descriptions include the respective FTAM commands.

Note

OpenVMS users will access the FTAM user facilities to perform FTAM tasks. Users on other operating systems will use an initiator on their systems to access FTAM on OpenVMS through the FTAM responder.

Table E-1 FTAM User Facilities Supported by DTF Software

Facility	Description
Append	Enables you to append files by using the APPEND command. The facility allows the appending of one or more input files to the end of a single output file, within or between FTAM applications.
Copy	Enables you to copy files by using the COPY command. The facility allows the copying of one or more input files to a single output file, within or between FTAM applications.
Delete	Enables you to delete files by using the DELETE command.
Directory	Enables you to display file attributes for one or more files by using the DIRECTORY command.

Note

RENAME is the FTAM user facility that enables you to rename files using the RENAME command. The RENAME facility is the only FTAM user facility that DTF software does not support.

E.2.1 FTAM Components

FTAM consists of the following components:

- FTAM initiators—Initiate associations for user processes.
- FTAM responder—Receives an initiator's request for an association.
- DAP-FTAM Gateway—Allows any DECnet system to communicate with an OSI FTAM system.
- FTAM Configuration—Allows you to configure the FTAM database. See *DECnet/OSI FTAM and VT Use and Management* for more information on this component.

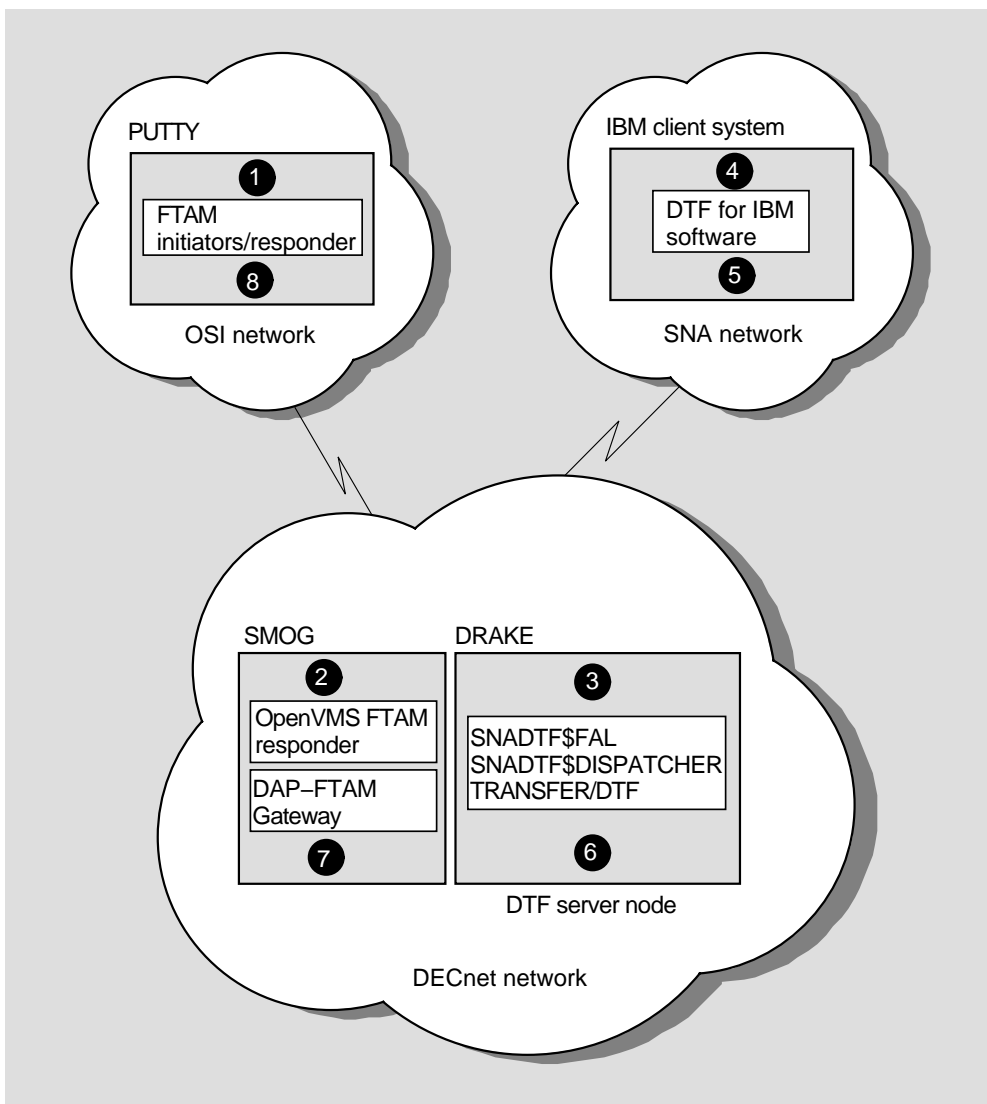
E.2.2 File Transfers Using FTAM Software and a DAP-FTAM Gateway

Figure E-1 shows an OSI-initiated file transfer request and an IBM-initiated file transfer request.

Note

All references to PUTTY in Figure E-1 are based on the premise that PUTTY is an OpenVMS system running an FTAM initiator.

Figure E-1 File Transfers Using FTAM Software and a DAP-FTAM Gateway



LKG-5578-93R

Callouts for Figure E-1

OSI-Initiated File Transfer to the SNA Network

1. An OSI user on PUTTY issues the COPY command, COPY/APP=FTAM /LOG OSISSMOG::DRAKE::"SNADTF"IBM-FILESPEC CAPE.EXT, to request the IBM file be copied to the OSI file CAPE.EXT. This request causes the FTAM initiator to request an association with a remote FTAM application defined by OSI\$FTAM. In this example, the association is to the node SMOG.
2. The remote FTAM application acts as the responder for the association and converts the FTAM protocol messages into RMS system services.
3. The DTF server is invoked by RMS system services. The DTF server connects to DTF for IBM.
4. DTF for IBM completes the request.

SNA-Network-Initiated File Transfer to an OSI Network

5. DTF for IBM receives a request to copy a file. DTF for IBM transfers the request to the DTF server. The following is an example of an MVS user request:

```
DEF INFILE(JONES.DTF)
DEF OUTFILE(SMOG"OSIGTWY"::OSI$PUTTY"SYSTEM -
  OSIUSERPASSWORD"::DTFTEST.DAT)
DEF SERV(DRAKE) NETMGR(DTFR) RECOVERY(N)
TRANS
```

6. The DTF server connects to the DAP-FTAM Gateway.
7. The DAP-FTAM Gateway makes the association with FTAM or the third-party responder.
8. OSI FTAM completes the transfer request.

Note

For more information on FTAM operation, and general OSI information, see the *DECnet/OSI FTAM and VT Use and Management* manual. For information on how to access the FTAM features of a non-OpenVMS system, see the FTAM user documentation for that system.

E.3 FTAM Document Types

The FTAM facilities require that you supply one or more file specifications for each operation. For FTAM, a file specification is a unique string of characters that an FTAM application uses to create or select a file stored on an FTAM system in the same network.

FTAM document types are:

- FTAM-1—Unstructured text files
- FTAM-2—Sequential files
- FTAM-3—Unstructured binary files
- NBS-9—NBS file directories

Note

The FTAM document types are supported by FTAM initiators and responders. The DAP-FTAM Gateway does not support NBS-9.

E.3.1 Restrictions

Table E-2 describes the restrictions on two of the supported FTAM document types.

Table E-2 Restrictions

Document Type	Restrictions
FTAM-1 StrSig VAR	Files can be copied to the IBM system but will lose the FTAM-1 file type definition. The file is translated as an FTAM-2 NOTsignificant file when the file is pulled back. File contents do not alter.
FTAM-3 StrSig FIX	Files can be copied to the IBM system but will lose the attributes when the file is pulled back. The FTAM-3 file type changes to an FTAM-1 StrSig FIX when the file is pulled back. File contents do not alter.

Table E-3 shows the DTF support and mapping among FTAM, OpenVMS, and IBM file attributes.

Table E-3 DTF Support and Mapping Among FTAM, OpenVMS, and IBM (MVS and VM) File Attributes

FTAM		OpenVMS Equivalent			IBM Equivalent		DTF
DocType	StrSig	RAT	RFM	ORG	RECFM	ORG	Support
FTAM-1	FIX	CR	FIX	SEQ	FIX	SEQ	Yes
FTAM-1	VAR	CR	VAR	SEQ	VAR	SEQ	Restrictions
FTAM-1	NOT	CR	STM	SEQ	Not supported		No
FTAM-2	FIX	Not supported			Not supported		No
FTAM-2	VAR	Not supported			Not supported		No
FTAM-2	NOT	CR	VAR	SEQ	VAR	SEQ	Yes
FTAM-3	FIX	None	FIX	SEQ	Not supported		Restrictions
FTAM-3	VAR	Not supported			Not supported		No
FTAM-3	NOT	None	UDF	SEQ	UDF	SEQ	No
NBS-9	—	—	—	—	—	—	No

Legend for Table E-3:

- CR: Carriage Return
- FIX: Fixed
- NOT: NOTsignificant
- ORG: Organization
- PS: Partitioned Sets
- RAT: Record Attributes
- RECFM: and RFM: Record Format
- SEQ: Sequential
- STM: Stream Record Format
- StrSig: String Significance
- UDF: Undefined Format
- VAR: Variable Record

E.4 Examples of OSI-Initiated File Transfer Operations

To activate the FTAM facilities, the FTAM DCL interface requires a single command qualifier (/APPLICATION_PROTOCOL=FTAM), which operates with the APPEND, COPY, DELETE, and DIRECTORY DCL commands. See Chapter 5 for the level of support DTF provides for the DCL APPEND, COPY, DELETE, and DIRECTORY command qualifiers.

Note

All references to systems in the following examples are based on the configuration in Figure E-1.

APPEND Command Example

VM

```
$ APPEND/APP=FTAM FTAM2.NOT OSI$SMOG::DRAKE"DTFVM"::"FTAM2 NOT"
```

The example APPEND command illustrates how you can append (concatenate) a local input file (FTAM2.NOT, an FTAM-1 StrSig FIXED file) to a single remote IBM system output file (FTAM2 NOT). The input file and output file can be any combination of local and remote files. OSI\$SMOG is the association name which points to the FTAM responder. DRAKE is the OpenVMS/DTF server node name. SNADTF is the DTF server account. The file name that is created on the IBM VM system is FTAM2 NOT. (See Section E.3.1 for restrictions on DTF support for FTAM-1 StrSig VAR document type.)

Note

The /APP=FTAM qualifier for the APPEND command is supported by FTAM. However, it is not supported when used against the unsupported DTF file.

COPY Command Examples

VM

```
$ COPY/APP=FTAM OSI$SMOG::DRAKE"DTFVM"::"FTAM2 NOT" FTAM2.VAR
```

The example COPY command illustrates how you can copy a single local file, FTAM2 NOT, to FTAM2.VAR on a remote IBM VM system. OSISSMOG is the association name which points to the FTAM responder. DRAKE is the OpenVMS/DTF server node name. DTFVM is the DTF server account. The file name that is created on the IBM VM system is FTAM2 VAR.

MVS

```
$ COPY/APP=FTAM FTAM1.FIXED OSISSMOG::DRAKE"SNADTF"::"DTFUSER.FTAM1"
```

The example COPY command illustrates how you can copy a single local file, FTAM1.FIXED, to a remote IBM MVS system. OSISSMOG is the association name which points to the TAM responder. DRAKE is the OpenVMS/DTF server node name. SNADTF is the DTF server account. The file name that is created on the IBM MVS system is DTFUSER.FTAM1.

Note

The /APP=FTAM qualifier for the COPY command is supported by FTAM. However, it is not supported when used against the unsupported DTF file.

DELETE Command Example

VM

```
$ DELETE/APP=FTAM OSISSMOG::DRAKE"DTFVM"::"FTAM1 FIXED"
```

The example DELETE command illustrates how you can delete a local file, FTAM1 FIXED. The DELETE/APP=FTAM command allows you to delete any remote file, any local file, or a combination of local and remote files. OSISSMOG is the association name which points to the FTAM responder. DRAKE is the OpenVMS/DTF server node name. DTFVM is the DTF server account.

DIRECTORY Command Example

MVS

```
$ DIRECTORY/APP=FTAM OSISSMOG::DRAKE"SNADTF"::"DTFUSER.DTFPDS(MEM1)/US:DTFUSER"
```

DTF can support directory listings only when the user gives a full file specification. The example shows a PDS member file which is a sequential flat file by itself. When FTAM does a \$OPEN, DTF for IBM can provide the necessary information for directory output.

E.4.1 Examples of File Transfers Using the DAP-FTAM Gateway

You can use the DAP-FTAM Gateway to transfer files to or from an OSI network. IBM users can transfer files to or from an OSI network. DECnet users can also do a third party copy of IBM files to and from an OSI network using the DAP-FTAM Gateway.

Note

All references to systems in the following examples are based on the configuration in Figure E-1.

COPY Command Examples

VM

Initiators to OSI

```
$ COPY/APP=FTAM OSI$SMOG::DRAKE"DTFVM"::"FTAM2 FILE"-  
SMOG"OSIGTWY"::OSI$PUTTY:"OSIUSER OSIPASS"::DTFTEST.DAT
```

The example illustrates an OpenVMS user typing a COPY command to copy an IBM VM file using DTF to an OSI node using the DAP-FTAM Gateway. DRAKE is the OpenVMS/DTF server node. SNADTF is the server account. SMOG is the target node where the DAP-FTAM Gateway software is running. OSI\$PUTTY is the association name.

MVS

Initiated from IBM

```
DEF INFILE(JONES.DTF) [1]  
DEF OUTFILE(SMOG"OSIGTWY"::OSI$PUTTY"SYSTEM OSIUSERPASSWORD"::DTFTEST.DAT)  
[2]  
DEF SERV(DRAKE) NETMGR(DTFR) RECOVERY(N) [3]  
TRANS [4]  
  
DTFM030I REQUEST SUBMITTED TO DTFR [5]  
DTFM037I THIS REQUEST ASSIGNED REQUEST NUMBER 01143  
DTFM055I SUBSYSTEM DTFR      REPORTS COMPLETION OF REQUEST  
DTFM101I SUCCESSFUL TRANSFER - RECORD COUNT IS 163  
====> %SNADTF-S-COPIED, DRAKE"SNADTF"::"ETSOV1:JONES.DTF/CORR:01143"  
====> COPIED TO SMOG"OSIGTWY ">::OSI$PUTTY"OSIUSER password"::SYS$SYSROOT:φ  
====> OSIUSER!DTFTEST.DAT;2 (163 RECORDS)  
  
DTFM048I ENTER CLEAR, DEFINE, END, LIST, RECALL, RESUME, SET, SHOW, OR TRANS  
***
```


The example illustrates an IBM TSO user using the DTF command processor commands to transfer an FTAM 2 NOTsignificant file to the OSI network.

1. The input file in the transfer request is defined as JONES.DTF.
2. The output file is defined as DTFTEST.DAT. The output file is identified with the node name, DRAKE, where the DAP-FTAM Gateway is running and OSISPUTTY, the association name, points to the OSI node address. The OSI user and OSI password are included with the file name.
3. DRAKE has been defined as the server session to which the transfer request is sent for execution. DTFR has been defined as the DTF Network Manager subsystem that should receive the transfer request. Nonrecoverable has been defined as the mode of execution.
4. Executes the request.
5. Job message output for transfer.

Note

The DEFINE subcommand defines input and output file specifications and the environment associated with a file transfer. See Chapter 9 for more information on the DEFINE subcommand.

Note

The COPY and APPEND commands are the only commands supported in a third-party transfer request.

F

DTF Data Translation

This appendix discusses both Digital and IBM data types and the way that data is translated between Digital and IBM data types. It also briefly describes the way to create a record definition (referred to as a data definition in CDD terminology). For more information on Digital data types, refer to the *OpenVMS VAX Architecture Handbook*. For more detailed information about Digital data types and the creation of a data definition, refer to the *Common Data Dictionary Data Definition Language Reference*. For more information about IBM data types, refer to the IBM *Principles of Operations* manual.

F.1 Controlling Data Translation on DECnet-Initiated File Transfers

Translation of data types is controlled by the `/TRANSLATE` and `/RECORD_DEFINITION` qualifiers. The three basic translation modes you should be aware of are as follows:

- DMCS to EBCDIC
- No translation
- Field level translation

By default a standard DMCS-to-EBCDIC translation table is used for data translation. If you specify the `/TRANSLATE` qualifier, however, you can use a user-defined translation table. Translation using the standard translation table or a user-defined translation table can occur on the OpenVMS/DTF server node or on the IBM system. The location of the translation is defined by the OpenVMS/DTF server manager. See Section F.3 for information on translation tables.

If you use the `/NOTRANSLATE` qualifier, the data is transferred without being translated. You might want to use this mode of translation for transferring binary or image data.

The `/RECORD_DEFINITION` qualifier allows you to specify a CDD record definition to control field level translation. If you use the `/RECORD_DEFINITION` qualifier with the `/TRANSLATE` qualifier, you can use a loadable translation table to define the way character data is translated at the field level. Translation using a CDD record definition always occurs on the OpenVMS/DTF server node. See Section F.4 for information on field-level data translation.

F.2 Controlling Data Translation on IBM-Initiated File Transfers

Translation of data types is controlled by following:

- For the command processor: the `TRANS` and `RECDEF` keywords.
- For the panel interface: the `TRANSLATE` field of the Send and Receive panels and the `RECORD DEFINITION` field of the Translate Options panel.
- For the single-line interface: the `TRANSLATE` option.

The `TRANS` and `TRANSLATE` options control whether translation is done. The `RECDEF` and `RECORD DEFINITION` options control which translation tables are used in the translation process.

The three basic translation modes you should be aware of are as follows:

- EBCDIC to DMCS
- No translation
- Field level translation

By default a standard EBCDIC-to-DMCS translation table is used for data translation. If you specify data translation, however, you can use a user-defined translation table. Translation using the standard translation table or a user-defined translation table can occur on the OpenVMS/DTF server node or on the IBM system. The location of the translation is defined by the OpenVMS/DTF server manager. See Section F.3 for information on translation tables.

If you specify no translation, the data is transferred without being translated. You might want to use this mode of translation for transferring binary or image data.

The `RECDEF` and `RECORD DEFINITION` options allow you to specify a CDD record definition to control field level translation. If you use these options with the `translate` option, you can use a loadable translation table to define the way character data is translated at the field level. Translation using a CDD record

definition always occurs on the OpenVMS/DTF server node. See Section F.4 for information on field-level data translation.

F.3 DMCS/EBCDIC Translation Tables

If you need information about modifying translation tables, refer to the *VMS /SNA Management Guide*.

DTF uses translation tables located on the OpenVMS/DTF server node to convert EBCDIC characters received from IBM to DMCS characters used by Digital, and from DMCS to EBCDIC. EBCDIC is a standard convention used by IBM, and DMCS is a superset of the standard American Standard Code for Information Interchange (ASCII) character set.

Note

The file name you specify with the translation qualifier or option must reside on the OpenVMS/DTF server node. By default, DTF looks for the file in SYSS\$LIBRARY with a default extension of .TBL.

F.4 Field-Level Translation

The following sections describe the steps necessary to create and use a field-level translation table.

F.4.1 Creating a New Record Definition

To create a new record definition you must do one of the following:

- Use the CDD DEFINE RECORD command.
- Create a common data definition language (CDDL) file.

After you create the CDDL file, you must add the record definition to the data dictionary by running the CDDL compiler. In either case, the record definition should describe the contents of the file. You can then use the path name of the entry in the data dictionary as the value for the /RECORD_DEFINITION qualifier.

Refer to the *Common Data Dictionary Data Definition Language Reference* for more detailed information about creating new record definitions.

F.4.2 Supported CDD Data Structures

You can use any of the following CDD structures for creating a record definition:

- **ARRAY**—You can use the **ARRAY**, **OCCURS...DEPENDING**, or **OCCURS** field attribute statements to define an array data structure.
- **STRUCTURE**—You can use the **STRUCTURE** field description statement to define a nested data structure.
- **VARIANTS**—You can use the **VARIANTS** field description statement to define a variant data structure. All translations are based on the first variant in the list.
- **VARIANTS OF**—The **VARIANTS OF** data type allows you to define a record that may have one of several formats depending on the value of a field known as the tag variable. DTF supports the following data types for the tag variable: **TEXT**, **SIGNED WORD**, and **SIGNED LONGWORD**.

F.4.3 Translation of Digital CDD and IBM Data Types

Table F-1 lists the Digital CDD data types, indicates whether DTF supports the data type, and, if supported, shows the corresponding IBM data type.

Table F-1 Correspondence of CDD Data Types and IBM Data Types

CDD Data Type	IBM Data Type
DATE	not supported
VIRTUAL	not supported
UNSPECIFIED	not specified
BIT (Bit string)	Bit string
TEXT (DMCS text)	EBCDIC text
VARYING STRING	not supported
POINTER	not supported
[UN]SIGNED BYTE	not supported
[UN]SIGNED WORD (2-byte integer)	2-byte integer (half-word)
[UN]LONGWORD (4-byte integer)	4-byte integer (word)

(continued on next page)

Table F-1 (Cont.) Correspondence of CDD Data Types and IBM Data Types

CDD Data Type	IBM Data Type
[UN]SIGNED QUADWORD	not supported
[UN]SIGNED OCTAWORD	not supported
D_FLOATING	Long floating point
D_FLOATING COMPLEX	not supported
F_FLOATING	Short floating point
F_FLOATING COMPLEX	not supported
G_FLOATING	Long floating point
G_FLOATING COMPLEX	not supported
H_FLOATING	Extended floating point
H_FLOATING COMPLEX	not supported
PACKED DECIMAL	Packed decimal
ZONED NUMERIC	Zoned decimal
[LEFT RIGHT]SEPARATE NUMERIC	not supported
[LEFT RIGHT]OVERPUNCHED NUMERIC	not supported

Most supported data types transferred between Digital and IBM systems are converted without any restrictions or limitations, but you should be aware of the few exceptions that do exist. Table F-2 and Table F-3 list all the data types supported for transfer and any limitations that might occur in data transfers between Digital and IBM.

Note

When a field cannot be mapped exactly between the two different data formats, an error is reported and the file transfer is aborted.

Table F-2 Limitations on Data Transferred from Digital to IBM

Digital Data Types	IBM Data Types	Limitations
2-byte integer (word)	2-byte integer (half-word)	None.
4-byte integer (longword)	4-byte integer (word)	None.
Packed decimal	Packed decimal	None.
DMCS text	EBCDIC text	None.
Zoned numeric	Zoned decimal	An error might be reported when running an IBM application program. IBM accepts a variety of sign conventions for zoned decimal data types, while Digital has only one sign convention for zoned numeric data types. When translating from the Digital format to the IBM format, DTF always uses the most common IBM sign convention for zoned decimal.
Bit string	Bit string	None.
D_FLOATING	Long floating point	If precision is lost in the data mapping, an error is reported and the file transfer is aborted. You can tell DTF to ignore precision loss errors by using the /IGNORE qualifier (see Chapter 2).
F_FLOATING	Short floating point	If precision is lost in the data mapping, an error is reported and the file transfer is aborted. You can tell DTF to ignore precision loss errors by using the /IGNORE qualifier (see Chapter 2).
G_FLOATING	Long floating point	If an underflow or overflow condition occurs in the data mapping, an error is reported and the file transfer is aborted.
H_FLOATING	Extended floating point	If precision is lost or if an underflow or overflow condition occurs in the data mapping, an error is reported and the file transfer is aborted. You can tell DTF to ignore precision loss errors by using the /IGNORE qualifier (see Chapter 2).

Table F-3 Limitations on Data Transferred from IBM to Digital

IBM Data Type	Digital Data Type	Limitations
2-byte integer (half-word)	2-byte integer (word)	None.
4-byte integer (word)	4-byte integer (longword)	None.
Packed decimal	Packed decimal	None.
EBCDIC text	DMCS text	None.
Zoned decimal	Zoned numeric	None.
Bit string	Bit string	None.
Long floating point	D_FLOATING	If an underflow or overflow condition occurs in the data mapping, an error is reported and the file transfer is aborted.
Short floating point	F_FLOATING	If an underflow or overflow condition occurs in the data mapping, an error is reported and the file transfer is aborted.
Long floating point	G_FLOATING	If precision is lost or if an underflow or overflow condition occurs in the data mapping, an error is reported and the file transfer is aborted. You can tell DTF to ignore precision loss errors by using the /IGNORE qualifier (see Chapter 2).
Extended floating point	H_FLOATING	None.

G

Correspondence Between DTF for IBM Keywords and OpenVMS/DTF Qualifiers

Table G-1 shows the correspondence between OpenVMS/DTF IBM file specification qualifiers and the DTF for IBM keywords and fields used in the TSO and CMS command processors and in the ISPF interface. You should be familiar with the relationships between these qualifiers when transferring files between IBM systems and Digital networks.

Table G-1 Correspondence Between OpenVMS/DTF Qualifiers, DTF for IBM Command Processor Keywords, and DTF ISPF Entry Fields

OpenVMS/DTF IBM File Specification Qualifier	Equivalent DTF for IBM Command Processor Keyword	DTFSEND DTFRECV DTFRESUM	Equivalent DTF ISPF Entry Field and DTF ISPF Panels
Server's node and access control information	INRMT, OUTRMT	None ¹	SERVER NODE, SERVER ACCOUNT, and SERVER PASSWORD on the Network Node Options panel
IBM file name	INFILE, OUTFILE	<i>file</i> and <i>node-name</i> parameters	FILENAME on the Input and Output File Options panels
/ALIGNMENT MVS	None ¹	None ¹	None ¹
/ALLOCATION	OUTPRIMARY, OUTSECONDARY	None ¹	PRIMARY/SECONDARY ALLOCATION on Output File Option panels.

¹Specify using the IBM file specification qualifiers described in Chapter 2 in the file name.

(continued on next page)

Table G–1 (Cont.) Correspondence Between OpenVMS/DTF Qualifiers, DTF for IBM Command Processor Keywords, and DTF ISPF Entry Fields

OpenVMS/DTF IBM File Specification Qualifier	Equivalent DTF for IBM Command Processor Keyword	DTFSEND DTFRECV DTFRESUM	Equivalent DTF ISPF Entry Field and DTF ISPF Panels
/BLOCK_SIZE MVS	None ¹	None ¹	None ¹
/CASE	INCASE, OUTCASE	None ¹	MIXED CASE NAME on Input/Output File Option panels.
/CATALOG MVS	None ¹	None ¹	CATALOG on the MVS Output File Options panel
/DEFAULT	None ¹	None ¹	None ¹
/DENSITY	INDEN, OUTDEN	DENSITY	DENSITY on the Input and Output File Options panels
Choice of DCL CREATE or COPY command	DISP	DISP	DISP on the Send and Receive panels
/DIRECTORY_BLOCKS MVS	None ¹	None ¹	None ¹
/FILE_DEFINITION	INFDEF, OUTFDEF	None ¹	FILE DEFINITION on the Input and Output File Options panels
/KEY	None ¹	None ¹	None ¹
/HSMRECALL MVS	None ¹	None ¹	None ¹
/LABEL	INLABEL, OUTLABEL	LABEL	TAPE LABEL on the Input and Output File Options panels
/MRS MVS	None ¹	None ¹	None ¹
/MDADDRESS VM	INMDADDR, OUTMDADDR	VSAMDISK, MDADDR	DEVICE ADDR. in the local file section of the Send and Receive panels
/MDxPASSWORD VM	INMDPASS, OUTMDPASS	MDPASS	PASSWORD in the local file section of the Send and Receive panels

¹Specify using the IBM file specification qualifiers described in Chapter 2 in the file name.

(continued on next page)

Table G–1 (Cont.) Correspondence Between OpenVMS/DTF Qualifiers, DTF for IBM Command Processor Keywords, and DTF ISPF Entry Fields

OpenVMS/DTF IBM File Specification Qualifier	Equivalent DTF for IBM Command Processor Keyword	DTFSEND DTFRECV DTFRESUM	Equivalent DTF ISPF Entry Field and DTF ISPF Panels
Not applicable	NETMGR	NETMGR	SUBSYSTEM on the Operations Selection panel
/NULL	NULL	NULL	NULL RECORDS on Output File Options panels.
/OWNERID VM	INMDOWNER, OUTMDOWNER	MOWNER	OWNER in the local file section of the Send and Receive panels
/PASSWORD	INPASS, OUTPASS	PASS	PASSWORD on the Send, Receive, and Resume panels
/POST	POST	None ¹	FILENAME on the Post Processing Options panel.
/RECORD_DEFINITION	RECDEF	None ¹	RECORD DEFINITION on the Data Translation Options panel
/[NO]RELEASE MVS	None ¹	None ¹	None ¹
/RETENTION_PERIOD	None ¹	None ¹	None ¹
/RFA	None ¹	None ¹	None ¹
/SECURITY_DATA	None ¹	None ¹	None ¹
/SEQUENCE_NUMBER	INNUM, OUTNUM	NUMBER	FILE NUMBER on the MVS Input and Output File Options panels
No equivalent	SERVER	SERVER	SERVER NODE on the Operations Selection panel
/[NO]SINGLE	None ¹	None ¹	None ¹
/SMSxCLASS MVS	None ¹	None ¹	None ¹

¹Specify using the IBM file specification qualifiers described in Chapter 2 in the file name.

(continued on next page)

Table G–1 (Cont.) Correspondence Between OpenVMS/DTF Qualifiers, DTF for IBM Command Processor Keywords, and DTF ISPF Entry Fields

OpenVMS/DTF IBM File Specification Qualifier	Equivalent DTF for IBM Command Processor Keyword	DTFSEND DTFRECV DTFRESUM	Equivalent DTF ISPF Entry Field and DTF ISPF Panels
/[NO]SPANNED	None ¹	None ¹	None ¹
/[NO]TRANSLATE	TRANS	TRANSLATE	TRANSLATE on the Send and Receive panels
/UNIT	INUNIT, OUTUNIT	UNIT	UNIT and TAPE UNIT on the Input and Output File Options panels
/USERID	INUSER, OUTUSER	<i>user-name</i> parameter	USERID on the Send and Receive panels
/VOLUME	INVOL, OUTVOL	VOLUME	VOLUME SERIAL on the MVS Input and Output File Options panels
/VSAM	None ¹	VSAMDISK	VSAM on the Input and Output File Options panels
TRANSFER/DTF /CHECKPOINT	CHKPT	CHECKPOINT	CHECKPOINT on the Send or Receive panels
TRANSFER/DTF /RECOVER	RECOVERY	RECOVERY	RECOVERABLE on the Send or Receive panels

¹Specify using the IBM file specification qualifiers described in Chapter 2 in the file name.

Index

A

Access control information

- for accessing DECnet files, 2-4
- for accessing IBM files, 2-1
- for the server account, 2-1

ADD FILE_DEFINITION (SNADTFCFG)
command, 2-34, C-1

/ALIGNMENT:*option*, 2-15

/ALLOCATION:*number-of-512-byte-blocks*,
2-15

APPEND command

- OpenVMS DCL, 5-2
- PATHWORKS for DOS NFT, 12-2
- PATHWORKS for OS/2 NFT, 12-2
- RSX-11M/M-PLUS DCL, 11-2

Appending files

See Files, adding contents to

Appending files to files
from OpenVMS DCL, 5-2

B

Backing up files

- See* Files, backing up
- from OpenVMS DCL, 5-5

BACKUP command (OpenVMS DCL), 5-5

Batch processing

- /BATCHID qualifier, 2-16
- /CLASS qualifier, 2-18
- command processor running in batch
mode, 9-39, 9-40
- introduction, 1-7
- submitting jobs

Batch processing

submitting jobs (cont'd)

- from MS-DOS clients, 12-9
- from OS/2 clients, 12-9
- with DECnet-ULTRIX commands,
10-3
- with RSX-11M DCL, 11-14
- with DECnet-RSX FTS, 11-14
- with DECnet-RSX NFT, 11-14
- with OpenVMS DCL, 5-31
- with the command processor, 9-23
- with the panel interface, 7-42
- with TRANSFER/DTF, 6-9

/BATCHID:*name*, 2-16

/BLOCK_SIZE:*blocksize*, 2-16

BUFFERSPACE file definition attribute,
2-42

C

CANCEL JOB (TRANSFER/DTF) command,
6-20

/CASE, 2-16

/CATALOG:*option*, 2-17

CDDL

See Common data definition language

Checkpoint support

See Recoverable copy operations

CHKPT keyword, 9-18

/CLASS:*class-name*, 2-18

CLEAR subcommand, 9-4
example, 9-4

SCLIENT, 9-46
 CLIENT keyword, 9-33
 SCLIENTPASS, 9-47
 CLIENTPASS keyword, 9-32
 \$CLIENTUSER, 9-46
 CLIENTUSER keyword, 9-32
 CLIST variables
 SCLIENT, 9-46
 \$CLIENTPASS, 9-47
 \$CLIENTUSER, 9-46
 \$RETNUM, 9-46
 CLOSE command (OpenVMS DCL), 5-8
 \$CLOSE RMS service call, B-7
 Closing files
 See Files, closing
 from OpenVMS DCL, 5-8
 Command processor
 batch mode execution, 9-39, 9-40
 batch mode restrictions, 9-44
 CLEAR subcommand, 9-4
 CLIST variables set, 9-46
 DEFINE subcommand, 9-5
 END subcommand, 9-22
 entering a DECnet file specification, 9-3
 entering an IBM file specification, 9-3
 HELP subcommand, 9-26
 introduction, 1-5
 invoking, 9-1
 keyvalue syntax, 9-3
 keyword syntax, 9-2
 list of subcommands, 9-2
 LIST subcommand, 9-28
 overriding the server, 9-39
 overriding the server account, 9-39
 RECALL subcommand, 9-29
 RESUME subcommand, 9-30
 return codes, 9-45
 REXX exec variables set, 9-46
 SET subcommand, 9-36
 SHOW subcommand, 9-38
 TRANSFER subcommand, 9-23
 Common data definition language (CDDL)
 creating CDDL file, F-3
 /RECORD_DEFINITION qualifier, 2-27
 supported data structures, F-4

Comparing files
 See Files, comparing contents
 from OpenVMS DCL, 5-18
 COMPONENT_ALLOCATION file definition
 attribute, 2-42
 \$CONNECT RMS service call, B-7
 CONVERT command (OpenVMS DCL), 5-9
 Converting files
 See Files, copying indexed and relative
 from OpenVMS DCL, 5-9
 COPY (TRANSFER/DTF) command, 6-3
 COPY command
 OpenVMS DCL, 5-11
 PATHWORKS for DOS NFT, 12-3
 PATHWORKS for OS/2 NFT, 12-3
 RSX-11M/M-PLUS DCL, 11-4
 Copying files
 See Files, copying indexed and relative
 See Files, copying sequential
 See Files, copying with checkpoint and
 recovery
 from OpenVMS DCL, 5-11, 5-24
 CREATE command
 OpenVMS DCL, 5-14
 RSX-11M/M-PLUS DCL, 11-7
 \$CREATE RMS service call, B-8
 Creating a data definition, F-3
 Creating files
 See Files, creating
 from OpenVMS DCL, 5-14

D

Data definition
 creating, F-3
 Data security
 IBM passwords, 2-27
 introduction, 4-1
 /SECURITY_DATA qualifier, 2-29
 specifying IBM user IDs, 2-32
 Data translation, F-1
 See also /TRANSLATE and TRANSLATE
 keyword
 See also Translation modes

Data translation (cont'd)

- errors during, F-5
- introduction, 1-6
- /RECORD_DEFINITION qualifier, 2-27
- /TRANSLATE qualifier, 2-31
- using the panel interface, 7-39

Data types

- translation of Digital and IBM, F-4

DATATRIEVE

- and DTF, A-1

Date and Time Restrictions, 5-24

dcat DECnet-ULTRIX command, 10-1

DCL commands

- OpenVMS
 - APPEND, 5-2
 - BACKUP, 5-5
 - CLOSE, 5-8
 - CONVERT, 5-9
 - COPY, 5-11
 - CREATE, 5-14
 - DELETE, 5-16
 - DIFFERENCES, 5-18
 - DIRECTORY, 5-20
 - EXCHANGE/NETWORK, 5-24
 - OPEN, 5-27
 - READ, 5-28
 - SEARCH, 5-29
 - SUBMIT, 5-31
 - supported commands, 1-8
 - TYPE, 5-33
 - WRITE, 5-35
- RSX-11M/M-PLUS
 - APPEND, 11-2
 - COPY, 11-4
 - CREATE, 11-7
 - DELETE, 11-9
 - DIRECTORY, 11-11
 - SUBMIT, 11-14
 - supported commands, 1-10
 - TYPE, 11-16

dcp DECnet-ULTRIX command, 10-3

dcp_i DECnet-ULTRIX command, 10-5

DECnet file specification format, 2-4

DECnet-ULTRIX

- backup and restore operations, 10-5
- dcat** command, 10-1
- dcp** command, 10-3
- dcp_i** command, 10-5
- dls** command, 10-6
- drm** command, 10-8
- supported file commands, 1-10

DECnet-Internet Gateway

- specifying node name and account for, 9-11, 9-17

/DEFAULT, 2-18

DEFINE subcommand, 9-5

- CHKPT keyword, 9-18
- DISP keyword, 9-13
- example, 9-22
- INCASE keyword, 9-7
- INDEN keyword, 9-8
- INFDEF keyword, 9-9
- INFILE keyword, 9-7
- INLABEL keyword, 9-9
- INMDADDR keyword, 9-9
- INMDOWNER keyword, 9-10
- INMDPASS keyword, 9-10
- INNUM keyword, 9-10
- INPASS keyword, 9-10
- input file keywords, 9-7
- INRMT keyword, 9-11
- INUNIT keyword, 9-11
- INUSER keyword, 9-11
- INVOL keyword, 9-12
- NETMGR keyword, 9-20, 9-37
- OUTCASE keyword, 9-12
- OUTDEN keyword, 9-14
- OUTFDEF keyword, 9-14
- OUTFILE keyword, 9-12
- OUTLABEL keyword, 9-14
- OUTMDADDR keyword, 9-15
- OUTMDOWNER keyword, 9-15
- OUTMDPASS keyword, 9-15
- OUTNUM keyword, 9-16
- OUTPASS keyword, 9-16
- OUTPRIMARY keyword, 9-16
- output file keywords, 9-12
- OUTRMT keyword, 9-17

DEFINE subcommand (cont'd)

- OUTSECONDARY keyword, 9-17
- OUTUNIT keyword, 9-17
- OUTUSER keyword, 9-17
- OUTVOL keyword, 9-18
- RECDEF keyword, 9-19
- RECOVERY keyword, 9-19, 9-37
- SERVER keyword, 9-20, 9-37
- transfer parameter keywords, 9-18
- TRANSLATE keyword, 9-21

Defining file names

- DECnet file specification syntax, 2-4
- IBM file specification syntax, 2-1
- MS-DOS file specification syntax, 2-7
- OpenVMS file specification syntax, 2-5
- OS/2 file specification syntax, 2-7
- RSX-11M/M-PLUS file specification syntax, 2-6
- ULTRIX and Digital UNIX file specification syntax, 2-8

DELETE command

- OpenVMS DCL, 5-16
- PATHWORKS for DOS NFT, 12-5
- PATHWORKS for OS/2 NFT, 12-5
- RSX-11M/M-PLUS DCL, 11-9

\$DELETE RMS service call, B-11

Deleting files

- See Files, deleting
- from OpenVMS DCL, 5-16

DENSITY entry field, 7-26, 7-29, 7-32, 7-37

/DENSITY: *density*, 2-18

DIFFERENCES command (OpenVMS DCL), 5-18

Digital UNIX file specification format, 2-8

DIRECTORY command

- OpenVMS DCL, 5-20
- PATHWORKS for DOS NFT, 12-7
- PATHWORKS for OS/2 NFT, 12-7
- RSX-11M/M-PLUS DCL, 11-11

Directory listings

- from OpenVMS DCL, 5-20

Directory listings files

- See Files, listing a directory

Directory operations

- introduction, 1-7

/DIRECTORY_BLOCKS: *n*, 2-19

\$DISCONNECT RMS service call, B-11

DISP keyword, 9-13

\$DISPLAY RMS service call, B-12

Displaying file definitions from the file definition database, 2-35

Displaying files on a terminal

- See Files, displaying on a terminal
- from OpenVMS DCL, 5-33

dls DECnet-ULTRIX command, 10-6

drm DECnet-ULTRIX command, 10-8

DTF command, 9-2

DTFREC V file transfer command, 8-8

DTFRESUM file transfer command, 8-13

DTFSEND file transfer command, 8-3

E

END subcommand, 9-22

- example, 9-22

\$ERASE RMS service call, B-14

Error reporting

- displaying error messages, 6-21
- MS-DOS clients, 12-11
- OS/2 clients, 12-11
- ULTRIX clients, 10-9

Errors during data translation, F-5

EXCHANGE/NETWORK command (OpenVMS DCL), 5-24

F

File definition

- adding to the file definition database, 2-34, C-1

attributes

- BUFFERSPACE, 2-42
- COMPONENT_ALLOCATION, 2-42
- NAME, 2-43
- REPLICATE, 2-43
- REUSE, 2-43
- SHARE_OPTIONS, 2-44

- creating, 2-33

File definition (cont'd)

- displaying, 2-35, C-3
- examples, 2-37
- /FILE_DEFINITION qualifier, 2-19
- modifying, 2-35, C-7
- removing from the file definition database, C-8
- sections
 - ALTERNATE_INDEX, 2-39
 - BASE, 2-39
- subsections
 - DATA, 2-39
 - INDEX, 2-39
 - PATH, 2-39
- syntax, 2-36
- using IBM file qualifiers in, 2-40

File definition database

- adding file definitions, 2-34, C-1
- displaying file definitions from, 2-35, C-3
- modifying a file definition, 2-35, C-7
- removing file definitions, C-8

FILE_DEFINITION entry field, 7-26, 7-29, 7-37

FILE_NUMBER entry field, 7-25, 7-29, 7-32, 7-37

File specification syntax

- DECnet file specification, 2-4
- IBM file specification, 2-1
- MS-DOS file specification, 2-7
- OpenVMS file specification, 2-5
- OS/2 file specification, 2-7
- RSX-11M/M-PLUS file specification, 2-6
- ULTRIX and Digital UNIX file specification, 2-8

File types supported, 1-6

- as input files, 3-1
- as output files, 3-4
- Digital, 1-6
- IBM, 1-6

Files

See also VSAM files

- adding contents to, 11-2, 12-2
 - from OpenVMS DCL, 5-2
- backing up
 - from OpenVMS DCL, 5-5

Files (cont'd)

- closing
 - from OpenVMS DCL, 5-8
 - comparing contents
 - from OpenVMS DCL, 5-18
 - copying indexed and relative
 - from OpenVMS DCL, 5-9
 - copying sequential, 11-4
 - copying sequential files
 - from OpenVMS DCL, 5-11
 - copying with checkpoint and recovery, 6-3
 - creating
 - from OpenVMS DCL, 5-14
 - creating on an IBM system, 11-7
 - deleting, 11-9
 - from OpenVMS DCL, 5-16
 - displaying on a terminal, 11-16
 - from OpenVMS DCL, 5-33
 - listing a directory, 11-11
 - from OpenVMS DCL, 5-20
 - null record handling in VM, 2-26
 - opening
 - from OpenVMS DCL, 5-27
 - reading records from
 - from OpenVMS DCL, 5-28
 - searching for, 11-11
 - from OpenVMS DCL, 5-20
 - searching for strings in
 - from OpenVMS DCL, 5-29
 - submitting, 11-14
 - from OpenVMS DCL, 5-31
 - writing records to
 - from OpenVMS DCL, 5-35
- /FILE_DEFINITION:*file-definition*, 2-19
- \$FIND RMS service call, B-14
- \$FREE RMS service call, B-14
- FTAM
 - file types, E-6
 - overview, E-1
 - restrictions, E-6
- FTAM
 - append facility, E-2
 - components, E-3
 - copy facility, E-2

FTAM (cont'd)

- delete facility, E-2
- directory facility, E-2

FTS commands

DECnet-RSX

- /AP, 11-2
- copy, 11-4
- /DE, 11-9
- /EX, 11-14
- /SB, 11-14
- supported commands, 1-10

G

GDG

See Generation data group (GDG)

Generation data group (GDG)

- specifying member, 2-1

\$GET RMS service call, B-15

H

HELP subcommand, 9-26

/HSMRECALL, 2-19

I

IBM file specification format, 2-1

IBM file specification qualifiers, 2-9

- /ALIGNMENT, 2-15
- /ALLOCATION, 2-15
- /BATCHID, 2-16
- /BLOCK_SIZE:*blocksize*, 2-16
- /CASE, 2-16
- /CATALOG:*option*, 2-17
- /CLASS:*class-name*, 2-18
- /DEFAULT, 2-18
- defaults, 2-9
- /DENSITY:*density*, 2-18
- /DIRECTORY_BLOCKS:*n*, 2-19
- /FILE_DEFINITION:*file-definition*, 2-19
- /HSMRECALL, 2-19
- /IGNORE:*condition*, 2-20
- /KEY:*position,length[,position,length,...]*, 2-20
- /LABEL:*label*, 2-21

IBM file specification qualifiers (cont'd)

- /MDADDRESS:*disk-address*, 2-21
- /MDPASSWORD:*password*, 2-21
- /MDRPASSWORD:*password*, 2-21
- /MDWPASSWORD:*password*, 2-21
- /NULL:*option*, 2-26
- /OWNERID:*owner*, 2-26
- /PASSWORD:*password*, 2-27
- /RECORD_DEFINITION:*record-definition*, 2-27
- /RELEASE, 2-27
- restrictions, 2-10
- /RETENTION_PERIOD:*n*, 2-27
- /RFA, 2-28
- /SECONDARY_ALLOCATION, 2-28
- /SECURITY_DATA:*data*, 2-29
- /SEQUENCE_NUMBER:*n*, 2-29
- /SINGLE, 2-29
- /SMSDCCLASS:*data-class*, 2-30
- /SMSMCLASS:*data-class*, 2-30
- /SMSSCLASS:*data-class*, 2-30
- /SPANNED, 2-30
- /SUPERSEDE, 2-31
- /TRANSLATE:*file-spec*, 2-31
- /UNIT:*unit-spec*, 2-31, 2-32
- /USERID:*userid*, 2-32
- /VOLUME:(*vol-name*[,...]), 2-32
- /VSAM, 2-33
- /IGNORE:*condition*, 2-20
- INCASE keyword, 9-7
- INDEN keyword, 9-8
- INFDEF keyword, 9-9
- INFILE keyword, 9-7
- INLABEL keyword, 9-9
- INMDADDR keyword, 9-9
- INMDOWNER keyword, 9-10
- INMDPASS keyword, 9-10
- INNUM keyword, 9-10
- INPASS keyword, 9-10
- Input File Specification panel entry fields
 - DENSITY, 7-26, 7-29, 7-32, 7-37
 - FILE DEFINITION, 7-26, 7-29, 7-37
 - FILE NUMBER, 7-25, 7-29, 7-32, 7-37
 - MIXED CASE NAME, 7-24
 - PASSWORD, 7-11

Input File Specification panel entry fields
(cont'd)

TAPE LABEL, 7-25, 7-28, 7-32, 7-36
UNIT, 7-25
USERID, 7-11
VOLUME, 7-25

INRMT keyword, 9-11
INUNIT keyword, 9-11
INUSER keyword, 9-11
INVOL keyword, 9-12

ISPF

See Panel interface

K

/KEY:position,length[,position,length,...],
2-20

Keyed access to files, 3-10

L

/LABEL:label, 2-21

LIST subcommand, 9-28
example, 9-29

M

/MDADDRESS:disk-address, 2-21

/MDMPASSWORD:password, 2-21

/MDRPASSWORD:password, 2-21

/MDWPASSWORD:password, 2-21

Minidisks

specifying addresses, 2-21

specifying owners, 2-26

specifying passwords, 2-21

MIXED CASE NAME entry field, 7-24,
7-28, 7-30, 7-36

Modifying an entry in the file definition
database, 2-35

MS-DOS file specification format, 2-7

MVS Output File Options Panel entry fields

MIXED CASE NAME, 7-30

NULL RECORDS, 7-31

PRIMARY ALLOCATION, 7-31

SECONDARY ALLOCATION, 7-32

N

NAME file definition attribute, 2-43

NETMGR keyword, 9-20, 9-37

NFT commands

DECnet-RSX

/AP, 11-2

/AT, 11-11

/BR, 11-11

copy, 11-4

/DE, 11-9

/EX, 11-14

/FU, 11-11

/LI, 11-11

/SB, 11-14

supported commands, 1-10

MS-DOS

supported commands, 1-11

OS/2

supported commands, 1-11

PATHWORKS for DOS

APPEND, 12-2

COPY, 12-3

DELETE, 12-5

DIRECTORY, 12-7

SUBMIT, 12-9

TYPE, 12-10

PATHWORKS for OS/2

APPEND, 12-2

COPY, 12-3

DELETE, 12-5

DIRECTORY, 12-7

SUBMIT, 12-9

TYPE, 12-10

NODE entry field, 7-16

NULL RECORDS entry field, 7-31

/NULL:option, 2-26

O

OPEN command (OpenVMS DCL), 5-27

SOPEN RMS service call, B-16

Opening files

- See* Files, opening
- from OpenVMS DCL, 5-27
- OpenVMS file specification format, 2-5
- OS/2 file specification format, 2-7
- OUTCASE keyword, 9-12
- OUTDEN keyword, 9-14
- OUTFDEF keyword, 9-14
- OUTFILE keyword, 9-12
- OUTLABEL keyword, 9-14
- OUTMDADDR keyword, 9-15
- OUTMDOWNER keyword, 9-15
- OUTMDPASS keyword, 9-15
- OUTNUM keyword, 9-16
- OUTPASS keyword, 9-16
- OUTPRIMARY keyword, 9-16
- OUTRMT keyword, 9-17
- OUTSECONDARY keyword, 9-17
- OUTUNIT keyword, 9-17
- OUTUSER keyword, 9-17
- OUTVOL keyword, 9-18
- /OWNERID:owner, 2-26

P

Panel interface

- Data Translation Options panel, 7-39
 - introduction, 1-5
 - invoking, 7-1
 - MVS Input File Options panel, 7-24
 - MVS Output File Options panel, 7-30
 - Network File Options panel, 7-38
 - Operation Selection panel, 7-3
 - Post-Processing Options panel, 7-41
 - Receive from Remote panel, 7-10
 - Remote Node Options panel, 7-22
 - Resume Failed Transfer panel, 7-15
 - Send to Remote panel, 7-5
 - Transfer Execution panel, 7-18
 - VM Input File Options panel, 7-27, 7-42
 - VM Output File Options panel, 7-34
- SPARSE RMS service call, B-19
- Partitioned data set (PDS)
- /DIRECTORY_BLOCKS qualifier, 2-19
 - specifying member, 2-1

PASSWORD entry field, 7-11, 7-16

/PASSWORD:password, 2-27

Passwords

- specifying IBM, 2-27

PDS

See Partitioned data set (PDS)

Post processing

- with TRANSFER/DTF, 6-9

Post-processing

- using the panel interface, 7-41

PRIMARY ALLOCATION entry field, 7-31, 7-36

Printer files

- restrictions, 3-9

Proxy

- benefits, 4-4
- DECnet proxy, 4-7
- introduction, 4-4
- OpenVMS proxy, 4-5

Proxy database

- changing default entry, 2-18

SPUT RMS service call, B-19

Q

Queues

- canceling jobs, 6-20
- defining attributes, 6-14
- displaying characteristics of, 6-19
- displaying job status of entries, 6-17
- SNADTFSQUEUE, 6-14
- specifying with TRANSFER/DTF COPY command, 6-9

R

READ command (OpenVMS DCL), 5-28

Reading individual file records

- from OpenVMS DCL, 5-28

Reading records from a file

See Files, reading records from

RECALL subcommand, 9-29

- example, 9-29

RECDEF keyword, 9-19
 Receive from Remote Panel
 POSTPROCESS, 7-14
 Record attributes
 supported for input files, 3-3
 supported for output files, 3-7
 Record formats
 supported for input files, 3-2
 supported for output files, 3-5
 Record lengths
 supported for input files, 3-3
 supported for output files, 3-6
 Record level access
 CLOSE command, 5-8
 introduction, 1-8
 OPEN command, 5-27
 READ command, 5-28
 restrictions for, B-2
 IBM non-VSAM files, B-5
 VSAM files, B-2
 VSAM indexed files, B-4
 VSAM relative files, B-4
 VSAM sequential files, B-5
 WRITE command, 5-35
 /RECORD_DEFINITION:*record-definition*,
 2-27, F-1
 Recoverable copy operations
 See also RECOVERY keyword
 introduction, 1-7
 /RECOVER TRANSFER/DTF qualifier,
 6-9
 RECOVERY keyword, 9-19, 9-37
 /RELEASE, 2-27
 \$RELEASE RMS service call, B-19
 Remote backup and restore operations, 10-5
 REMOVE FILE_DEFINITION
 (SNADTFCFG) command, C-8
 REPLICATE file definition attribute, 2-43
 RESUME subcommand, 9-30
 canceling, 9-31
 CLIENT keyword, 9-33
 CLIENTPASS keyword, 9-32
 CLIENTUSER keyword, 9-32
 example, 9-34
 minimum entry, 9-31
 RESUME subcommand (cont'd)
 RETNUM keyword, 9-32
 /RETENTION_PERIOD:*n*, 2-27
 SRETNUM, 9-46
 RETNUM keyword, 9-32
 RETRY NUMBER entry field, 7-16
 Retry Request panel entry fields
 CLIENT USERID, 7-16
 NODENAME, 7-16
 PASSWORD, 7-16
 REUSE file definition attribute, 2-43
 \$REWIND RMS service call, B-19
 /RFA, 2-28
 RMS file system constraints, B-1
 RMS service calls
 \$CLOSE, B-7
 \$CONNECT, B-7
 \$CREATE, B-8
 \$DELETE, B-11
 \$DISCONNECT, B-11
 \$DISPLAY, B-12
 \$ERASE, B-14
 extended attribute blocks, B-3
 \$FIND, B-14
 \$FREE, B-14
 \$GET, B-15
 introduction, 1-8
 \$OPEN, B-16
 \$PARSE, B-19
 programming restrictions, B-2
 \$PUT, B-19
 record locking, B-3
 \$RELEASE, B-19
 \$REWIND, B-19
 \$SEARCH, B-20
 supported, 1-11, B-5
 unsupported, B-2
 \$UPDATE, B-20
 when accessing non-VSAM files, B-5
 when accessing VSAM indexed files, B-4
 when accessing VSAM relative files, B-4
 when accessing VSAM sequential files,
 B-5

RSX-11M/M-PLUS file specification format,
2-6

S

SEARCH command

OpenVMS DCL, 5-29

SEARCH RMS service call, B-20

Searching for strings in files

See Files, searching for strings in

SECONDARY ALLOCATION entry field,
7-32, 7-36

/SECONDARY_ALLOCATION:*number-of-*
512-byte-blocks, 2-28

/SECURITY_DATA:*data*, 2-29

/SEQUENCE_NUMBER:*n*, 2-29

Server account

specifying server account name, 2-1,
9-11, 9-17, 9-20, 9-37

SERVER keyword, 9-20, 9-37

Server node

specifying node name, 2-1, 8-5, 8-10,
9-11, 9-17, 9-20, 9-34, 9-37

Server software, 1-2

SET QUEUE (TRANSFER/DTF) command,
6-14

SET subcommand, 9-36

SHARE_OPTIONS file definition attribute,
2-44

SHOW ERROR (TRANSFER/DTF) command,
6-21

SHOW FILE_DEFINITION (SNADTFCFG)
command, C-3

SHOW JOBS (TRANSFER/DTF) command,
6-17

SHOW QUEUE (TRANSFER/DTF)
command, 6-19

SHOW subcommand, 9-38
examples, 9-38

SHOW VERSION (TRANSFER/DTF)
command, 6-20

/SINGLE, 2-29

/SMSDCLASS:*data-class*, 2-30

/SMSMCLASS:*data-class*, 2-30

/SMSSCLASS:*data-class*, 2-30

SNADTFCFG commands

ADD FILE_DEFINITION, C-1

REMOVE FILE_DEFINITION, C-8

SHOW FILE_DEFINITION, C-3

SNADTFCFG utility, 1-2

/SPANNED, 2-30

Strings

searching for

from OpenVMS DCL, 5-29

SUBMIT command

OpenVMS DCL, 5-31

PATHWORKS for DOS NFT, 12-9

PATHWORKS for OS/2 NFT, 12-9

RSX-11M/M-PLUS DCL, 11-14

/SUPERSEDE, 2-31

SYSTSIN DD card, 9-40

T

Tab characters in files, 3-10

TAPE LABEL entry field, 7-25, 7-28, 7-32,
7-36

Tape-resident files

restrictions, 3-9, 9-39

specifying density, 2-18, 9-8, 9-14

specifying sequence number, 2-29, 9-10,
9-16

specifying tape label, 2-21, 9-9, 9-14

specifying unit, 2-32, 9-11, 9-17

Transfer parameters

EXTENDED, 7-9, 7-14

NETWORK SUBSYSTEM, 7-4

POSTPROCESS, 7-9

RECOVERABLE, 7-13

SERVER NODE, 7-4

TRANSLATE, 7-9, 7-14

TRANSFER subcommand, 9-23

canceling, 9-24

examples, 9-26

POST keyword, 9-23

prompts, 9-24

TRANSFER/DTF utility, 1-2

- commands
 - CANCEL JOB, 6-20
 - COPY, 6-3
 - SET QUEUE, 6-14
 - SHOW ERROR, 6-21
 - SHOW JOBS, 6-17
 - SHOW QUEUE, 6-19
 - SHOW VERSION, 6-20
- DEFAULT INTERVAL value, 6-14
- DEFAULT RETRIES value, 6-14
- DEFAULT TIME value, 6-14
- SENTRY, 6-3
- exiting, 6-3
- invoking, 6-2
- MAXIMUM RETRIES value, 6-15
- MAXIMUM TIME value, 6-15
- MINIMUM INTERVAL value, 6-16
- qualifiers
 - /AFTER, 6-6
 - /CHARACTERISTICS, 6-6
 - /CHECKPOINT_INTERVAL, 6-6
 - /HOLD, 6-7
 - /IDENTIFY, 6-7
 - /INTERVAL, 6-7
 - /LOG_FILE, 6-7
 - /NAME, 6-8
 - /NOTIFY, 6-8
 - /POST_PROCESSING, 6-9
 - /PRIORITY, 6-9
 - /QUEUE, 6-9
 - /RECOVER, 6-9
 - /RETRIES, 6-9
 - /SINCE, 6-10
 - /TIME, 6-10
- supported commands, 1-9
- Transferring files
 - between MS-DOS and IBM, 12-1
 - between RSX-11M/M-PLUS and IBM, 11-1
 - between OpenVMS and IBM, 5-1
 - between OS/2 and IBM, 12-1
 - between ULTRIX, UNIX, and IBM, 10-1

- TRANSLATE keyword, 9-21
- /TRANSLATE qualifier, F-1
- /TRANSLATE:*file-spec*, 2-31
- Translating data types, F-1
- Translation modes
 - DMCS to EBCDIC, F-1, F-2
 - field level translation, F-1, F-2
 - no translation, F-1, F-2
- Translation of Digital and IBM data types, F-4
- Translation tables
 - DMCS to EBCDIC, F-3
- TYPE command
 - OpenVMS DCL, 5-33
 - PATHWORKS for DOS NFT, 12-10
 - PATHWORKS for OS/2 NFT, 12-10
 - RSX-11M/M-PLUS DCL, 11-16

U

- ULTRIX file specification format, 2-8
- UNIT entry field, 7-25
- /UNIT:*unit-spec*, 2-31, 2-32
- \$UPDATE RMS service call, B-20
- USERID entry field, 7-11, 7-16
- /USERID:*userid*, 2-32
- Using DATATRIEVE, A-1
- Utilities software, 1-2

V

- Version number
 - displaying, 6-20
- VM Input File Options Panel entry fields
 - MIXED CASE NAME, 7-28
- VM Output File Options Panel entry fields
 - MIXED CASE NAME, 7-36
 - PRIMARY ALLOCATION, 7-36
 - SECONDARY ALLOCATION, 7-36
- VOLUME entry field, 7-25
- /VOLUME:(*vol-name*[,...]), 2-32
- /VSAM, 2-33
- VSAM files
 - cluster attribute defaults, D-8
 - copying using the CONVERT command, 5-9

VSAM files (cont'd)

- creating file definitions, 2-33
- DTF naming conventions, D-6
- indexed file structure, D-2
- /KEY qualifier, 2-20
- naming ALTERNATE_INDEXes, D-7
- naming conventions, D-5
- record level access to, B-2
- relative file structure, D-1
- restrictions, 3-8
- /RFA qualifier, 2-28
- RMS programming considerations, B-4, B-5
- sample file definition, 2-38
- sequential file structure, D-1
- specifying buffer space for, 2-42

- specifying component allocation, 2-42
- specifying component names, 2-43
- specifying file sharing, 2-44
- structures, D-1
- supported as input files, 3-1
- supported as output files, 3-4
- VSAM naming conventions, D-6
- /VSAM qualifier, 2-33

W

WRITE command (OpenVMS DCL), 5-35

Writing individual file records
from OpenVMS DCL, 5-35

Writing records to a file

See Files, writing records to