

Data Transfer Facility for Digital UNIX

Client DTF

Order Number: AA-QUPJA-TK

May 1996

This manual describes how to install, customize, and use the DTF software.

**Digital Equipment Corporation
Maynard, Massachusetts**

Digital Equipment Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

Possession, use, or copying of the software described in this publication is authorized only pursuant to a valid written license from Digital or an authorized sublicensor.

Copyright 1996 Digital Equipment Corporation. All rights reserved. Printed in U.S.A.

The following are trademarks of Digital Equipment Corporation:

The DIGITAL logo, AlphaServer, AlphaStation, DEC, DECnet, Digital, OpenVMS, ULTRIX, VAX, and VMS.

The following are third-party trademarks:

IBM, OS/2, OS/400, and PS/2 are registered trademarks of International Business Machines Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company.

All other trademarks and registered trademarks are the property of their respective holders.

CONTENTS

1. INSTALLATION	1-1
1.1 Inspecting the Distribution Kit	1-1
1.2 Preparing the IBM SNA Environment	1-1
1.2.1 Required Software	1-2
1.2.2 Installation Time	1-2
1.2.3 Preparing to Install	1-2
1.3 Installing the DTF/DU Software	1-5
1.3.1 Installing from a CD-ROM Drive	1-5
1.3.2 Possible Installation Errors	1-8
1.4 Running the Installation Verification Procedure	1-8
1.5 IBM Initiated Transfers	1-9
1.6 Accessing the DTF/DU man Pages	1-9
1.7 Deleting the DTF/DU Software from Your System	1-10
1.7.1 Deleting DTF/DU Software	1-10
1.7.2 Deleting Other DTF/DU Files	1-10
1.8 Identifying and Reporting Problems	1-11
2. DATA TRANSFER FACILITY CLIENT FOR DIGITAL UNIX (DTF/DU)	2-1
2.1 Defining the DTF/DU Product	2-1
2.2 DTF/DU Product Capabilities	2-1
2.2.1 Multiple User Interfaces	2-2
2.2.2 Data Translation	2-2
2.2.3 Multiple File Types Support	2-3
2.2.4 Remote Job Submission and Post Processing	2-3
2.2.5 Remote File Printing on an MVS system	2-3
2.2.6 Directory Capability (Digital UNIX Clients)	2-3
2.2.7 Recoverable File Transfer	2-3
2.3 DTF/DU Product Benefits	2-4
	iii

3. CONFIGURATION FILES	3-1
3.1 ALIAS Configuration Files	3-1
3.1.1 ALIAS Configuration File Content	3-2
3.1.2 Customizing the ALIAS Configuration File	3-5
3.2 .rhosts Authorization File	3-5
4. ADMINISTRATIVE AND USER FILES	4-1
4.1 SYSTEM PREFERENCES Administrative File	4-1
4.2 SYSTEM UNIX PROLOG Files	4-3
4.3 SYSTEM IBM PROLOG File	4-5
4.4 USER PREFERENCES File	4-6
4.5 USER UNIX PROLOG File	4-6
4.6 USER IBM PROLOG File	4-6
4.7 USER JOB Files	4-7
4.8 DISPATCHER Files	4-8
4.9 TRANSLATE TABLE Files	4-9
4.10 EXAMPLE Files	4-9
5. DTF/DU FILE TRANSFER COMMANDS	5-1
5.1 DTF/DU Command Line Parameters	5-1
5.1.1 DTF/DU Flags	5-1
5.1.2 Environmental Variables Referenced by DTF/DU	5-4
5.1.3 DTF/DU Parameters	5-4
5.1.4 DTF/DU and MVS Filenames	5-6
5.1.5 DTF/DU and UNIX Filenames	5-6
5.2 dtfui - Copy UNIX File to MVS System	5-6
5.2.1 dtfui MVS File Qualifiers	5-7

5.2.2 dtfui EXAMPLES	5-10
5.3 dtfui - Copy UNIX File from MVS System to Digital UNIX	5-10
5.3.1 dtfui MVS File Qualifiers	5-11
5.3.2 Examples	5-12
5.4 dtfls Command - List Statistics for MVS File(s)	5-12
5.4.1 dtfls MVS File Qualifiers	5-13
5.4.2 dtfls Examples	5-14
5.5 dtfpr Command - Print a MVS File	5-15
5.5.1 dtfpr MVS File Qualifiers	5-15
5.5.2 dtfpr Examples	5-16
5.6 dtfrm - Remove (Deletes) an MVS File	5-16
5.6.1 dtfrm MVS File Qualifiers	5-16
5.6.2 dtfrm Examples	5-17
5.7 dtfsb SUBMIT MVS File for Execution	5-18
5.7.1 dtfsb MVS File Qualifiers	5-18
5.7.2 dtfsb Examples	5-19
6. DTF/DU FILE SPECIFICATIONS	6-1
6.1 Specifying Remote MVS Files for DTF/DU File Access	6-1
6.2 MVS File Specification Optional Flags	6-2
6.2.1 Optional Flags Defaults	6-2
6.2.2 Qualifier Restrictions When Accessing MVS files	6-3
6.2.3 Qualifier Descriptions	6-4
7. SUPPORTED MVS FILE TYPES	7-1
7.1 Supported MVS Input File Types	7-1
7.1.1 Supported File Organizations	7-1
7.1.2 Supported Record Formats	7-1
7.1.3 Supported Record Lengths	7-1
7.1.4 Supported Record Attributes	7-1
7.2 Supported MVS Output File Types	7-1
7.2.1 Supported File Organizations	7-2
7.2.2 Supported Record Formats	7-2
7.2.3 Supported Record Lengths	7-2
7.2.4 Additional File Attributes	7-3

7.2.5 Files with ANSI Control Characters	7-3
7.3 File Transfer Considerations	7-3
7.3.1 VSAM File Transfer Restrictions	7-3
7.3.2 Transferring MVS Tape-Resident Files	7-3
7.3.3 Transferring Files with Horizontal Tab Characters	7-4
8. DATA SECURITY	8-1
8.1 Data Security Rules	8-1
8.2 Using USERIDs and PASSWORDs	8-1
8.3 Proxy	8-2
8.4 Proxy Access to the MVS System	8-2
8.5 Proxy Access to the Digital UNIX System	8-2
9. RECOVERABLE COPIES	9-1
9.1 Recoverable Copy Checkpoints	9-1
9.2 Status Display of Recoverable Copies	9-1
9.3 Recovering an Interrupted Transfer	9-2
9.3.1 MVS Initiated Transfers	9-2
9.3.2 UNIX Initiated Queued Jobs	9-2
10. DTFJOB - DISPLAY AND CONTROL QUEUED JOBS	10-1
10.1 dtfjob Description	10-1
10.1.1 dtfjob Actions Associated With a Single Job	10-1
10.1.2 dtfjob Actions Associated With Multiple Jobs	10-2
10.1.3 Output of 'list' and 'show' Action	10-3
10.1.4 User Job States	10-4
10.2 dtfjob Examples	10-5

11. ADMINISTRATIVE COMMANDS	11-1
11.1 Administrative Command: dtfboot	11-1
11.2 Administrative Command: dtfdsp	11-2
11.2.1 dtfdsp Examples	11-4
12. PROBLEM DETERMINATION	12-1
12.1 DTF/DU Components	12-1
12.1.1 DTF/DU Commands	12-1
12.1.2 Dispatcher	12-2
12.2 MDU Components	12-3
12.3 Walkthroughs and Component Diagrams	12-4
12.4 Estimating Resource Utilization	12-7
12.4.1 Storage Estimates	12-8
12.4.2 Process Estimates	12-8
12.5 Isolating DTF/DU Software Problems	12-9
12.5.1 Installation Problems	12-9
12.5.2 Usage Problems	12-10
12.6 Information Gathering	12-10
12.7 Messages and Codes	12-11
13. SAMPLE INSTALLATION DIALOG	13-1
14. FILES INSTALLED ON YOUR SYSTEM	14-1
14.1 Files Created	14-1
14.2 Softlinks Created	14-4

Tables

TABLE 1 ACRONYM LIST	x
TABLE 3-1 DATA TYPES AND SIZES	3-2
TABLE 5-1 DATA TRANSFER MODES AND TRANSLATE OPTIONS	5-3
TABLE 6-1 IBM FILE SPECIFICATION QUALIFIERS	6-3
TABLE 10-1 USER_HOLD JOB ACTIONS	10-4
TABLE 10-2 EXECUTING STATE ACTIONS	10-4
TABLE 10-3 SYS_HOLD STATE EXIT ACTION	10-4
TABLE 10-4 STOPPED STATE EXIT ACTION	10-5
TABLE 10-5 FAILED STATE EXIT ACTION	10-5
TABLE 10-6 COMPLETED STATE EXIT ACTION	10-5

Associated Documents

- *Mainframe Data Transfer Facility for Digital UNIX VI.0* (AA-QTFNA-TK)
- *DECnet SNA Gateway for Channel Transport Installation* (AA-MA07E-TE)
- *DECnet SNA Gateway for Channel Transport Guide to IBM Parameters* (AA-LU36E-TE)
- *DECnet SNA Gateway for Channel Transport Problem Solving* (AA-LU37E-TE)
- *DECnet SNA Gateway for Channel Transport Management* (AA-LU43E-TE)
- *DECnet SNA Gateway for Synchronous Transport Installation* (AA-JE89E-TE)
- *DECnet SNA Gateway for Synchronous Transport Guide to IBM Parameters* (AA-JE91E-TE)
- *DECnet SNA Gateway for Synchronous Transport Problem Solving* (AA-MA17E-TE)
- *DECnet SNA Gateway for Synchronous Transport Management* (AA-LU43E-TE)
- *DEC SNA Peer Server Installation and Configuration* (AA-Q1P8C-TE)
- *DEC SNA Peer Server Management* (AA-Q1PAC-TE)
- *DEC SNA Peer Server Network Control Language Reference* (AA-Q1PBC-TE)
- *DEC SNA Domain Gateway Management 2.0* (AA-PQCFC-TE)
- *DEC SNA Domain Gateway NCL Reference* (AA-QCLDA-TE)
- *DEC SNA Domain Gateway Installation OSF* (AA-QCLBA-TE)
- *DEC SNA Domain Gateway Planning and Configuration* (AA-QCLCA-TE)
- *VTAM V4R3 Resource Definition Reference* (SC31-6552)
- *VTAM V4R3 Customization* (LY43-0068)
- *VTAM V4R3 Diagnosis* (LY43-0069)
- *VTAM Messages and Codes* (SC31-6546)

Abbreviations and Acronyms

Table 1 lists the acronyms that are used throughout this manual

Table 1 Acronym List

ASCII	American Standard Code for Information Interchange
DMCS	Digital Multinational Character Set
DTF/DU	Data Transfer Facility for Digital UNIX
EBCDIC	Extended Binary Coded Decimal Interchange Code
ESDS	Entry Sequenced Data Set
GDG	Generation Data Group
HSM	Hierarchical Storage Manager (IBM)
MVS	Multiple Virtual Storage system (IBM)
SNA	IBM Systems Network Architecture (IBM)
ISPF	Interactive System Productivity Facility
KSDS	Key Sequenced Data Set
MDU	Mainframe Data Transfer Facility Server for Digital UNIX (IBM)
PDS	Partitioned Data Set
RRDS	Relative Record Data Set
SMS	Storage Management Subsystem (IBM)
TSO	Time Sharing Option (IBM)
VSAM	Virtual Storage Access Method (IBM)
VTAM	Virtual Telecommunications Access Method (IBM)

1. Installation

This chapter tells you how to prepare for, and install, the Data Transfer Facility client for Digital UNIX (DTF/DU) application software on the Digital UNIX operating system. Topics covered here include an overview of the installation procedure, system requirements, software license Product Authorization Key (PAK) requirements, and other system dependencies.

Refer to Chapter 12, Problem Determination, for component descriptions of the system.

1.1 Inspecting the Distribution Kit

The software Bill of Materials (BOM) shipped with your distribution kit shows the contents of the kit. Carefully compare the items you received against the BOM. Report any missing or damaged components to Digital Equipment Corporation before continuing with the installation.

1.2 Preparing the IBM SNA Environment

Before installing and using the DTF/DU software, you must prepare certain software components in the IBM environment. Depending on the type of DEC/SNA gateway available at your site, you may need to refer to one of the following documents:

- *DEC Peer Server Guide to IBM Resource Definition*
- *DEC SNA Domain Gateway-CT Guide to IBM Resource Definition*
- *DEC SNA Domain Gateway-ST Guide to IBM Resource Definition*
- *DECnet SNA Gateway for Channel Transport Management*
- *DECnet SNA Gateway for Synchronous Transport Management*

You should view the Digital documentation as an adjunct to the IBM documentation; the IBM documentation remains the authoritative source for IBM software concepts and procedures.

Note

Since MVS system programmers generally reconfigure and generate their systems according to a set schedule, give them as much advance notice as possible.

1.2.1 Required Software

The following software is required to support the DTF/DU application:

- Digital UNIX V3.2C or later

See the Software Product Description (SPD) for a complete list of software required to support the DTF/DU software.

1.2.2 Installation Time

Installing the DTF/DU software requires up to 25 minutes, depending on the type of medium used.

1.2.3 Preparing to Install

Before you begin the actual installation, prepare your system as follows:

- For maintenance releases, check the currently installed version level.
- Obtain superuser privileges.
- Check for sufficient disk space.
- Back up your system disk. Digital strongly recommends that you perform a full system backup before installation.
- Read the product release notes.
- Register your Product Authorization Key (PAK).

The following subsections describe these procedures in detail.

1.2.3.1 Check the Current Version Level

If you are installing a maintenance release, verify that the version you are installing is later than the version currently on your system. To check the software version currently installed, enter the following command at the system prompt:

```
# what /usr/bin/dtfui [Return]
```

The version number appears at the end of the line that begins with:

```
Data Transfer Facility Client for Digital UNIX.
```

1.2.3.2 Obtain Superuser Privileges

To install the DTF/DU software, you must login as the root user or have superuser (su) privileges. If you are not logged in as root, issue the following command followed by the root password to acquire superuser privileges.

```
% su [Return]  
password: password [Return]
```

1.2.3.3 Check for sufficient disk space

Check system free disk space. You must have sufficient disk space to install the DTF/DU software. The installation will fail if there is insufficient space for installing the software from the distribution media. Minimum space requirements are as follows:

- 2 Mb in the /usr partition
- 1 Mb in the /var partition

To check the free space in the /usr partition, enter the following command:

```
# df -k /usr [Return]
```

1.2.3.4 Preparing your system for installation

Before installing the DTF/DU software, back up your system disk.

Also, prior to installing read release notes for the product. They include late breaking information that may affect installation.

1.2.3.5 Register the Product Authorization Key (PAK)

The DTF/DU software supports the License Management Facility (LMF). You must register your License Product Authorization Key (License PAK) in the License Database (LDB) before using the DTF/DU software. You can install the DTF/DU software before registering the PAK, but all file transfer commands and the IVP will fail.

LMF maintains a file of registered software license PAKs. Also, LMF keeps a library of functions used by Digital licensed software. To register the DTF/DU software license PAK using LMF, do the following:

- Log on to your system as superuser.

```
% su [Return]
```

```
password: password [Return]
```

- Issue the following command from the **su** prompt. Press **[Return]** after the confirmation message appears to initiate the registration process:

```
# lmfsetup [Return]
```

```
Register PAK (type q or quit to exit) [template] [Return]
```

- After you confirm the procedure, the system prompts for information related to the fields on the PAK form which ships as part of the DTF/DU software distribution kit when the license and media are ordered together. When ordered separately, Digital ships the PAK form to a location based on your license order.

Using the information from your DTF/DU software PAK, reply to each question as it appears onscreen. Leave blank any fields left blank in your PAK form.

- After you have answered all questions, the system issues the following completion message. Enter **quit (q)** and press **[Return]**:

```
Register PAK (type q or quit to exit)template q [Return]
```

- After leaving *lmfsetup*, issue the **lmf reset** command as follows:

```
# lmf reset [Return]
```

- If you attempt to load a PAK when a previous PAK is already installed, a message similar to the following appears:

```
Combine PAKNAME auth-num with PAKNAME auth
```

- After completing the LMF procedure, verify your registration as follows:

```
# lmf list [Return]
```

For further information concerning the use of the LMF software see `lmf(8)` and `lmfsetup(8)` man pages.

1.3 Installing the DTF/DU Software

This section describes the procedure for installing the DTF/DU software. Before using the software, you must register your software license PAK using LMF as described in Section 1.2.3.5.

To stop the installation at any time, press **Ctrl/C**. You must then manually delete any files created up to this point.

Note that Remote Installation Service (RIS) is not supported at this time.

1.3.1 Installing from a CD-ROM Drive

To install the DTF/DU software from the distribution CD-ROM, do the following:

- Determine the location of the DTF/DU files on the CD-ROM.
- Insert the CD-ROM into the drive and mount the disk using the drive's device name. If you do not know the device name, issue the following command to list available drives. The drive is either RRD40 or RRD42.

```
# file /dev/rr*c [Return]
```

To mount the disk, issue the following command, where *dev-name* is the drive device name.

```
# mount -r -d /dev/dev-name /mnt [Return]
```

- Install the DTF/DU software using the **setld** command, where *dtf100* is the name of the directory on the CD-ROM where the DTF/DU files reside:

```
# setld -l /mnt/dtf100 [Return]
```

After you enter **setld**, the system displays the following messages:

DTFDU for Digital UNIX, V1.0

Copyright Digital Equipment Corporation. 1994, 1995. All Rights Reserved.
This software is proprietary to and embodies the confidential technology of Digital Equipment Corporation. Possession, use, duplication or dissemination of this software and media is authorized only pursuant to a valid written license from Digital Equipment Corporation or an authorized sublicensor.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in Subparagraph (c)(1)(ii) of DFARS 252.227-7013, or in FAR 52.227-19, or in FAR 52.227-14 Alt. III, as applicable.

The subsets listed below are optional:

There may be more optional subsets than can be presented on a single screen. If this is the case, you can choose subsets screen by screen or all at once on the last screen. All of the choices you make will be collected for your confirmation before any subsets are installed.

- 1) DTF for DIGITAL UNIX - MAN Pages (optional)
- 2) DTF for DIGITAL UNIX - Run Time Components

Or you may choose one of the following options:

- 3) ALL of the above
- 4) CANCEL selections and redisplay menus
- 5) EXIT without installing any subsets

Enter your choices or press RETURN to redisplay menus.

- After the messages display, choose the appropriate DTF/DU subset installation for your site by responding to the prompt that appears as follows:

```
Choices (for example, 1 2 4-6):3 [Return]
```

Entering 3 at the command line installs both the DTF/DU MAN pages and the Run Time Components. The following messages are displayed.

You are installing the following optional subsets:

- DTF for DIGITAL UNIX - MAN Pages (optional)
- DTF for DIGITAL UNIX - Run Time Components

- The system then prompts you to verify your choice. If the choice is correct, respond to the prompt as follows:

```
Is this correct? (y/n):y [Return]
```

- The system displays the following messages.
 - Checking file system space required to install selected subsets:
 - File system space checked OK.
 - DTF for DIGITAL UNIX - Run Time Components
 - Copying from . (disk)
 - Verifying
 - DTF for DIGITAL UNIX - MAN Pages (optional)
 - Copying from . (disk)
 - Verifying
 - Configuring "DTF for DIGITAL UNIX - Run Time Components" (DTFRUN100)
- If you wish to edit the ALIAS configuration file for DTF/DU now respond to the following prompt with a 'y'. If you are not already familiar with the ALIAS file content, respond with a 'n'. A later chapter of this manual describes the ALIAS file and the methods for editing the ALIAS file.
 - Would you like to edit the alias file now?
(y/n) [y]:[Return]
 - The system invokes the editor defined by the environmental variable EDITOR to modify the default alias file. Brief instructions for editing the file are included within the default file. Editing the alias file may be deferred by exiting the edit session without making any changes to the alias file.
- Run the IVP for the DTF/DU by responding to the following prompt:
 - Would you like to run the IVP now? (y/n) [y]:[Return]
 - If you choose to run the IVP, the system displays the messages shown below. The IVP can be run even if the ALIAS file was not edited during installation..
 - DTFRUN100 IVP PHASE 1 of 5 in progress
 - DTFRUN100 IVP PHASE 2 of 5 in progress
 - DTFRUN100 IVP PHASE 3 of 5 in progress
 - DTFRUN100 IVP PHASE 4 of 5 in progress
 - DTFRUN100 IVP PHASE 5 of 5 in progress
 - DTFRUN100 IVP COMPLETE - NO ERRORS DETECTED
 - Configuring "DTF for DIGITAL UNIX - MAN Pages (optional)" (DTFMAN100)

1.3.2 Possible Installation Errors

If an error occurs during the installation procedure, the procedure displays one of the following failure messages:

File system space check

This error indicates that the installation procedure did not find enough space in the /usr or /var partition. Increase the amount of space available and retry the installation.

Required file name already in use

This error indicates that filenames required by DTFRUN100 are already in use. The installation of DTFRUN100 is terminated, and the existing files are not modified. This message is followed by a list of file names that caused the message. Rename, move, or delete each file and retry the installation.

Subset DTFRUN100 is already installed

This error indicates that DTFRUN100 has already been installed.

1.4 Running the Installation Verification Procedure

If you have chosen the IVP option, it runs automatically at the end of the installation. You can also run the IVP after the installation to verify that the DTF/DU software is installed and working properly. To run the DTF/DU IVP, use the following command:

```
# setld -v DTFRUN100 [Return]
```

After you enter the **setld** command, the system displays the following messages.

```
DTF for DIGITAL UNIX - Run Time Components (DTFRUN100)
DTFRUN100 IVP PHASE 1 of 5 in progress
DTFRUN100 IVP PHASE 2 of 5 in progress
DTFRUN100 IVP PHASE 3 of 5 in progress
DTFRUN100 IVP PHASE 4 of 5 in progress
DTFRUN100 IVP PHASE 5 of 5 in progress
DTFRUN100 IVP COMPLETE - NO ERRORS DETECTED
```

If the IVP detects an error condition, the following message is displayed:

```
DTFRUN100 IVP FAILURE
```

One or more additional messages should follow to explain the nature of the failure more fully.

```
DTFRUN100 file not installed, or deleted since installation
```

```
DTFRUN100 directory not created, or deleted since installation
```

```
DTFRUN100 soft link not created, or deleted since installation
```

Additional error messages document the name of the file, directory, or link name associated with the error condition.

1.5 IBM Initiated Transfers

In addition to the UNIX command line interface, there is an MDU user interface that allows IBM/MVS users to initiate file transfers. Refer to the MDU installation documentation. Once the MDU installation is completed, refer to the UNIX **dtfdsp** command for instructions on how to start a dispatcher (SNA) session with the MVS system. This session is required in order to manage IBM/MVS initiated requests.

1.6 Accessing the DTF/DU man Pages

If the optional saveset DTFMAN100 is installed, then man pages are available. To access the man page called **dtf**, enter the following command:

```
# man dtf [Return]
```

MAN pages in saveset DTFMAN100 and brief descriptions are as follows :

dtf	summary of interactive commands and error message explanation
dtfdsp	UNIX daemon for IBM initiated transfers
dtfui	copy a file from IBM to UNIX
dtfjob	manage queued (or batch) jobs
dtflslist	MVS file names and attributes
dtfpr	print MVS file at MVS system
dtfrm	remove (delete) a file from MVS system
dtfsb	submit an MVS file (JCL stream) for processing on the MVS system
dtfui	copy a file from UNIX to MVS

All of the man pages except **dtf** describe an individual command. The **dtf** man page contains a one line summary of the commands, a brief description of the environmental variables, and additional error message information.

1.7 Deleting the DTF/DU Software from Your System

If you find it necessary to delete the DTF/DU files from your system, use the procedure described in this section. This procedure does not delete all of the files and directories created during the installation process. Files not deleted include various configuration files, log files, and user-specific work areas. These files are needed if the reason for removing the software is to upgrade to a later version.

1.7.1 Deleting DTF/DU Software

Before deleting DTF/DU software, make sure that the software is not currently in use. In addition to verifying that no users are submitting interactive file transfer requests, make sure that no DTF/DU dispatcher processes are running and that UNIX **cron** has not scheduled any DTF/DU scripts on behalf of users. (see later sections for descriptions of DTF/DU commands **dtfdsp** and **dtfjob**).

NOTE:

Failure to ensure that DTF/DU scripts are not scheduled by UNIX **cron** before deleting DTF/DU software can result in UNIX **cron** sending numerous mail messages indicating failure.

To delete the software files from your system, log in as superuser and issue the **setld** command with the **-d** option, as follows:

```
# setld -d DTFRUN100 DTFMAN100 [Return]
```

1.7.2 Deleting Other DTF/DU Files

The **setld -d** command described above does not delete all directories that were created during the **setld -l** processing. In addition, the **setld -d** command does not delete any of the files created by DTF/DU commands. To remove all of these files and directories, use the following commands:

```
rm -fR /var/spool/dtf
rm -fR /etc/ibm_dtf*
rm -fR /var/tmp/dtf
rm -fR /usr/opt/DTF100
rm -fR /var/opt/DTF100
rm -fR /usr/.smdb./DTFRUN100*
rm -fR /usr/.smdb./DTFMAN100*
```

1.8 Identifying and Reporting Problems

Refer to the problem determination section of this guide when you encounter a problem with the DTF/DU component. Included in that section is documentation on how to debug problems and collect information.

Before contacting Digital, determine the version of the DTF/DU software and the Digital UNIX operating system software. Then, depending on the nature of the problem and the type of support you have, do one of the following:

- If your software contract or warranty agreement entitles you to telephone support, call Digital using the number supplied as part of your contract.
- Submit your query or suggestion electronically through Digital DSN or DSNLINK

Review the Software Product Description (SPD) and Warranty Addendum for an explanation of the product warranty. If you encounter a problem during the warranty period, report the problem as indicated above, or follow alternate instructions provided by Digital for reporting SPD nonconformance problems.

2. Data Transfer Facility Client for Digital UNIX (DTF/DU)

This chapter discusses the following topics concerning the DTF/DU product:

- Defining the DTF/DU product
- Capabilities of the DTF/DU product
- Benefits of the DTF/DU product

2.1 Defining the DTF/DU Product

The DTF/DU software is a Digital Equipment Corporation product that allows you to transfer files between an IBM MVS system and a Digital UNIX system in an SNA network. To use the DTF/DU interfaces, 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 purpose of DTF/DU is to support file transfer between MVS systems and Digital UNIX nodes, not between two Digital nodes. The DTF/DU routines consist of two parts as follows:

- IBM resident software Mainframe DTF for Digital UNIX (MDU) running on an MVS system
- Digital UNIX resident software (DTF/DU) running on Digital UNIX

A Digital/SNA gateway is also required, but is not part of the DTF/DU product. See the Software Product Description (SPD) for a list of the supported gateways. The Digital UNIX user interface consists of a set of commands entered via the UNIX command line interface. Commands are described in a later chapter.

The MVS system user interface is a menu-driven panel interface, a basic set of file transfer commands.

2.2 DTF/DU Product Capabilities

The DTF/DU software has the following capabilities:

- Multiple concurrent users on both Digital UNIX and MVS systems
- EBCDIC/ASCII Data translation

- Access to various MVS file types
- Remote job submission to MVS systems and post processing on UNIX nodes
- Remote file printing on an MVS system
- Directory capability of MVS file from Digital UNIX clients
- Recoverable file transfers

2.2.1 Multiple User Interfaces

The DTF/DU product provides Digital and IBM users with a familiar user interface. This allows users in both environments to transfer data using an interface to which they are already accustomed, significantly reducing the learning curve associated with a new software application. These interfaces are as follows:

- **Digital UNIX User Interfaces**

Most of the DTF/DU file transfer commands are patterned after similar UNIX commands. Besides the difference in command names, one of the file specification parameters is replaced with an MVS file specification. Refer to the chapter on file specification syntax for more information on the MVS file specification parameters and MVS file specification qualifiers.

- **MVS User Interfaces**

MDU provides three user interfaces for MVS clients that you can use to copy files. The interfaces for MVS clients areas follows:

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

Refer to the MDU documentation for information about these interfaces.

2.2.2 Data Translation

The DTF/DU product allows you to translate text between the Digital Multi-National Character Set (DMCS), an ASCII superset, and EBCDIC. Data translation is performed by default on all text mode transfers and inhibited for image mode transfers.

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

- Turn data translation on or off to override the default.
- Specify a user-defined translation table.
- Specify translation to be performed by the UNIX node or by the MVS system.

2.2.3 Multiple File Types Support

You can use the DTF/DU software to transfer a variety of Digital UNIX and MVS file types. Supported file types include the following MVS and Digital UNIX files:

- **Supported Digital UNIX**

On Digital UNIX systems, regular files are supported. For interactive transfers, STDIN and STDOUT are also supported. Other file types, including named pipes, are NOT supported.

- **Supported MVS Files**

On MVS, the MDU software can transfer and create VSAM (ESDS), physical sequential, and partitioned file types as follows:

VSAM ESDS and KSDS file types are supported for some operations.

PHYSICAL SEQUENTIAL files are supported in transfers between IBM clients and Digital UNIX. This includes partitioned data set (PDS) members on MVS clients. Both IBM disk- and tape-resident files are supported and can be transferred to and from files on the Digital UNIX system.

PARTITIONED (PDS and PDSE) file types can be created on the MVS system. As noted, members of existing PDS/PDSE data sets are supported.

2.2.4 Remote Job Submission and Post Processing

Digital UNIX users can use the DTF/DU product to submit an IBM resident file to IBM batch subsystems. In addition, IBM and Digital UNIX clients have the capability of specifying a command or script to execute on the Digital UNIX system after a successful file transfer.

2.2.5 Remote File Printing on an MVS system

Digital UNIX users can use the DTF/DU product to print an MVS resident file on an MVS system controlled printer.

2.2.6 Directory Capability (Digital UNIX Clients)

Digital UNIX clients can use DTF/DU software to obtain listings of files on IBM clients. All clients supporting this directory capability allow you to specify wild card characters for displaying groups of files.

2.2.7 Recoverable File Transfer

The DTF/DU and MDU software allow you to initiate a recoverable file transfer. If a recoverable file transfer operation fails because of a system or network error, you can resume the transfer from the last checkpoint after the system or network recovers. The checkpoint and recovery feature is a valuable time saver, especially

when transferring large amounts of data. Recoverable file transfer is not available in the following instances:

- UNIX file type is STDIN or STDOUT
- MVS file type is VSAM

2.3 DTF/DU Product Benefits

DTF/DU provides a UNIX command line interface to do the following:

- manipulate files resident on an MVS system
- copy files between a UNIX system and an MVS system

DTF/DU also supports an MVS interface that allows an MVS user to do the following:

- copy files between a UNIX system and an MVS system
- specify a UNIX command to execute on the UNIX node after a file transfer

3. Configuration Files

To configure the Digital UNIX resident component of the DTF/DU product, several ASCII text files must be edited or created. This chapter describes these files and their content.

The MVS component of this product (MDU), the IBM SNA network components, and the Digital/SNA gateway must also be configured. This chapter does NOT describe this process. Where appropriate, this chapter does identify the parameters on the Digital UNIX system that must match parameters in one of the other components.

3.1 ALIAS Configuration Files

For a DTF/DU command to successfully establish communications with an MVS system based MDU component, several parameters are required. These include the Digital/SNA gateway name and IBM SNA network parameter names. If a site had several MVS systems or several Digital/SNA gateways, there could be several different sets of valid and useful parameters.

Each set of parameters is assigned a name. The name is said to be an ALIAS entry name for the set of parameters. When a DTF/DU command is executed, the ALIAS entry name can be used to indicate a particular set of communications-related parameters.

ALIAS names and their associated parameters are stored in simple ASCII text files. Each ALIAS entry name and associated parameters are stored as one line of text. Each of these lines is called an ALIAS ENTRY. These files can also contain comment lines. A file that contains ALIAS ENTRIES is called an ALIAS FILE.

When each DTF/DU command executes, an ALIAS FILE is accessed to retrieve an ALIAS ENTRY. The DTF/DU command then has all the information it requires to reach a specific MDU component on a specific MVS system. If no ALIAS FILE is explicitly specified with the DTF/DU command, a default ALIAS FILE is used. The default file name is `/etc/ibm_dtf_alias`. If no ALIAS ENTRY NAME is explicitly specified with the DTF/DU command, the first entry in the file is used.

3.1.1 ALIAS Configuration File Content

Each line in the ALIAS file is either a comment or an entry. Comment lines are indicated by a pound sign/asterisk # or * in the first column. All other lines are considered entries. There is no limit on the number of entries in an ALIAS file.

Each ALIAS ENTRY must consist of nine or more positional parameters separated by spaces or tabs. Refer to Table 3-1 for a list of field data types and sizes. All parameters that accept alphabetical input are case insensitive.

Table 3-1 Data Types and Sizes

Entry name	<= 16 bytes	text (should start in column 1 of line)
Gateway node	<= 64 bytes	text
Transport	<= 6 bytes	LOCAL, TCP, or DECNET
Access name	<= 16 bytes	text
Application name	<= 8 bytes	text
Logmode	<= 8 bytes	text
PU	<= 2 bytes	decimal number (0, 1, ...) or '*'
Translate	<= 5 bytes	LOCAL, REMOTE, NONE
NULL	<= 7 bytes	none, ignore, value
DELAY	<= 8 bytes	decimal number

ALIAS FILE field definitions are as follows:

Entry name (Positional Parameter 1)

The user-defined name for ALIAS entry. This name may be used on the DTF/DU command line. The user may assign any name alphanumeric character string.

This entry name does NOT need to match any other parameter in the UNIX node, the Digital/SNA gateway, or the MVS system. The name is used only by the DTF/DU commands on the UNIX node when searching an ALIAS FILE. Duplicate entry names are not flagged as errors. The first entry is used.

Gateway node (Positional Parameter 2)

The node name of the node acting as the Digital/SNA gateway.

This Gateway node should be either the name or the alias of the Digital/SNA gateway node. If using a Peer Server running on the same node as the DTF/DU component, a value of 0 (number zero) can be used instead of the actual node name.

Transport (Positional Parameter 3)

Transport type in use between the Digital/UNIX node and Digital/SNA gateway. Pick a transport type that is available on both nodes. The possible transport types are DECnet, TCP, and local. Not all transport types are available with all Digital/SNA gateways. Possible combinations are:

- ◆ Peer server in same node TCP or DECnet or local
- ◆ Peer server in different node TCP or DECnet
- ◆ Domain CT or ST Gateway DECnet or TCP
- ◆ CT or ST gateway: DECnet

The Transport parameter does NOT need to match any other parameter on the Digital UNIX node, the Digital/SNA gateway, or the MVS system. It does need to match a transport type that is available on both the Digital UNIX node and on the Digital/SNA gateway node.

Access name (Positional Parameter 4)

An access name defined on the Digital/SNA gateway.

An access name corresponds to a pool of SNA logical units defined in the Digital/SNA gateway and in the IBM SNA network. An access name can optionally provide default values for an MVS application name and logmode table entry name.

During execution of a DTF/DU command, the access name is passed to the Digital/SNA gateway in order to select an individual LU for use for the duration of the DTF/DU command.

The access name must exactly match an access name defined on the Digital/SNA gateway. Refer to the gateway documentation for information on how to install access names.

Application name (Positional Parameter 5)

The name assigned on the MVS system to the MDU component or an *.

If an explicit application name is provided, it must exactly match the MVS system APPLID parameter value assigned to the MDU component.

If an * is used in place of an application name, then the default application name defined within the access name on the Digital/SNA gateway is used.

Logmode (Positional Parameter 6)

The name of the MVS system VTAM logmode table entry to use. A VTAM logmode table entry is a collection of parameters used to define the attributes of the IBM SNA sessions that are established between the MVS system and the Digital/SNA gateway.

If an explicit logmode name is provided, it must exactly match some entry name in the VTAM logmode table associated with the SNA logical unit used.

If an * is used in place of a Logmode entry name, then the default logmode name defined within the access name on the Digital/SNA gateway is used.

NOTE

When a Domain CT or Domain ST gateway is used, the logmode name must be defined on the Domain gateway as well as on the MVS system.

DTF/DU supports data compression between itself and the MVS system. To enable compression, the MVS system programmer must enable certain VTAM options as well as configure a logon mode entry for DTF/DU to use. If data compression is a feature you want to utilize, contact the MVS system programmer and request that they review the MDU documentation section that refers to VTAM logon mode tables.

PU - Physical Unit (Positional Parameter 7)

If the Gateway node parameter specifies a Peer Server node, a Domain ST gateway, or a Domain CT gateway, this parameter has no meaning and should have a value of '*'. Do not omit the parameter.

For non-Domain CT and non-Domain ST gateways, this parameter specifies a Physical Unit number. These gateway types may support multiple physical units as well as multiple logical units. An explicit physical unit number is used along with the access name by Digital/SNA gateway when selecting a logical unit to use for a DTF/DU command.

If an explicit PU number is provided, it must exactly match a physical unit number defined on the Digital/SNA gateway.

If an '*' is used in place of a PU number, then a logical unit (LU) on any of the physical units may be used (within the constraints of the access name definition).

Xlate (Positional Parameter 8)

Indicates the node where ASCII/EBCDIC translation takes place. Supported values are:

- ◆ local: translation performed on UNIX node
- ◆ remote: translation performed on MVS system
- ◆ none: no translation takes place

NOTE

Depending on the relative speeds and availability of processor cycles, the choice of translation sites can have an impact on performance.

Null (Positional Parameter 9)

This option indicates the action to take with null text lines when sending data

to the MVS system. In this context, a null text line in a UNIX file is a line that consists of only a newline character. The MVS file system uses a 'logical record length' instead of newline characters.

This option indicates the action that MDU takes when it receives a null text line. The choices are:

- ♦ none: write null record to MVS file system
- ♦ ignore: discard null records
- ♦ value: write a 1 byte record with the value specified

To specify a value, use decimal, octal, or hex notation to specify a value. For instance, to specify use of an EBCDIC upper case letter A, code 193, or 0301, or %XC1.

Delay: (optional)

This optional field specifies the number of microseconds to delay during the startup phase of each DTF/DU command. A delay is sometimes required with some non-Domain CT and non-Domain ST gateways. This parameter should normally be omitted or specified as zero.

3.1.2 Customizing the ALIAS Configuration File

The default ALIAS file is created with write access restricted to root. Only root and super users can update the ALIAS file. Modify the default ALIAS file using a text editor or the configure option of the setld command as follows:

```
# vi /etc/ibm_dtf_alias
# setld -c DTFRUN100 edit
```

Any updates take effect for all DTF/DU processes started after the file is updated. This includes IBM initiated transfers and UNIX initiated batch jobs. The change does not impact DTF/DU processes that have already started execution. To use an ALIAS file other than the default file, use the *-A ALTERNATE_ALIAS_FILE* parameter on DTF/DU commands (or *_DTF_USER_FLAGS* environmental variable).

3.2 .rhosts Authorization File

When an MVS user requests a file transfer operation, the request is forwarded to a DTF/DU process on the UNIX node called a "dispatcher" (The chapter Administrative Commands describes the command to control these processes). When the MVS-initiated request is received, the Digital UNIX based dispatcher validates the UNIX userid and password supplied in the request. Then, the UNIX **rsh** command is used to start a new process that runs under the UNIX userid specified by the IBM user.

The Digital/UNIX **rsh** command imposes its own security requirements. For each UNIX userid that an MVS user will specify in a file transfer command, a file

named `.rhosts` must be defined in the target UNIX users' home directory. The file must indicate that the DTF/DU dispatcher is allowed to invoke processes via **rsh** on the UNIX user's behalf. The Digital UNIX supplied man page for `rhosts` has further details.

The script shown below could be used to create a new `.rhosts` file for user 'joe' on a node 'axpnode'. The single line in the new file corresponds to a DTF/DU dispatcher running under the root userid.

```
#!/usr/bin/ksh
##
if [ -f ~joe/.rhosts ] ; then
    print ".rhosts already exists for account joe"
    exit 1
fi
##
print "axpnode root" >~joe/.rhosts
chmod 600 ~joe/.rhosts
chown joe:users ~joe/.rhosts
```

NOTE

The `.rhosts` file is needed only for DTF/DU operations initiated from the MVS system. The file is NOT needed when DTF/DU operations are requested from the UNIX command line interface.

4. Administrative and User Files

Data Transfer Facility Client for Digital UNIX (DTF/DU) creates and uses a variety of files. Some of these files modify the functions performed by DTF/DU. Other files are created as transfer requests execute. These files are NOT automatically deleted. Instead, they are retained for audit and use. These files should be purged when no longer needed.

The administrative and user files are as follows:

- System Preferences
- System UNIX Prolog
- System IBM Prolog
- User Preferences
- User UNIX Prolog
- User IBM Prolog
- User Job Files
- Dispatcher Files

NOTE

Default administrative files are created during the installation process. None of the files discussed in this chapter need to be created, modified, or deleted before using DTF/DU.

4.1 System Preferences Administrative File

The system preferences file provides a way to customize the operation of queued and MVS initiated jobs. These files have no impact on UNIX-initiated DTF/DU commands (those initiated from the command line without the *-q* flag).

A systems preferences file is created as part of the initial product installation procedure. The file name is `/etc/ibm_dtf_defaults/system_preferences`. This file is included in every shell script generated to service IBM- and UNIX-initiated queued transfer requests. All DTF/DU generated scripts use the Korn shell (`/usr/bin/ksh`). This file must conform to all Korn shell requirements.

NOTE

To modify the system preferences file, log in as root or superuser and then execute the DTF/DU command `dtfjob sys_pref`.

The only capability supported for use in this file is to define the environmental variables described below. Use of this file for any other purpose is not supported.

- `__DTF_MAX_RETRY` 20
- `__DTF_MIN_TIME_DELAY` 10
- `__DTF_INC_TIME_DELAY` 20
- `__DTF_MAX_BATCH_RUN` 3
- `__DTF_MAX_IBM_RUN` 3
- `__DTF_MAX_JOB_NUMBER` 00999
- `__DTF_MAX_LOCK_TO` 5

`__DTF_MAX_RETRY`

Maximum number of attempts to retry a recoverable file transfer. UNIX-initiated queued jobs that fail with a transient error are retried. For example, transfers that terminate due to network failures or system outages are considered transient. To prevent a job from retrying forever, an upper retry limit is established with this parameter. The maximum supported value is 999. The minimum value, 0, indicates No Retries. There is no way to specify *retry forever*. This parameter does not affect IBM initiated jobs.

`__DTF_MIN_TIME_DELAY`:

Minimum time in minutes to wait before retrying a job. This value can be used to prevent a job from being retried immediately after a failure. The actual minimum delay time can be up to 59 seconds shorter than the number of minutes specified by this parameter. Thus, a value of 10 actually means “at least 9 minutes and 1 second”. This parameter has no impact on IBM initiated jobs. Minimum value is zero.

`__DTF_INC_TIME_DELAY`

Minimum additional time in minutes to wait before retrying a job on each successive retry after the first. This value can be used to prevent a single job that fails repeatedly from hogging system resources. For retries after the first retry, this value is multiplied by the retry number to derive a minimum time to wait. This parameter has no impact on IBM initiated jobs.

__DTF_MAX_BATCH_RUN

Limit on number of batch jobs concurrently executing for a user. This parameter can be used to prevent DTF/DU from starting an excessive number of batch jobs for a user simultaneously. When the number of batch jobs for a user reaches this value, other batch jobs will not be started. This applies to both new queued requests and to retries of previous jobs even when the minimum wait time has expired. This variable has no impact on interactive jobs, on IBM initiated jobs, or on manually executed batch jobs (see description of the **dtfjob** command, *exec* parameter).

__DTF_MAX_IBM_RUN

Limit on number of batch jobs concurrently executing for a user. This parameter can be used to prevent DTF/DU from starting an excessive number of IBM initiated transfers. When the number of IBM initiated jobs for a user reaches this value, additional IBM initiated transfer requests will be rejected with an appropriate error message. This variable has no impact on UNIX initiated file transfer requests.

__DTF_MAX_JOB_NUMBER

Maximum number of jobs that can be defined for each user. Each UNIX-initiated queued transfer request, and each IBM-initiated transfer is assigned a JOB number which remains in use until the job is explicitly deleted by the user with the **dtfjob** command. (The intent is to provide an audit trail for users). If the total number of jobs is exceeded, subsequent attempts to define jobs fail. The minimum supported value for this parameter is 00099. The maximum value is 99999. This parameter should be specified as a five digit decimal number using leading zeros as needed.

__DTF_MAX_LOCK_TO

A mutual exclusion lock is maintained on each user batch- and IBM-initiated job to prevent multiple processes from attempting to concurrently update a job's status. Under very heavy system load conditions, a process that has acquired the lock may abnormally terminate without releasing it. To prevent this condition from preventing all access to user jobs, a timeout is used. When a process fails to acquire the required lock for the time-out interval, the process checks to make sure that the process holding the lock is still executing. This variable specifies the lock timeout value in seconds.

NOTE

DTF/DU attempts to use the */proc* file system to verify the existence of the lock holding process. On systems that do not mount the */proc* file system, the timeout lock check procedure is not performed.

4.2 System UNIX Prolog Files

The system UNIX prolog file provides a way to customize each queued job that is defined. These files are copied into the queued job scripts when the job is initially

defined. This file has no impact on IBM initiated transfer requests or on nonqueued UNIX initiated requests (those without the *-q* flag).

A single systemwide system UNIX prolog file, `/etc/ibm_dtf_defaults/system_prolog_unix`, is created during initial product installation. All DTF/DU generated scripts use the Korn shell (*/usr/bin/ksh*). This file must conform to all Korn shell requirements.

The only capability supported for use in this file is to define the environmental variables described below. Use of this file for any other purpose is not supported.

NOTE

To modify the system UNIX prolog file, log in as root or superuser and then execute the DTF/DU command **dtfjob sys_unix**.

The supported environmental variables are:

SNALOG_MASK

Set this variable to select the type of trace records that should be included in the trace file. Refer to the Problem Determination chapter for more information

SNALOG_FILE

Set this variable to the file name to be used as a trace file. Unless overridden with a leading */*, *~*, or *variable name*, the file is created in the current working directory of the executing batch job. Use the **dtfjob** command with the *dir* option to display the full path name of a current working directory. Refer to the Problem Determination chapter for more information

SNALOG_SIZE

Set this to the maximum log file size allowed. The value is specified in 512 byte blocks. When the log file size is reached, the log file is closed and re-opened. Refer to the Problem Determination chapter for more information

SNALOG_BUFLLEN

Set this variable to the maximum number of bytes per buffer to be logged. Refer to the Problem Determination chapter for more information

_DTF_USER_FLAGS

Set this variable to those values that you wish to include on the command line with the **dtfui**, **dtfiu**, **dtfpr**, **dtfrm**, and **dtfls** commands. If the value includes spaces, enclose the value in double quotes (“ ”).

Example:

Force each batch job to use an alternate alias file name and translate table:

```
# Use an alternate ALIAS file and Translate Table
```

```
export _DTF_USER_FLAGS="-A /etc/my_alias -K trans:n_xl"
#
```

NOTE

Changes made to the SYSTEM UNIX PROLOG file take effect starting with the next queued job defined. The changes do not modify already existing queued jobs.

4.3 System IBM Prolog File

The system IBM prolog file provides a way to customize each defined IBM initiated job. This file is copied into the IBM initiated job script when a job is initially defined. This file has no impact on queued or nonqueued UNIX initiated requests.

A single system wide system IBM prolog file, `/etc/ibm_dtf_defaults/system_prolog_unix`, is created during initial product installation. All DTF/DU-generated scripts use the Korn shell (`/usr/bin/ksh`). This file must conform to all Korn shell requirements.

The only capability supported for use in this file is to define the environmental variables described below. Use of this file for any other purpose is not supported.

NOTE

To modify the system IBM prolog file, log in as root or superuser and then execute the DTF/DU command **dtfjob sys_ibm**.

The supported environmental variables are:

`_DTF_USER_FLAGS`

Set this variable to any value you wish to include on the command line with the **dtfui** or **dtfiu** command. If the value includes spaces, bracket the value with “ ”.

`SNALOG_MASK`

Set this variable to select the type of trace records that should be included in the trace file. Refer to *Problem Determination* chapter for more information.

`SNALOG_FILE`

Set this variable to the file name to be used as a trace file. Unless overridden with a leading slash (/), tilde (~), or *variable name*, the file will be created in the current working directory of the executing batch job. Use **dtfjob** with the *dir* option to display the full path name of the current working directory.

`SNALOG_SIZE`

Set this to the maximum log file size allowed. The value is specified in 512 byte blocks. When the log file size is reached, the log file is closed and re-opened.

SNALOG_BUFLLEN

Set this variable to the maximum number of bytes per buffer to be logged. If not specified, all data in each message is traced.

Example:

Force each batch job to use an alternate alias file name and translate table:

```
# Use an alternate alias file and Translate Table
export _DTF_USER_FLAGS="-A /etc/my_alias -K trans:new_xlate"
#
```

4.4 User Preferences File

A user preferences file is automatically created for each DTF/DU user. The format of this file is identical to that of the system preferences file. Variables specified in this file override corresponding variables specified in the system preferences file.

The file name is `/var/spool/dtf/user_id/defaults/preferences`.

NOTE

To create or modify the USER PREFERENCES file, log in to the UNIX system and execute the DTF/DU command, **dtfjob preferences**.

4.5 User UNIX Prolog File

A user UNIX prolog file is automatically created for each DTF/DU user. The format of this file is identical to that of the system UNIX prolog file. Variables specified in this file override corresponding variables specified in the system UNIX prolog file.

The file name is `/var/spool/dtf/user_id/defaults/prolog_unix`.

NOTE

To modify the USER UNIX PROLOG file, log in with the desired UNIX userid and then execute the DTF/DU command **'dtfjob unix'**.

4.6 User IBM Prolog File

A user IBM prolog file is automatically created for each DTF/DU user. The format of this file is identical to that of the system IBM prolog file. Variables specified in this file override corresponding variables specified in the system IBM prolog file.

The file name is `/var/spool/df/user_id/defaults/prolog_ibm`.

NOTE

To modify the user UNIX prolog file, log in with the desired UNIX userid and then execute the DTF/DU command **'dtfjob ibm'**.

4.7 USER JOB Files

The DTF/DU product supports file transfer operations in three different modes: UNIX initiated interactive, UNIX initiated queued, and IBM initiated.

UNIX initiated interactive file transfers are invoked using the **dtfui** or **dtfiu** command (without the `-q` flag). The copy operation begins immediately. Status information is displayed on `STDERR`. Interactive file transfers are suitable for casual use, and for inclusion in user written scripts. These file transfers do not create or use any of the Job files.

UNIX initiated batch transfers are invoked using the **dtfui** or **dtfiu** command with the `-q` flag on the command line. Instead of executing immediately, a script file containing all of the relevant information is created. The script file is then executed as a separate process under the initiating UNIX user's ID.

IBM initiated file transfers are invoked by a user of an MVS system. Typically, TSO terminal users describe a transfer operation and then wait for the operation to complete. MVS initiated file transfers can also be invoked by batch jobs executing on the MVS system.

Both UNIX initiated batch transfers and MVS initiated transfers are implemented by creating a Korn shell script that contains a UNIX initiated interactive file transfer command. This Korn shell script, and associated files, are automatically assigned a five-digit DTF/DU job number. All of the files associated with a job number are created in a user job directory. The location of the jobs directory is `/var/spool/df/user_id/jobs/job_number`, where *user_id* and *job_number* are variable information.

The following files are created for each job:

- `checkpoint` Checkpoint data used when recovering.
- `corr_file` Used to store IBM specified correlation numbers. The contents are meaningful only to the MDU software.
- `data` Status information.
- `job_valid` Built as a result of a batch job or IBM initiated job being successfully created.
- `log` History file reflecting jobs execution history.
- `log.create` Contains errors messages that may have been generated while creating a script file for a batch job. The file usually remains empty.
- `pid.script` Used by DTF/DU to monitor job execution attempts.
- `script` Korn shell script containing UNIX initiated DTF/DU command.
- `socket` Created by transfers executing in the background. The socket is written to by dtfjob to issue stop commands.
- `statistics` Transfer summary information
- `text` User provided description of the job

In addition, the following files are created as a result of an MDU requester on the MVS system specifying a post-process step in its transfer request:

- `pp.script` Contains a Korn shell script used to invoke the post processing step.
- `pp.stdout` Contains the stdout output from the `pp.script` execution.
- `pp.stderr` Contains the stderr output from the `pp.script` execution.

4.8 Dispatcher Files

The DTF/DU dispatcher daemon is the UNIX based part of the product that receives IBM initiated transfer requests. The dispatcher validates the request, including UNIX userid and password, and then invokes the transfer request. Upon completion of the transfer, the dispatcher notifies the MVS system. DTF/DU dispatchers are controlled by means of the DTF/DU command, **dtfdsp**, described in a later chapter.

Several files are created for each DTF/DU dispatcher in directory `/var/tmp/df`. All files for a particular DTF/DU dispatcher start with the dispatcher *server_name* followed by a period. File names with brief descriptions are as follows:

server_name.lock

This is a UNIX lock file that is used to prevent multiple copies of a dispatcher using the same server name. When locked, a dispatcher is running and the file contains the process id of the dispatcher.

server_name.log

This is the current log file for the dispatcher.

server_name.log.TIMESTAMP

This is a log file that was created on the date and time indicated by the **TIMESTAMP** portion of the name. A dispatcher can be told to close its current log file and create a new file.

server_name.socket

This is a UNIX socket. It is used when the DTF/DU command **'dtfdsp'** directs a dispatcher process to start a new log file or to shut down.

NOTE

DTF/DU dispatcher log files can become quite large. Script *dsp_logs* in `/usr/ibm_dtf_examples` shows one automated method for limiting the size and/or number of dispatcher log files.

4.9 Translate Table Files

By default, translation between ASCII and EBCDIC is performed using internally coded tables. Alternate translation tables can be specified by using the `-K trans:table_name` parameter on a transfer command, or by including the parameter in a prolog file.

New translate tables can be created using the sample translate table as a model. All translate tables must be in `/etc/ibm_dtf_translate`. Instructions on translate table content and format are included in the sample translate table as comments.

4.10 Example Files

During installation, a directory containing examples is created. This directory is at `/etc/ibm_dtf_examples`. The two files that may be most useful are:

dtf_dsp

A sample Korn shell script that can be used to limit the size and number of DTF/DU dispatcher log files

example

An interactive Korn shell script that demonstrates the use of the UNIX initiated DTF/DU file transfer and file manipulation commands.

5. DTF/DU File Transfer Commands

The Data Transfer Facility Client for Digital UNIX (DTF/DU) communications oriented commands are as follows:

- **dtfui** copy file from UNIX system to MVS system
- **dtfiu** copy file from MVS system to UNIX system
- **dtfls** directory listing of files on MVS system
- **dtfrm** remove (delete) file on MVS system
- **dtfsb** submit an MVS file for batch processing on the MVS system
- **dtfpr** submit an MVS file for printing on the MVS system

The flags and parameters common to multiple commands are described in the next section. The rest of this chapter describes the individual commands.

5.1 DTF/DU Command Line Parameters

A variety of flags and parameters are supported for each command.

5.1.1 DTF/DU Flags

The following flags are supported for the indicated commands:

- **-adhimqr** dtfiu and dtfui
- **-f** dtfls and dtfrm
- **-l** dtfls
- **-v** all commands

The meaning of the individual flags is as follows:

-a or -r (-append or -replace)

These flags indicate the action to take when the destination file already exists. Append means add to the end of the existing file. If both *-a* and *-r* are specified, the replace option is used. These options are mutually exclusive with the *-C* option.

For UNIX to MVS file transfer (**dtfui**):

- ◆ Replace means create a new file (and remove the existing file after a successful copy). The previously existing file is not deleted until the replacement file is ready to take its place.

- ◆ If -a is specified and the MVS file does not exist, an error condition is reported and the data is not copied.
- ◆ If neither flag is specified, the MVS file must not exist when the transfer begins.

For IBM to UNIX file transfer (**dtfiu**):

- ◆ If either flag is specified and the UNIX file does not exist, a file is created and no error is reported.
- ◆ Both flags are ignored when the UNIX output is being directed to STDOUT.

The replace option (-r) is performed by truncating any existing file. This means that after an interrupted or failed transfer attempt, the original file is no longer available and the new data in the file is NOT a complete copy of the original file.

-h (-hold job)

Ignored except with -q option: Place batch job in HOLD status instead of READY status. Use dtfjob to manipulate or view files associated with the batch job.

-d or -i (transfer mode and translation)

Three modes of data transfer are available as follows:

text mode: no flag

EBCDIC/ASCII translate and NewLine insert/delete

data mode: -d

EBCDIC/ASCII translate but No NewLine insert/delete

image mode: -i

No translation and No NewLine insert/delete

By default, text mode, which is appropriate for ordinary files containing printable characters, is used. UNIX style NewLine characters are inserted or deleted at points that correspond to MVS file logical record boundaries. Because of the NewLine characters, the byte count of files on MVS systems and UNIX systems can differ. The number of records transferred should match.

Translation takes place as indicated by the translate option in the alias file entry. Table 5-1 lists these options and expected results. The *-d* flag indicates data mode. In this mode, NewLine characters are NOT inserted or deleted. Translation takes place as indicated by the translate option in the alias file entry, as shown in the table.

The *-i* flag indicates image mode. In this mode, NewLine characters are NOT inserted or deleted. Translation is inhibited and does NOT take place regardless of the translate option specified in the alias file entry.

Table 5-1 Data Transfer Modes and Translate Options

User choices		Expected Results	
Command Line Mode Choice	Alias File Translate Option	NewLine Insert/Deleted	Data Translated
default	none	yes	no
default	local	yes	yes
default	remote	yes	yes
<i>-d</i>	none	no	no
<i>-d</i>	local	no	yes
<i>-d</i>	remote	no	yes
<i>-i</i>	none	no	no
<i>-i</i>	local	no	no
<i>-i</i>	remote	no	no

-f (-force)

Inhibits display of error message and non-zero return code when specified file does not exist. Exception conditions other than "file does not exist" are as errors. This flag is valid only for commands **dtfls** and **dtfrm**.

-l (-long)

Display characteristics of file as well as file name. The information displayed on each line includes volume name, size of file in bytes, and creation date. This command is valid only for command **dtfls**.

-m (send mail at job completion)

Ignored except with *-q* option. When batch job completes, send a mail message containing a job statistics and job completion status.

-q (-queue as batch job)

Add job to a batch queue instead of performing the copy (that is, **dtfiu** and **dtfui**) in the foreground. Entries in the DTF/DU batch queue can be viewed and manipulated with the dtfjob command. When *-q* is specified, the *-m* (mail)

and *-h* (hold) options are also available. The default mail option is do not send mail on batch job completion. The default batch job status is READY.

-[v]v (-[very]verbose)

The *-v* option results in progress messages being printed to STDERR at SNA session startup and at each 1 megabyte interval during file transfer. The *-vv* option results in additional messages being printed during the SNA session startup process and for each nondata DEC Access Protocol Message.

5.1.2 Environmental Variables Referenced by DTF/DU

The following environmental variables are referenced by the DTF/DU file access commands:

DTF_USER_FLAGS

Set this variable to those values that you wish to include on the command line with the **dtfui** or **dtfiu** commands. If the value includes spaces, enclose the value with “ ”.

SNALOG_MASK

Set this variable to select the type of trace records that should be included in the trace file. Refer to the Problem Determination chapter for more information.

SNALOG_FILE

Set this variable to the file name to be used as a trace file. Unless overridden with a leading */*, *~*, or *variable name*, the file is created in the current working directory of the executing batch job. Use the **dtfjob** command with the *dir* option to display the full path name of a current working directory. Refer to the Problem Determination chapter for more information.

SNALOG_SIZE

Set this to the maximum log file size allowed. The value is specified in 512 byte blocks. When the log file size is reached, the log file is closed and re-opened. Refer to the Problem Determination chapter for more information.

SNALOG_BUFLLEN

Set this variable to the maximum number of bytes per buffer to be logged. If not specified, all data in each message is traced. Refer to the Problem Determination chapter for more information.

5.1.3 DTF/DU Parameters

The following parameters are supported for the indicated commands:

- *-C* dtfiu and dtfui only (check point interval)
- *-M* dtfiu and dtfui only (buffer size)

- -A and -I all commands (alias file and entry in file)
- -K all commands (MVS filename qualifier)
- -U and -P all commands (MVS system userid and password)

Meanings of the individual parameters are as follows:

-A Alias File Name

Name of file containing an entry for each available DEC/SNA gateway. The default value points to the alias file created when DTF for Digital UNIX was installed (/etc/ibm_dtf_alias).

-C checkpoint_interval

This parameter specifies the number of records between checkpoints. When a failed file transfer operation is restarted, the restart occurs by positioning both the UNIX and MVS files based on the most recent checkpoint information. Since checkpoint processing takes time, the frequency of checkpoints should be balanced against the impact on throughput.

The default value is 10000. The *-C checkpoint_interval* and the *-a* (append) options are mutually exclusive. The *-C checkpoint_interval* and the use "-" in place of UNIX filename are mutually exclusive. The *-C* option is ignored unless the *-q* option is also selected.

-I entry in alias file

Identifier of an individual record in the ALIAS file. Entry names are customer-defined when the ALIAS file is edited. The default for this value is the first record in the ALIAS file. This parameter is used to indicate an alias file entry to be used for the transfer session.

-K IBM_file_qualifier[:value]

A large number of MVS system oriented parameters can be specified. These keyword/parameter combinations are used to supply parameters needed by MDU to create or access MVS files. Several different keyword or keyword value parameters may be needed to perform a transfer. To specify multiple keywords, simply use the **-K keyword:keyvalue** sequence multiple times.

-M buffer_size

This parameter specifies the buffer size used to cache messages. The value must be at least as large as the largest record being transferred. The minimum value is 1024 and the maximum value is 32767. The default is 4096 and should be adequate for most transfers.

-P MVS_user_password

The MVS account password can be supplied as command line parameters. If not specified on the command line, the value of the environmental variable *_DTF_PASS_IBM* is used.

NOTE

The *-P* password value is the password associated with a userid on the MVS system. This is completely independent and unrelated to any password associated with an individual MVS file. The file related password is specify using the *-K password:value* syntax.

-U MVS_userid

The MVS account userid can be supplied as command line parameters. If not specified on the command line, the value of environmental variable `_DTF_USER_IBM` is used. This userid is typically identified as the Time Sharing Option (TSO) userid on the MVS system.

NOTE

Refer to Chapter 8, Data Security, for more information on MVS system security and security options.

5.1.4 DTF/DU and MVS File Names

MVS file names are translated to EBCDIC before use on the MVS system. By default, the names are also folded to upper case before use on the MVS system. The file names must follow the MVS naming conventions. Some special characters (@ \$ # ()) that are valid in MVS file names require special treatment when they appear on UNIX command lines. These characters should contained within quotes or preceded with an escape `\`.

NOTE

Specify *-K case* on the command line to inhibit folding lowercase letters to uppercase letters for the MVS file name..

Refer to Chapter 6, DTF/DU File Specifications, for more information.

5.1.5 DTF/DU and UNIX File Names

UNIX file names are case sensitive and may be specified as full or relative path names. If "-" is used in place of a UNIX file name, then UNIX STDIN or STDOUT is used as the source or destination of data.

5.2 dtfui - Copy UNIX File to MVS System

```
dtfui [ -adhimqrv ]  
      [ -A alias_file ]  
      [ -C checkpoint_interval ]  
      [ -I alias_file_entry ]  
      [ -K IBM_file_qualifiers ]
```

```
[ -M transfer_buffer_size ]  
[ -P IBM_pass ]  
[ -U IBM_userid ]  
{ UNIX_source_file | - } IBM_destination_file
```

If no parameters are provided, usage information is displayed. If insufficient or incorrect parameters are supplied, an error message is written to *STDERR* and a nonzero completion code is generated.

When *-* is used to indicate that the commands input data is to come from *STDIN*:

- ◆ *-q* is not supported
Queued jobs run in a separate context and do not have access to the current process *STDIN*. In this release, no attempt is made to spool or copy *STDIN* to a temporary file for later use.
- ◆ *-m* is not supported
Mail notification is valid only in conjunction with *-q*.
- ◆ *-h* is not supported
Hold option is valid only in conjunction with *-q*.
- ◆ *-C* is not supported
Checkpoint parameter is meaningful only with *-q* or with IBM initiated requests.

When a new file is created on the IBM mainframe and when a UNIX file name is specified as the source of input data, the UNIX file size is used to calculate the amount of space to allocate on the MVS system. When UNIX *STDIN* is used, the DTF/DU product cannot predict the size of the MVS file, so a default MVS file size is used. To explicitly specify the MVS file size, use *-K alloc:* or *-K balloc:* to specify primary allocation file size.

5.2.1 dtfui MVS File Qualifiers

alignment:*option*

Specifies TRK or CYL alignment.

allocation:*number-of-512-byte-blocks*

Specifies the primary allocation in 512 byte blocks.

ballocation:*number-of-bytes*

Specifies the primary allocation in bytes.

block_size:*blocksize*

Specifies block size for newly created files.

bsecondary:*number-of-bytes*

Specifies the secondary allocation in bytes.

case

Specifies whether the MVS file specification contains lowercase characters.

catalog:option

Specifies whether the data set you create is cataloged.

density:density

Specifies tape densities when writing a data set on tape.

directory_block:n

Specifies the number of directory blocks to allocate.

[no]hsmrecall

[Do not] Recall file from hierarchical storage.

label

Specifies the label formats for writing data sets to tape.

mrs:value

Specifies the maximum record size for the remote MVS file.

null:option

Specifies the action to take when a null record is to be written to an MVS file.

password:option

Specifies the MVS file's password.

pdse

Instructs the remote MVS system to create a PDSE data set rather than PDS.

post:post_processing_command

Specifies the UNIX command to be executed after successful completion of the copy operation. This parameter is valid only when the -q flag is specified.

recfm:value

Specifies fixed or variable length record format for the remote MVS file.

[no]release

Specifies the release of any unused tracks after creating an MVS file.

retention_period:n

Specifies the retention period in days before which a data set can be deleted.

secondary_allocation:number-of-512-bytes-blocks

Specifies secondary allocation quantity.

security_data:data

Data to forward to the security software executing in MDU.

sequence_number:number

For tape units only, the sequence number on the tape.

[no]single

Specifies blocked or nonblocked record format.

smsdclass:*data-class/*

smsmclass:*management-class/*

smsclass:*storage-class*

Specifies the SMS classes for a new MVS data set.

[no]spanned

Specifies whether variable-length records span physical blocks.

[no]supersede

Specifies creation of a new output file or, if the file already exists, replacing the existing file. If the SUPERSEDE qualifier is specified and the file transfer fails, the original MVS file is unaffected.

[no]translate

Specifies whether the data is to be translated. If translation is required then a default or customized translate table can be used for the translation process.

unit:*unit-spec*

Specifies a unit name which classifies the device type for a new data set.

volume:*(vol-name[,,,])*

Specifies a user-defined volume serial name(s).

[no]vsam

Specifies creation of [non] VSAM file.

5.2.2 dtfui EXAMPLES

For simplicity, the following examples assume that the default values for ALIAS file and ALIAS entry are appropriate; that a translate option or either local or remote is in effect; and that the IBM userid and password are supplied via environmental variables.

Copy UNIX Text File to MVS System with EBCDIC Translation

The output from the following command should be a copy of the UNIX node /etc/hosts file on the MVS system. The MVS file contains EBCDIC data and is created with default attributes. To control space allocation, record format, logical record size, and other attributes, use the *-K* qualifier.

```
dtfui /etc/hosts userid.hosts
```

Concatenate UNIX Text Files into a Single MVS File

In the following example, the **dtfui** command parameter of *-* means that the copy command input comes from UNIX STDIN.

```
cat /etc/hosts /etc/services /etc/fstab|\
dtfui - userid.several
```

Copy a tar File to an MVS System

The *-i* option is needed in the following example because tar files can contain data other than text characters. The resulting file on the MVS system is only useful as a source for tar data on a UNIX node. The command below shows how to extract a file named *x.c* from the IBM based tar file.

```
(cd mysource; tar cf - *.c )|\
dtfui -iv - ibm.mysource.tar
```

5.3 dtfui - Copy UNIX File from MVS System to Digital UNIX

```
dtfui [ -adhimqrv ]
      [ -A      alias_entry ]
      [ -C      checkpoint_interval ]
      [ -I      alias_file_entry ]
      [ -M      transfer_buffer_size ]
      [ -K      IBM_file_qualifiers ]
      [ -P      IBM_pass ]
      [ -U      IBM_userid ]
      IBM_source_file { UNIX_source_file | - }
```

If no parameters are provided, usage information is displayed. If insufficient or incorrect parameters are supplied, an error message is written to `STDERR` and a nonzero completion code is generated.

When `-` is used to indicate that the output data is to be written to `STDOUT`, the following applies:

-q is not supported

Queued jobs run in a separate context and do not have access to the current processes `STDIN`. In this release, no attempt is made to spool or copy `STDIN` to a temporary file for later use.

-m is not supported

Mail notification is valid only in conjunction with `-q`.

-h is not supported

Hold option is valid only in conjunction with `-q`.

-C is not supported

Checkpoint parameter is meaningful only with `-q`.

5.3.1 dtfiu MVS File Qualifiers

case

Specifies whether the MVS file specification contains lowercase characters.

[no]hsmrecall

The `HSMRECALL` qualifier indicates whether the MVS system should recall HSM archived data sets.

password:option

Specifies the MVS file password.

post:post_processing_command

Specifies a UNIX command that is to be executed after successful completion of the copy operation. This parameter is valid only when the `-q` flag is specified.

security_data:data

`SECURITY_DATA` is an IBM site-dependent qualifier.

[no]translate

Specifies if the data is to be translated. If translation is required then a default or a customized translate table can be used for translation.

unit:unit-spec

Specifies the unit name of the device the MVS file resides on.

volume:(vol-name[,,,])

Specifies a user-defined volume serial name(s). This keyword is required if the MVS file is not cataloged

5.3.2 Examples

For simplicity, the following examples assume that the default values for Alias file and Alias entry are appropriate, that a translate option or either local or remote is in effect, and that the IBM userid and password are supplied via environmental variables.

Display an MVS Partitioned Data Set Member on STDOUT

In the following example, the \ are needed to indicate that parentheses () are part of the command parameter instead of UNIX control indicators. The output from this command should be a copy of the IBM assembly language macro instruction OPEN. Translation from EBCDIC to ASCII and insertion of new line characters should have taken place.

```
dtfiu sys1.maclib\(\open\)
```

Concatenate an MVS File to an Existing UNIX File

```
echo "Here's a copy of the IBM OPEN macro"      >open_macro  
dtfiu -a sys1.maclib\(\open\)                  open_macro
```

or

```
echo "Here's a copy of the IBM OPEN macro"      >open_macro  
dtfiu sys1.maclib\(\open\)                    >open_macro
```

Copy MVS File in Image Mode (No Translate or New Line)

This command makes an exact copy of the MVS file. File content is not examined or translated. The IBM style record size information is discarded. The -v (verbose) flag causes progress messages to display as the file is being transferred.

```
dtfiu -iv backup.source.tar restore_my_source_tar
```

List Files in a tar Backup File on MVS

In the following example, **dtfiu** causes the entire file to be transferred from MVS to the UNIX system and written to STDOUT

```
dtfiu -i backup_source_tar - | tar tf -
```

5.4 dtfls Command - List Statistics for MVS File(s)

dtfls [-flv]

- [-A *alias_file*]
- [-I *alias_file_entry*]
- [-K *IBM_file_qualifiers*]
- [-P *IBM_pass*]
- [-U *IBM_userid*]
- { *IBM_filename* / *IBM_partial_filename* }

A line is generated for each filename (or partitioned data set member name) that matches the specification. If the MVS file does not exist and the *-f* option is specified, the command completes without an error message or other indicator.

The MVS file name may include wild card indicators. In this case, a line is written to STDOUT for each file that matches the specification. The rules for wild card indicators are as follows:

- Wild cards cannot be used within the high level qualifier (the portion of the file name to the left of the first period).
- A % symbol can be used to indicate *any single character*.
- An * can be used to indicate *any number of characters (including 0)*.
- Multiple wild cards can be used.
- Wild cards can be used within the member name portion of a file with PDS or PDSE organization.
- For MVS file wildcards to work consistently, the * must be preceded by a UNIX escape character, “\”. If this is not performed then the * will be parsed by UNIX and resolve to a local filename (if possible). If a local filename satisfies the wildcard file specification then it is that file specification that is sent to the MVS system and results will vary. Furthermore, if more than one local file specification satisfies the wildcard specification the following messages result.:

15:52:47 DUN0168W Extraneous parameters ignored after dtf.a

15:52:48 DUN7512E File Access, file not found.

The messages above appeared after a user entered: dtfls dtf.*. DUN0168W appears because more than a single file specification was handed to dtfls. DUN7512E appears because dtf.a, the first filespec found on the local disk, does not exist on the MVS system.

5.4.1 dtfls MVS File Qualifiers

case

Specifies whether the MVS file specification contains lowercase characters

[no]hsmrecall

The HSMRECALL qualifier indicates whether the MVS system should recall archived data sets.

security_data:data

SECURITY_DATA is an IBM site-dependent qualifier.

unit:unit-spec

Specifies the unit name of the device which classifies the MVS file resides on.

volume:(vol-name[,,,])

Specifies a volume serial name(s). This keyword is required if the MVS file is not cataloged.

5.4.2 dtfls Examples

For simplicity, the following examples assume that the default values for Alias file and Alias entry are appropriate; that a translate option or either local or remote is in effect; and that the IBM userid and password are supplied by means of environmental variables.

List Summary and All Information About File SYS1.MACLIB

To list summary and detailed information about SYS1.MACLIB, use the following command.

```
dtfls sys1.maclib
```

The expected output from this command should be a single line as follows:

```
SYS1.MACLIB
```

To list all available information about file SYS1.MACLIB

```
dtfls -l sys1.maclib
```

A typical output would be:

```
ESARS2 06DEC94 70315008 SYS1.MACLIB
```

Note that the resultant date is the file creation date.

List Information About A Group Of Files That Start with SYS1.P

The following example lists information about file groups using the * wildcard:

```
dtfls -l sys1.p\*
```

The output would normally resemble the following:

```
ESACAT 19FEB94 5137408 SYS1.PARMLIB
```

```
ESACAT 23MAR90 131072 SYS1.PPMACDEF
```

```
ESACAT 18JAN90 1340416 SYS1.PROCLIB
```

List SYS1.MACLIB Members That Start with OP

To list group members using the * wildcard, do the following:

```
dtfls sys1.maclib\ (op\*\)
```

The output would normally resemble:

```
SYS1.MACLIB(OPEN)
```

```
SYS1.MACLIB(OPNDST)
```

```
SYS1.MACLIB(OPNSEC)
```

Test for File Existence

The following example ignores lists of files and error messages and just checks completion code:

```
#!/usr/bin/ksh
dtfls joe 1>/dev/null 2>/dev/null
if [ $? -eq 0 ] ; then
echo "file joe exists"
fi
```

The following example suppresses error messages and completion code and just checks STDOUT:

```
#!/usr/bin/ksh
if [ `l -eq `dtfls -f joe | wc -l` ] ; then
echo "file joe exists"
fi
```

5.5 dtfpr Command - Print an MVS File

```
dtfpr [ -v ]
      [ -A alias_file ]
      [ -I alias_file_entry ]
      [ -K IBM_file_qualifier ]
      [ -P IBM_password ]
      [ -U IBM_userid ]
      IBM_filename
```

Using **dtfpr** with appropriate qualifiers, you can queue the specified MVS file for printing on MVS.

5.5.1 dtfpr MVS File Qualifiers

case

Specifies whether the MVS file specification contains lowercase characters.

class:*class-letter*

Specifies the output class value of the SYSOUT data set being printed.

[no]hsmrecall

Depending on form of qualifier selected, indicates whether or not the MVS system should recall archived data sets .

password:*option*

Specifies the MVS file password.

security_data:*data*

SECURITY_DATA is an IBM site-dependent qualifier.

unit:*unit-spec*

Specifies the unit name of the device the MVS file resides on.

volume:(*vol-name*[,,,])

Specifies a volume serial name(s) on which the MVS file resides. This keyword is required if the MVS file is not cataloged.

5.5.2 dtfpr Examples

For simplicity, this example assumes that the default values for Alias file and Alias entry are appropriate. The example also assumes that MVS userid and password information are supplied by means of environment variables or by MVS proxy facility.

In the following example, **dtfpr** prints a member JES2 from an IBM PDS file called SYS1.PROCLIB to class X output.

```
dtfpr -K class:X SYS1.PROCLIB\ (JES2\)
```

5.6 dtfrm - Remove (Deletes) an MVS File

```
dtfrm [ -vv ]  
      [ -A alias_file ]  
      [ -I alias_file_entry ]  
      [ -K IBM_file_qualifier ]  
      [ -P IBM_password ]  
      [ -U IBM_userid ]  
      IBM_filename
```

Deletes a file from MVS.

5.6.1 dtfrm MVS File Qualifiers

case

Specifies whether the MVS file specification contains lowercase characters.

[no]hsmrecall

Depending on form of qualifier selected, indicates whether or not the MVS system should recall archived data sets.

password:option

Specifies the MVS file password.

security_data:data

SECURITY_DATA is an IBM site-dependent qualifier.

unit:unit-spec

Specifies the unit name of the device the MVS file resides on.

volume:(*vol-name*[,,,])

Specifies volume serial name. This keyword is required if the MVS file is not cataloged.

5.6.2 dtfrm Examples

For simplicity, the following examples assume that default values for ALIAS file and ALIAS entry are appropriate, and that IBM userid and password information are supplied by means of environment variables or by MVS proxy facility.

Remove or Delete an MVS File

```
dtfrm user23.temp1
```

No output to STDOUT or STDERR is generated when successful.

Error Output when IBM File Does Not Exist

```
dtfrm user23.temp1
```

Output when file does not exist on MVS:

```
13:12:56 DUN7512E File Open, file not found.
```

Attempt to Delete a File and Suppress Error Messages

```
dtfrm -f user23.temp1
if [ $? -eq 0 ] ; then
echo "File deleted"
fi
```

5.7 dtfsb Submit MVS File for Execution

```
dtfsb [ -vv ]
      [ -A alias_file ]
      [ -Ialias_file_entry ]
      [ -KIBM_file_qualifiers ]
      [ -P IBM_pass ]
      [ -U IBM_userid ]
      IBM_filename
```

Use **dtfsb** to submit an MVS file for printing on MVS.

5.7.1 dtfsb MVS File Qualifiers

case

Specifies whether the MVS file specification contains lowercase characters.

class:*class-name*

Specifies the value of the SYSOUT data set being printed.

[no]hsmrecall

Depending on form of qualifier selected, indicates whether or not the MVS system should recall archived data sets.

password:*option*

Specifies the MVS file password.

security_data:*data*

SECURITY_DATA is an IBM site-dependent qualifier.

unit:*unit-spec*

Specifies the unit name of the device the MVS file resides on.

volume:*(vol-name[,,,])*

Specifies a volume serial name. This keyword is required if the MVS file is not cataloged.

5.7.2 dtfsb Example

For simplicity, this example assumes that the default values for Alias file and Alias entry are appropriate. The example also assumes that IBM userid and password information are supplied via environment variables or by MVS proxy facility.

The following example submits a member, JOB33, from an MVS PDS file called USER1.JOBLIB to an MVS system printer using class job class A.

```
dtfsb -vv -K class:A USER1.JOBLIB\JOB33\)
```

6. DTF/DU File Specifications

This chapter covers the Data Transfer Facility Client for Digital UNIX (DTF/DU) file specification syntax for remote file names. On Digital UNIX clients, you substitute an IBM/MVS file specification for one of the file names in any DTF/DU-supported file transfer command.

6.1 Specifying Remote MVS Files for DTF/DU File Access

To specify a remote MVS file to DTF/DU, use the following syntax:

```
-K qualifier -K qualifier2 ibm_file_name
```

where

```
ibm-filename
```

specifies an MVS 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 one to eight 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 on tape restrictions.

Many IBM sites require you to use your userid as the first segment in a qualified file name. This is not required by MDU, 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 one to eight alphanumeric characters long, and must begin with an alphabetic character.

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 member DATA in the partitioned data set JONES.TEST.FILE.

```
ibm-file-spec-qualifiers
```

can be appended after a file name when using DTF. Most MVS file specification qualifiers can be associated with defaults that are set up during installation of the MDU software. These defaults are often modified, so you may need to contact an MVS system programmer for the most current MVS file specification qualifier defaults.

Refer to the IBM JCL Reference manual for a complete description of MVS file names.

6.2 MVS File Specification Optional Flags

Use the MVS file optional flags (also referred to as file specification qualifiers) described in this section to supply attributes that are unique to MVS files. Precede each qualifier with *-K*.

When you specify MVS file specification flags with a value, you must use a colon (:) to separate the qualifier from the value. Multiple qualifiers may be specified for one command. Each qualifier should be preceded by a *-K*.

All of the qualifier information should be specified after the command name and before the IBM and UNIX file names.

6.2.1 Optional Flags Defaults

Most MVS file specification optional flags have a default value. You select the default value by omitting the MVS file specification flag. The default value is always controlled by the MDU software on MVS. In some cases, the default value can be defined when installing the MDU software.

Depending on whether or not the default value can be changed, the default value for an MVS 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 MDU is installed:** You should contact your IBM site for the most current value of the qualifier's default.

6.2.2 Qualifier Restrictions When Accessing MVS files

Table 6-1 lists the MDU qualifiers and MVS file types for which each qualifier can be used. Applicable file types are non-VSAM disk-resident, tape-resident, and VSAM.

Table 6-1 MVS File Specification Qualifiers

MDU Qualifiers	Non-VSAM	Tape-Resident	VSAM Files
alignment ³	supported	ignored	ignored
allocation	supported	ignored	supported
ballocation	supported	ignored	supported
block_size ³	supported	supported	ignored
bsecondary	supported	ignored	supported
case	supported	supported	rejected
catalog ³	supported	supported	ignored
class	ignored	ignored	ignored
density ^{1,2,3,6}	ignored	supported	ignored
directory_blocks ³	supported	ignored	ignored
hsmrecall ^{1,2,4,5}	supported	supported	supported
label ^{1,2,3,6}	ignored	supported	ignored
null	supported	supported	ignored
password ^{1,2,3,4,5}	supported	supported	supported
pdse	supported	ignored	ignored
post	supported	supported	supported
recfm	supported	supported	supported
release ³	supported	ignored	ignored
retention_period ³	supported	supported	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

MDU Qualifiers	Non-VSAM	Tape-Resident	VSAM Files
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
volume ^{1,2,3,4,5}	supported	supported	supported
vsam ³	ignored	ignored	sequential files only

¹ Used when reading MVS input files.

² Used when writing over existing IBM output files.

³ Used when creating a new MVS file.

⁴ Used when deleting an MVS file.

⁵ Used when listing an MVS file.

⁶ Used only for IBM tape-resident files

6.2.3 Qualifier Descriptions

This section describes the MVS file specification qualifiers used when accessing files on MVS systems. The qualifiers are described in alphabetical order. All qualifiers can be abbreviated as long as the abbreviation remains unique.

alignment: *option*

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 MDU should supply a default. The MDU default is specified at the IBM site when MDU 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. For most applications, use this option 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 expressed in 512 byte blocks. Use **ballocation** to specify the size in bytes.

Usage Notes:

- ◆ This parameter should be used only for those instances when piping input to the **dtfui** command. This is because DTF/DU cannot inform MDU what the size of the input file is when data is input from a pipe.
- ◆ Use of this qualifier with **dtfui** could cause the file to be over allocated or under allocated. This results in a space error file exceeded allocation.
- ◆ This qualifier is useful when backing up UNIX files to MVS systems (using tar with a pipe to **dtfui**).

ballocation: number-of-bytes

Same as **allocation** except value is expressed in bytes instead of blocks.

block_size:blocksize

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

Usage Notes:

- ◆ This qualifier is valid only when creating a file.
- ◆ By default, MDU creates a blocked file with the block size set to the default value specified by MDU.
- ◆ The MDU default is recommended for this qualifier.
- ◆ The SINGLE qualifier overrides **block_size** and creates an unblocked file.
- ◆ If you specify a block size, you must adhere to 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 minimum block size of 8.

bsecondary_allocation:number-of-bytes

Same as SECONDARY_ALLOCATION except value is expressed in bytes instead of blocks.

case

Specifies whether the MVS file specification contains lowercase characters.

Usage Notes:

- ◆ If the MVS 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.
- ◆ Wild card directories that return file specifications with lowercase characters will have a CASE qualifier returned with those file specifications.
- ◆ The MDU system installer must specify that NODEPARMS LOWERCASECREATE is a supported feature.
- ◆ If the IBM file is SMS managed, then lowercase file specifications are unsupported.

catalog:option

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 MDU should supply a default. The MDU default is specified at the IBM site when MDU 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:value

For print requests (dtfpr), class specifies the JES SYSOUT class for the output file on MVS.

For submit requests (dtfsb), class specifies the JES initiator class for job submission.

density:density

Specifies one of the following densities when reading or writing files to tape:

- ◆ 800

- ◆ 1600
- ◆ 6250

Usage Notes:

- ◆ Default is to select the highest density allowed by the device.
- ◆ This qualifier is ignored with disk devices.

directory_blocks:n

Specifies a number from 1 to 999999 to define how many 256-byte directory blocks are allocated to a partitioned data set PDS directory when creating a new PDS.

Usage Notes:

- ◆ This qualifier is valid only for creating a PDS.
- ◆ If you do not specify a value, MDU supplies the IBM site default.
- ◆ 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 changed.
- ◆ The ratio of member names to directory block varies depending on the MVS applications use of the directory. Generally between 5 and 20 PDS members can share single directory block.

[no]hsmrecall

Depending on the form used, this qualifier indicates if the MVS system should recall an archived file if the file is not in system 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 your request to hang, waiting for a file to be recalled. The recall process may require a tape to be mounted.
- ◆ **hsmrecall** can be disabled when MDU is installed and configured. Check with your MVS support group if you receive indications that HSM archived files are not supported by MDU.

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 - Nonlabeled
- ◆ SL - Standard label

Usage Notes:

- ◆ This qualifier is ignored with disk devices.
- ◆ The default for this qualifier is label:SL.
- ◆ Only SL or AL can be specified with an MVS tape input file.
- ◆ You can specify AL, BLP, NL, or SL with an MVS tape output file. BLP and NL tape files can not be used in recovery operation.

null:option

Specifies the action to take when a null record is to be written to an MVS file.

You can choose from the following options:

- ◆ %Cchar specifies that the indicated character should be stored in place of the null record.
- ◆ %Dnumber specifies that the indicated decimal number should be stored in place of the null record. The number must be between -128 and +127 inclusive.
- ◆ %Xnumber specifies that the indicated hexadecimal number should be stored in place of the null record. The number must be between 0 and FF inclusive.
- ◆ %Onumber 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 specifies no special processing occurs. 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 UNIX to IBM transfers. It is ignored when copying files from IBM to UNIX.

password:password

Specifies a 1- to 8-character password for an existing MVS file.

pdse

Instructs the remote MVS system to create a PDSE data set rather than a PDS

[no]release

Specifies the release of any unused tracks after creating an MVS data set. The unused space is released to the smallest allocation space unit (TRK or CYL) used to create the file.

Usage Notes:

- ◆ This qualifier is valid only for creating or modifying a file.
- ◆ The default qualifier is **norelease**.

secondary_allocation:number-of-512-byte-blocks

Specifies or overrides the defined default extension quantity for the input file.

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 MDU. Check with the MVS 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**.
- ◆ 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*

smsmclass:*management-class*

smsclass:*storage-class*

These qualifiers specify the class type used during storage allocation at file creation time. The three classes are: **smsdclass**, data class; **smsmclass**, management class; and **smsclass**, storage class.

Usage Notes:

- ◆ These qualifiers are valid only for creating a file.
- ◆ If you do not specify values for these qualifiers, MDU supplies the defaults specified at the IBM site.
- ◆ Prior to allocating storage MDU 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 MVS system programmer can specify values that will override any values you may specify. Check with the MVS system programmer for more information.
- ◆ Access to SMS files can be disabled when MDU 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.
- ◆ Fixed-block standard files DSORG=FBS cannot be appended to.

- ◆ 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:table_name

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 table name with the **translate** qualifier, you can specify a translation table. If you specify the **nottranslate** qualifier, data translation does not occur.

Usage Notes:

- ◆ The default qualifier is **translate**.
- ◆ Refer to the DTF/DU installation section for information on customizing an installation specific translate table.

unit:unit-spec

Specifies a unit from one to eight alphanumeric characters long to classify devices for creating a data set or accessing an uncataloged data set.

Usage Note:

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

userid:userid

Specifies a userid of one to eight alphanumeric characters to verify access through the appropriate IBM security system.

Usage Notes:

- ◆ The default is not to supply a userid.
- ◆ 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 the chapter concerning security for more information.

volume:vol-name[,...]

Specifies a volume serial name of 1 to 6 alphanumeric characters for creating an MVS file or reading an uncataloged MVS file. You can include a total of 252 alphanumeric characters. The parentheses are required if more than one volume serial name is supplied, with each vol-name separated by commas.

Usage Notes:

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

[no]vsam

Specifies whether an MVS file is a VSAM file.

Usage Notes:

- ◆ 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**.
- ◆ This qualifier is supported only if it is indicated during MDU installation time that VSAM file creation is supported.

7. Supported MVS File Types

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

7.1 Supported MVS Input File Types

This section describes considerations that you should be aware of before you use MVS files as input to a **dtfui** copy command.

7.1.1 Supported File Organizations

MDU supports the following MVS file organizations as input:

- **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.
- **VSAM** - This group of files includes Key Sequenced Data Set (KSDS), and Entry Sequenced Data Set (ESDS) file organizations. Relative Record Data Sets (RRDS) are not supported.

7.1.2 Supported Record Formats

MDU supports input files with fixed and variable length records.

7.1.3 Supported Record Lengths

MDU supports input files with a record length of up to 32760 bytes.

7.1.4 Supported Record Attributes

MDU supports the following combinations of record attributes for MVS input files:

- Any combination of the supported record formats with or without blocking, including fixed blocked standard (FBS) and variable blocked spanned (VBS).

7.2 Supported MVS 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/DU transfers.

7.2.1 Supported File Organizations

MDU supports the following MVS file organizations as output:

- **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.
- **VSAM ESDS** - Records can be added to the end of an ESDS files. The UNIX-to-IBM copy command should include *append* or *-a*. The **DTF/DU** commands **dtfpr** (print) and **dtfsb** (submit) do not support VSAM ESDS files.
- **VSAM KSDS** - Records can be inserted into an existing KSDS data set. The UNIX-to-IBM copy command should include *append* or *-a*. The **DTF/DU** commands **dtfpr** (print) and **dtfsb** (submit) do not support VSAM ESDS files.

7.2.2 Supported Record Formats

For *replace*, *append*, and *creation* of new files, fixed and variable length records are supported. Blocking and spanned (with variable length records) are also supported.

7.2.3 Supported Record Lengths

MDU supports output files with a record length of up to 32760 bytes. The MVS file system requires, at creation time, the maximum permitted record length be specified prior to creating a file, so MDU must provide a default for this when it creates an output file.

MVS output files with variable record lengths require record lengths 4 bytes greater than the maximum record length. This difference is due to MVS file system requirements. Conversely, output files on Digital UNIX nodes have record lengths 4 bytes less than that of MVS input files.

MDU determines maximum record length as follows:

1. Round the longest record in file (LRECL) attribute up to the next highest multiple of 256.
2. Subtract 1.
3. Add the 4 byte requirement of the MVS file system, if record format is variable.

The default maximum record length for MVS output files is usually 255 (259 if the 4 overhead bytes are included). Use *-K mrs:number* to explicitly select a size.

7.2.4 Additional File Attributes

The BLOCK_SIZE and SINGLE option flags control the blocking factor of an IBM output file. Blocksize will have an affect on the physical size of the MVS file. In general, it is not efficient to use the SINGLE option flag as that creates unblocked MVS files.

7.2.5 Files with ANSI Control Characters

When files with ANSI control characters are copied to UNIX, the ANSI control characters are NOT converted in any way. They are simply included in the UNIX file as data. See the man page on the Digital UNIX command **asa** for information on processing files with ANSI control characters

7.3 File Transfer Considerations

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

7.3.1 VSAM File Transfer Restrictions

The following restrictions apply to VSAM file transfers:

- If a transfer request involves VSAM files, the transfer must execute in non-recoverable mode only.
- If a transfer request involves an ESDS file, you cannot overlay the existing file.

7.3.2 Transferring MVS Tape-Resident Files

MDU does not support IBM input tapes that have a label type of NL.

The following restrictions apply if you select recovery when reading and writing IBM tape-resident files:

- The label type must be SL or AL.
- 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.

MDU supports tapes that have a label type of AL for both input and output. MDU 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. MDU interfaces permit longer data set names. MVS data set names on tape, however, are truncated to 17 characters. If you specify more than 17 characters when accessing a tape-resident file, the name is truncated to the LEFT, resulting in the 17 RIGHTMOST characters being used as the file name.

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. MDU will issue a mount requested message on the MVS console indicating a request to mount a disk not a tape.

7.3.3 Transferring Files with Horizontal Tab Characters

MDU does not expand tab characters into repeated spaces. Instead, a tab character is treated like an ordinary letter, number, or space character. If translation between EBCDIC and ASCII takes place, the tab character is translated as well.

8. Data Security

This chapter covers the data security measures provided by the operating systems that are supported by MDU. 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.

8.1 Data Security Rules

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

- When the Digital UNIX user attempts to access (or create) a MVS data file, the local (that is, Digital UNIX) file system is used to validate that the Digital UNIX user is authorized to access (or create) the UNIX file. Similarly, MDU is responsible for verifying the right to access (or create) the MVS file.
- When an MVS user attempts to access a Digital UNIX file, the Digital UNIX file system determines if the Digital UNIX userid (supplied in the transfer request submitted on the MVS system) is permitted to access the file.

8.2 Using Userids and Passwords

The DTF/DU software sends userid and password, if specified, to the remote system. It is the responsibility of the components (e.g. DTF/security system/file system) executing on the remote system to check if access is allowed to the remote file. The rules for the userid and password depend on the remote operating system.

For MVS initiated requests, the UNIX userid and passwords can be sent. Maximum lengths are:

- UNIX userid 7 bytes
- UNIX password 32 bytes

For UNIX initiated requests, the UNIX node name and UNIX userid are always sent to MVS. If supplied by the UNIX user, MVS userid and password are also sent. The maximum lengths are as follows:

- UNIX node name 8 bytes
- UNIX password 32 bytes
- MVS userid 8 bytes
- MVS password 8 bytes

8.3 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. Userids and passwords are used when proxy accounts are not defined. In addition, if default proxy userids are implemented, you do not need to specify the remote userids. For information on recommended use, see your system administrator or the person in charge of the proxy database.

8.4 Proxy Access to the MVS System

The MDU proxy mechanism passes the user's Digital UNIX node name and account directly to the security system that is installed on the MVS system. The MVS security system verifies that a proxy entry has been made for that user at that node and grants access based on rules set up by the MVS security administrator. The MVS security administrator implements security based on predetermined rules.

8.5 Proxy Access to the Digital UNIX System

Proxy access has not been implemented in this version of the software. MDU initiated requests must specify Digital UNIX userid and password information in order to perform a transfer between the two systems.

9. Recoverable Copies

A recoverable copy is a copy request that can be restarted from a checkpoint after a transfer interruption.

For UNIX initiated queued transfers, the value associated with the `-C` command line parameter is used to differentiate recoverable and non-recoverable copies. If a non-zero value is specified or allowed to default, then the transfer operation is recoverable. If a zero value is specified, then no checkpoints are taken and no recovery is attempted.

UNIX initiated nonqueued transfers (that is, without the `-q` flag) are not recoverable.

For information on how to specify IBM initiated recoverable copies, see the use guide for the IBM based component of this product.

9.1 Recoverable Copy Checkpoints

Recoverable copies always restart a transfer at the latest checkpoint. The beginning of a file is a defacto checkpoint. Additional checkpoints are taken at fixed intervals as a recoverable copy proceeds.

The frequency of checkpoints can have a significant impact on throughput. Frequent checkpoints minimize the amount of data that is retransmitted when restarting. But taking a checkpoint also takes time that would otherwise be spent transmitting data. So, frequency of checkpoints must be balanced against the performance impact of taking them.

9.2 Status Display of Recoverable Copies

The status of an IBM initiated or a UNIX initiated recoverable copy job can be displayed with the DTF/DU command 'dtfjob list'.

NOTE

For MDU initiated jobs, the DTF/DU JOB_NUMBER and the MDU RETRY NUMBER match.

See the next chapter for more information about using the DTF/DU **dtfjob** command.

9.3 Recovering an Interrupted Transfer

When a recoverable copy is interrupted, DTF/DU changes the status of the job to either “failed” or “held”. Jobs are placed in the “failed” state when the nature of the error suggests that re-trying the job will probably not be successful, or if the limit on the number of retries has been reached. Jobs in the “failed” state can not be re-tried or re-started.

Jobs are placed in the “held” state when the reason for the failure appears to be transient, or after node restart when a recoverable transfer was in progress.

9.3.1 MVS Initiated Transfers

Recovery of an interrupted MDU initiated job is initiated by the MVS user. See the MDU users guide for more information.

9.3.2 UNIX Initiated Queued Jobs

When a UNIX initiated job is placed in hold state, a minimum time-to-wait-before-retry is calculated. In general, the time to wait increase with the number of retries. When this minimum time to wait has passed, the job becomes eligible to be automatically restarted. When the number of UNIX initiated batch jobs for the user fall below the user specified (or default) limit, the job will be retried. No user action is required.

If a queued transfer is interrupted and DTF/DU determines that the interruption is not fatal then the transfer may be continued. The state of the transfer job has changed from EXECUTING to SYSTEM_HOLD if DTF/DU determines the transfer should be continued.

To force an immediate retry, use the following DTF/DU command:

```
dtfjob exec JOBNO
```

10. dtfjob - Display and Control Queued Jobs

The **dtfjob** command is the only supported method to display and control queued (or batch) jobs. The command is primarily intended to manage DTF/DU initiated queued jobs. The **dtfjob** command also displays (but does not control) information about IBM initiated jobs. Queued jobs are created with a dtfui or dtfiu command and the -q option. MVS initiated jobs are jobs created via the MVS MDU copy facility.

The two formats of the dtfjob command are:

- dtfjob list
- dtfjob *action jobid*

10.1 dtfjob Description

The parameter following the command indicates the action required. Actions can be divided into two categories: actions that affect an individual job and actions that affect multiple jobs.

10.1.1 dtfjob Actions Associated with a Single Job

Actions that affect an individual job require the job's number as the second parameter. Leading zeros need not be typed. Some of the actions also require a third parameter. Actions are as follows:

delete

remove all files associated with the specified job.

dir

display name of directory containing all files associated with job.

execute

"run" the job by executing the job's script file.

log

display the contents of the job's log file.

priority

change the jobs scheduling priority. This command requires that the new priority be supplied following the job number. The priority value range is 01 (best) through 99 (worst). Jobs start with a priority of 25.

script

invoke a text editor to edit the script file.

set

replace the job's description with the information supplied following the job number. The description can be as long as the user wants, but only the first few bytes (~20) are shown with the list and show options of dtfjob. The dtfjob command with the text option can be used to display the entire job description. When initially created, the job description simply states that the job is a batch or IBM initiated job.

show

show status information about the job. The information displayed is described below under the heading "dtfjob list/show output".

stop

if a job is running, terminate the transfer operation. If a job is in "hold" status, change it to "stopped" status. While stopped, a job cannot be automatically restarted by a background cron process. Only batch jobs can be stopped. MDU initiated jobs are controlled from MVS.

text

show the job's description as specified by the set command.

10.1.2 dtfjob Actions Associated with Multiple Jobs

list

display a list of current user jobs and their status. This action is similar to the show option, except this action displays information for all jobs. Jobs are displayed based on job status modification time with oldest jobs first.

cronlog

display cron log file. The content on the cron log file indicates the time that the dtf cron process last executed and the actions taken during that execution. This information is generally of interest only when batch jobs do not seem to be retried automatically.

preferences

invoke a text editor to edit user's preferences file. Instructions for editing the file are shown as comments in the file itself.

sys_pref

invoke a text editor to edit systemwide preferences file. The contents of this file are overridden by any values specified in the user's preferences file.

ibm

invoke a text editor to edit user's IBM initiated job prolog file. The content of this file is included in each IBM initiated job script file.

sys_ibm

invoke a text editor to edit the systemwide IBM initiated job prolog file. The

content of this file is included in each IBM initiated job script file BEFORE the user's IBM prolog file. This allows a user's prolog file to contain overrides to the systemwide file. You must be su to invoke this function.

UNIX

invoke a text editor to edit user's UNIX initiated (or batch or queued) job prolog file. The content of this file is included in each UNIX initiated job script file.

sys_unix

invoke a text editor to edit the systemwide UNIX initiated job prolog file. The content of this file is included in each UNIX initiated job script file BEFORE the user's UNIX prolog file. This allows a user's prolog file to override the systemwide file. You must be su to invoke this function.

10.1.3 Output of 'list' and 'show' Action

Both the list and show actions result in a header line followed by one row of data for each job. The list action displays all jobs for the current user while the show job displays information about a single job.

JOB ID

Job IDs are assigned automatically when jobs are created. For MDU initiated jobs, the DTF/DU JOB ID matches the MDU RETRY NUMBER.

TYPE

IBM or UNIX identifies the job as an MDU initiated job or a batch job.

STATUS

indicates the jobs current status. see the section below for a description of job status values and transitions among status values.

TRIES

contains two values separated by a "/". The first value is the number of times that execution of the job has been attempted. The second number is the limit on the number of times that a job can be retried.

DELAY

contains "N.A." for not applicable or the number of minutes to wait before a retry of the job will be attempted. A value other than N.A. will only appear for UNIX initiated jobs that have been interrupted.

DESCRIPTION

contains the 'text' information provided via the dtfjob set action.

NOTE

To list all jobs in order by JOB ID, pipe the output through the UNIX sort command **dtfjob list | sort.**

10.1.4 User Job States

Each IBM initiated and UNIX initiated queued job is always in a specific state. The states and state transitions are:

USER_HOLD

The **dtfui** or **dtfiu** command that created the job included the *-q* and *-h* flags. The reasons for defining a job but *holding* it include: modifying the job script file before running job, or explicitly schedule the job at a later time. IBM initiated jobs are never in this state. Jobs leave this state by one of the actions listed in Table 10-1:

Table 10-1 User_Hold Job Actions

Action	New State	Comment
dtfjob delete	-	removes job files
dtfjob execute	EXECUTING	

EXECUTING

Jobs in this state have had (or are about to have) the script file from the job directory executed in the background. In this state, data is actually being transferred between the Digital UNIX system and MVS. Jobs leave this state by one of the actions in Table 10-2:

Table 10-2 Executing State Actions

Action	New State	Comment
interrupt	SYS_HOLD	recoverable error
system re-boot	SYS_HOLD	recoverable error
MDU request*	SYS_HOLD	MDU request
failure	FAILED	permanent error
success	COMPLETED	transfer completed
dtfjob stop**	STOPPED	operator action

* MVS initiated jobs only

** UNIX initiated jobs only. Only UNIX-initiated jobs can be stopped by UNIX operator action.

SYS_HOLD

Jobs in this state were executing when a recoverable error was detected. After a suitable delay, the job will be retried. Jobs leave this state by one of the actions listed in Table 10-3:

Table 10-3 Sys_Hold State Exit Action

Action	New State	Comment
dtfjob delete	-	removes all job files
MDU request(*)	EXECUTING	MDU request
dtfjob execute(**)	EXECUTING	
dtfcron(**)	EXECUTING	after time interval

* MVS initiated jobs only

** UNIX initiated jobs only

STOPPED

Jobs in this state were executing but have been interrupted at the users request. Jobs leave this state by one of the actions listed in Table 10-4:

Table 10-4 Stopped State Exit Action

Action	New State	Comment
dtfjob delete	-	removes all job files
dtfjob execute	EXECUTING	

FAILED

Jobs in this state were executing when a permanent error was encountered.

The job cannot be retried. Jobs leave this state by one of the actions shown in Table 10-5:

Table 10-5 Failed State Exit Action

Action	New State	Comment
dtfjob delete	-	removes all job files

COMPLETED

Jobs in this state ran to normal completion. All data was transferred. Post processing, if specified, was attempted and may or may not have completed successfully. Jobs leave this state by means of the action listed in Table 10-6:

Table 10-6 Completed State Exit Action

Action	New State	Comment
dtfjob delete	-	removes all job files

The **dtfjob** command is the only supported method for users to examine and manipulate jobs in their `/var/spool/df` directory tree.

10.2 dtfjob Examples

For simplicity, the following examples assume that: the default values for Alias file and Alias entry are appropriate; that a translate option or either local or remote

is in effect; and that the MVS userid and password are supplied through environmental variables. In these examples, the assumed UNIX userid is *guest*.

Show List of All Batch and IBM Initiated Jobs

```
dtfjob list
```

Job	Type	Priority	Status	Tries	Delay	Description
00001	IBM	25	Completed	1/10	N/A	IBM initiated
00002	UNIX	25	Executing	1/10	N/A	UNIX initiated
00003	UNIX	25	User_Hold	0/10	N/A	UNIX initiated

This display shows three jobs: an IBM initiated job, and two UNIX initiated queued jobs

Show Information About an Individual Job

```
dtfjob show 2
```

Job	Type	Priority	Status	Tries	Delay	Description
00002	UNIX	25	Executing	1/10	N/A	UNIX initiated

Change a Job Description, Then Display All Jobs

```
dtfjob set 3 Demo - copy /etc/fstab  
dtfjob list
```

Job	Type	Priority	Status	Tries	Delay	Description
00001	IBM	25	Completed	1/10	N/A	IBM initiated
00002	UNIX	25	Executing	1/10	N/A	UNIX initiated
00003	UNIX	25	User_Hold	0/10	N/A	Demo - copy etc/fstab

Change a Job Status and Verify Change in Status

```
dtfjob exec 3  
dtfjob list
```

Job	Type	Priority	Status	Tries	Delay	Description
00001	IBM	25	Completed	1/10	N/A	IBM initiated
00002	UNIX	25	Executing	1/10	N/A	UNIX initiated
00003	UNIX	25	Executing	1/10	N/A	Demo - copy etc/fstab

Delete a Job

```
dtfjob delete 1  
dtfjob list
```

Job	Type	Priority	Status	Tries	Delay	Description
00002	UNIX	25	Executing	1/10	N/A	UNIX initiated
00003	UNIX	25	Executing	1/10	N/A	Demo - copy etc/fstab

11. Administrative Commands

The DTF/DU includes two additional commands for use by the DTF/DU or system administrators. These commands are as follows:

- **dtfboot** change state of DTF/DU jobs that were running when node rebooted
- **dtfdsp** control dispatcher

11.1 Administrative Command: dtfboot

dtfboot { start | reset }

The **dtfboot** command is executed at system boot. For each DTF/DU user defined in `/var/spool/DTF`, this command automatically sets an indicator in a directory associated with that user. It also resets any “locked” indicator to allow appropriate access to each user DTF/DU jobs directory. Because this command, when invoked with the ‘start’ parameter, modifies files for all DTF/DU users, you should run this process as root or super user.

This command must execute before any other DTF/DU command. Since DTF/DU utilizes the Digital UNIX cron facility, **dtfboot** must execute before Digital UNIX starts the cron daemon.

During DTF/DU product installation, an entry named “S57b4_dtfcron” is added to the `/sbin/rc3.d` directory. The “S57b4” prefix ensures that the entry is processed before S57cron, which starts Digital UNIX cron processing. The S57b4_dtfcron points to an entry in `/sbin/init.d` which in turn points to the **dtfboot** command in `/usr/opt/DTF100/bin/dtfboot`.

Under normal conditions, there should be no need to execute this command manually. As a debug aid, Digital may request that you execute this command to “reset” a given user’s job status information. To do this, login as the user in question and run the command:

```
/usr/bin/dtfboot reset
```

The environmental variable *USER* selects the DTF/DU user files that are reset.

NOTE

This command is intended for diagnostic use only. It should NOT be used as part of normal operations.

11.2 Administrative Command: dtfdsp

dtfdsp start [-dvv] [-A Alias_file] [-I Tran] [-J Dsp] [-W time]
[server_name]

or

dtfdsp {stop |log_file_name|new_log_file|list_log_names} [server_name]

or

dtfdsp display [server_name | ALL]

The **dtfdsp** command is used to start, stop, and control DTF/DU “dispatcher” daemons. A *dispatcher* is a process that listens for requests from MVS users to perform file transfers. When a request is received, a job (similar to a UNIX initiated queued job) is created and executed. The *server_name* value must be communicated to your MDU users. The value is used in MDU command processing.

Entries for the *dtfdsp* command are not automatically added to */sbin/rc3.d* and */sbin/init.d*. The dispatcher uses the ALIAS file as input. Since the ALIAS file will not contain any valid entries when initially installed, an automatically invoked dispatcher would always fail. After the ALIAS file contains valid entries, an entry could be added to */sbin/rc3.d* to automatically invoke *dtfdsp*.

NOTE

If you add an entry for **dtfdsp** to */sbin/rc3.d*, it MUST follow *S57b4_dtfcron* in the ASCII collating sequence to ensure that **dtfboot** executes before IBM initiated jobs.

The first parameter for **dtfdsp** is required and indicates the action that **dtfdsp** is to perform:

start

start a new dispatcher daemon. The daemon will run until told to stop via the dtfdsp stop command, or until a nonretry error condition is encountered.

stop

stop execution of a dispatcher daemon.

log_file_name

display full path name of dispatcher daemon’s log file.

new_log_file

request the dispatcher to close the current log file and start a new log file.

list_log_names

display a list of all log files associated with this dispatcher.

display

display status of this dispatcher or all dispatchers. The status is displayed along with the server name and process id. For this release, possible status values and their meanings are:

<i>not available</i>	a session with MVS is NOT available
<i>available</i>	a session with MVS is available

The SERVER_NAME parameter is the name that the **dtfdsp** process uses to identify itself to MVS. This name corresponds to the MVS user field, SERVER NODE. If not supplied, the UNIX node name (actually the output of UNIX command **hostname -s**) is used.

The following flags and parameters are supported only when the first parameter is *start*:

-d --daemon

Forces dispatcher to run as foreground process instead of as a background daemon. The dispatcher log file is routed to STDERR instead of to a log file.

-[v]v --[very]verbose

The *-v* option results in progress messages being printed to STDERR at the beginning and end of each IBM initiated requests. The *-vv* option results in additional messages being printed during the SNA session startup process.

-A Alias File Name

Name of file containing an entry for each available DEC/SNA gateway. The default value points to the common alias file, /etc/ibm_dtf_alias, created when DTF for Digital UNIX was installed.

-I entry in alias file

Identifier of an individual record in the ALIAS FILE used for MVS initiated file transfer requests. Entry names are defined by the customer when the ALIAS file is edited. The default for this value is the first record in the ALIAS file.

-J entry in alias file

Identifier of an individual record in the ALIAS FILE used by the dispatcher to converse with MDU, the MVS component of DTF/DU. Entry names are defined by the customer when the ALIAS file is edited. The default for this value is the first record in the ALIAS file. This parameter indicates an alias file entry to be used for the dispatcher session.

11.2.1 dtfdsp Examples

Start a dispatcher daemon with very verbose logging using the UNIX node name as the DTF/DU server name.

```
dtfdsp start -vv
```

Display the content of the current log file associated with the dispatcher daemon started in the previous example.

```
cat $(dtfdsp log)
```

Force the dispatcher to close the current log file, open a new one, and display the names of all log files associated with this dispatcher

```
dtfdsp new
```

```
dtfdsp list
```

Terminate the dispatcher started above.

```
dtfdsp stop
```

12. Problem Determination

The Data Transfer Facility for Digital UNIX involves cooperative software on an MVS system and software on a Digital UNIX system. Therefore, solving DTF/DU software problems is a cooperative effort between the UNIX administrator and the IBM system programmer.

The UNIX administrator is responsible for problem resolution on a UNIX node. This individual may also be responsible for trouble shooting problems related to the Digital SNA gateway being used by DTF/DU. The gateway is the component that connects the UNIX system to the SNA network. Analyzing gateway problems may necessitate involving the IBM system's network administrator.

It is easier to detect and solve problems with the DTF/DU software if you understand what the components are and how they interact. This chapter describes each component and explains what occurs during the following DTF/DU file transfers:

- UNIX initiated accesses to an IBM system executed interactively
- UNIX initiated accesses to an IBM system executed in the background
- IBM initiated transfers to a UNIX system that has DTF/DU installed

12.1 DTF/DU Components

- The DTF/DU UNIX commands (dtfiu, dtfui, dtfls, dtfpr, dtfrm, dtfsb) that perform various MVS file access functions.
- The DTF/DU dispatcher daemon is used for IBM initiated transfers.
- The DTF/DU related files that maintain system information for DTF/DU activities.

Referred to in the introduction is the gateway component. Describing the gateway component in this chapter is limited to showing where the gateway fits in the picture. Refer to the gateway documentation listed in the bibliography for detailed information on how to configure/debug gateway related components.

12.1.1 DTF/DU Commands

Commands are separated into file access commands and administrative commands:

File access commands:

dtfui - transfer file from MVS
dtfls - directory list a MVS file
dtfpr - print an MVS file on an MVS printer
dtfrm - remove an MVS file
dtfsb - submit an MVS file for execution on MVS
dtfui - transfer a UNIX resident file to MVS

Administrative commands:

dtfjob - used to display/alter state of user's queued transfer requests.
dtfdsp - used by DTF/DU administrator to start/stop/display dispatcher states.

System files accessed by file access commands:***/etc/ibm_dtf_alias***

contains gateway, transport mode (DECNET or TCP/IP), and SNA information that DTF/DU uses to access the MVS system.

/etc/ibm_dtf_defaults

is a directory which contains files for system and user defaults.

/var/spool/dtf/userid/jobs/

contains files that are used to control the execution of UNIX-initiated queued transfers. Also stored in this path is information related to IBM-initiated transfer requests.

/var/tmp/dtf/userid_lock_batch

is used to serialize submission of queued requests.

System files accessed by administrative commands:***/var/tmp/dtf/***

contains files specific to dispatcher session.

/var/spool/dtf/userid/jobs/

see above.

12.1.2 Dispatcher

The dispatcher process is started by the DTF/DU dtfdsp command and typically executes as a background process. The dispatcher accepts file transfer requests from MVS users and causes these transfers to execute. More than one dispatcher can be started as long as the user-specified server session name is unique. The dispatcher appears as a daemon

running the dtfiid program. The DTF/DU dispatcher daemon can only be started after the IBM network has been initialized. The successful start of the DTF/DU dispatcher daemon results in an SNA session established with MVS.

Files

System files accessed by the dispatcher:

/var/tmp/dtf/server_name.log contains log data from the dispatcher.

/var/tmp/dtf/server_name.socket is used by transfers executing in the background to communicate with the dispatcher.

/var/tmp/dtf/server_name.lock is used by dispatchers to ensure servers on a single system are not started with the same name.

12.2 MDU Components

When debugging problems specific to the MVS components, you will need to consult debugging sections in the MDU manual. The following brief overview is placed here so that, after reading, you will have an understanding of what the MVS components are when working MDU specific problems.

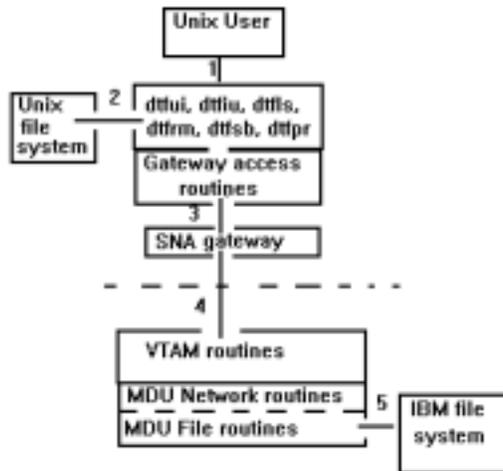
The MDU components are:

1. **MDU command processor** parses MDU commands entered by an MVS user and send requests to the Network Manager.
2. **Network Manager** accepts request from local MVS users, and requests from remote UNIX systems. For local users, it establishes a path between the File Manager and the remote UNIX system and then acts as a router to the remote system. For remotely initiated transfers, it passes the requests to the Node Manager.
3. **File Manager** controls the file transfer.

In MVS, the MDU command processor runs as a TSO command processor, with a set of subcommands. The Network and File Managers run as "subtasks" under a single MDU "address space". This is an important note. When debugging problems related to UNIX initiated transfers, you need only be concerned about what is happening in the address space running the MDU managers.

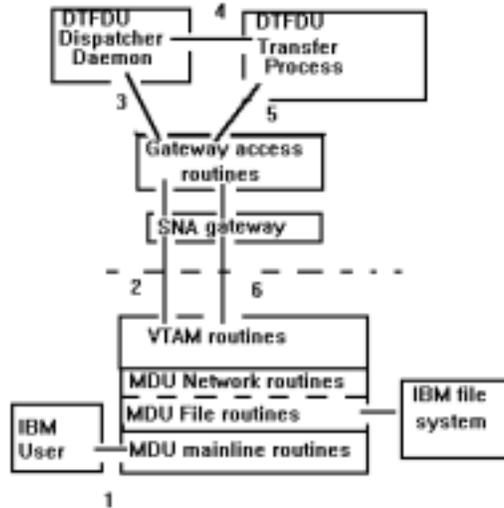
12.3 Walkthroughs and Component Diagrams

The following diagram shows the components involved in a UNIX initiated transfer request:



- 1 - The UNIX user issues a DTF/DU command via the terminal.
 - 2 - If the command must access a UNIX file, then the DTF/DU command connects to the file.
 - 3 - The DTF/DU command accesses the gateway via application program interfaces.
 - 4 - The gateway establishes a connection to the IBM VTAM network.
 - 5 - The MDU component connects to the MVS file system and performs the file access.
- The SNA connection is terminated by the DTF/DU component when the transfer completes.

IBM initiated transfers are somewhat more complex as they involve user interfaces on the IBM system and the DTF/DU dispatcher on the UNIX system:



The diagram does not indicate that the dispatcher process must be started prior to the MVS user initiating the file transfer request. The dispatcher is started via the **dtfdsp start** administrative command and is documented in the Administration section of this manual. The dispatcher session, like transfer sessions above, is an SNA session. However, only DTF/DU commands flow over this transfer. That is, user data does not flow on this session. The following is a review of the steps:

- 1 - The MVS user issues a request via one of the MDU interfaces.
- 2 - The request flows over the VTAM/gateway interfaces to the Dispatcher process.
- 3 - The Dispatcher process causes a file transfer process to be executed.
- 4 - The file transfer proceeds as if the request was UNIX initiated.

Upon completion of the transfer, the MVS user is notified of the success or failure of the transfer.

For a more detailed look, refer to the following transfer requests:

Example 1 - UNIX initiated interactive file transfer

Example 2 - UNIX initiated background transfer that requests checkpoint/restart

Example 3 - IBM initiated file transfer

Example 1 - UNIX initiated interactive file transfer

1. A UNIX user enters a DTF/DU file access command: dtfls, dtfui, dtfpr, dtfrm, dtfsb, dtfui.
2. The command establishes a SNA session with the MDU component which is executing on MVS. The session characteristics are selected from file /etc/ibm_dtf_alias. Session characteristics in the alias file describe whether the link to the Digital SNA gateway is using TCP/IP or DECnet. The alias file also points to a logon mode table used to define the characteristics of the SNA session.
3. The Network Manager, a component of the MDU software executing on the IBM system, passes the request to the Node Manager, another component of MDU.
4. If the data transfer operation is initiated by a dtfui command then data is being sent to MVS. This type of transfer will usually require a file to be created on MVS. The dtfui command, on the other hand, requests that data from the MVS be transferred to the UNIX system. In either case, the MVS file is opened and the data transfer commences. Note that other commands (dtfls, dtfrm, dtfsb) instruct MDU to perform a particular operation on an MVS file and do not initiate file transfers.
5. Once the requested operation is completed, the program executing on the Digital UNIX system terminates the transfer by sending an SNA unbind request to MDU. If the requested operation completes with success then the user is just returned to the UNIX command prompt.

Example 2 - UNIX initiated file transfer executed as a queued job

1. A UNIX user enters either the dtfui or dtfpr command and specifies the -q flag to execute the transfer in the background.
2. A batch job is defined and a directory path is created in /var/spool/dtf/*userid* /jobs/*xxxxx* where *xxxxx* is the jobid assigned to the batch request.
3. If the -h flag (hold) is not also specified the job begins execution and will continue until the transfer completes or is interrupted by a failure. At this

point the transfer behaves very much like the interactive transfer described above so the steps are omitted here. However, the significant difference between the two transfers is that queued transfers are, by default, recoverable. Note that the job will begin execution immediately, if and only if, the “per user” limit on concurrent batch jobs has not been reached. Refer to the administration section of this manual for information on how to modify the “per user” limit (dtfjob command specifying the “preferences” operation.

4. If the transfer completes successfully or with an unrecoverable error then the job enters the completion state and we end our example.
5. If the job terminates with a recoverable error, the job enters the SYSTEM_HOLD state. The UNIX utility “cron” is used to cause the job to reenter execution state.
6. Return to step 4.

Example 3 - IBM initiated file transfer

1. An MVS user supplies transfer parameters to the MDU command processor, command interface, or panel interface (components on the MVS system).
2. The MDU command processor submits the transfer request to the MDU Network Manager subsystem.
3. The MDU Network Manager contacts the UNIX dispatcher (i.e. the UNIX daemon started via dtfdsp) on the UNIX system by using the "server" parameter, entered by the MDU user.
4. The dispatcher starts a background process that creates a transfer session to the IBM side. The copy is assigned a job number and files associated with the transfer are created on /var/spool/df/unix_userid /jobs/job_number....
5. The file transfer proceeds to completion. If the transfer is interrupted and the recovery is possible then the transfer is placed back into SYSTEM_HOLD state. However, unlike the UNIX initiated transfer, the MDU requester must reinitiate the restart operation.
6. The transfer session returns the status to the MDU Network Manager and then terminates
7. The MDU Network Manager passes the status to the command processor, and the command processor passes it to the user.

12.4 Estimating Resource Utilization

In a development environment, DTF/DU will likely be used in an interactive mode where developers are locating, sending, and receiving files between MVS and Digital UNIX. In a production environment, it is expected that batch transfers run concurrently will more

likely be the mode of operation. Executing concurrent transfers in the background will, of course, increase the demand for system resources. This section provides estimates of the virtual storage requirements and process requirements for the DTF/DU file transfer commands, dtfui and dtfui.

12.4.1 Storage Estimates

Each invocation of **dtfui** and **dtfui** will require approximately 3.5 MB of virtual storage on the UNIX system.

To calculate the storage requirements for 10 concurrent transfers we simply perform the math (10*3.5) and arrive at 35MB for a storage requirement. Installations may find the ability to store backups (tar files) onto MVS by piping output from tar into dtfui (using dtfui -) useful. Storage requirements do not take this kind of processing into consideration and will require installations to take a closer look at these types of transfers to assess storage utilization.

DTF/DU does provide a fencing mechanism to limit the number of concurrent file transfers a single user can execute concurrently. Refer to the field `_DTF_MAX_BATCH_RUN` in the System Preferences file in the Administration section of this manual. Note that this value only limits the number of transfers started from the DTF/DU batch queue and does not have an effect on interactive transfers or transfers submitted interactively to execute in the background. To limit nonqueued transfers, you may need to implement process quotas or enforce usage of the queued option.

12.4.2 Process Estimates

Every invocation of a DTF/DU command will cause additional processes to be created by the UNIX system. The demand on process quotas will be higher when queuing transfers to execute from the DTF/DU queue than when executing transfers interactively or in the background.

Interactive use, or transfer commands submitted to the background, will add one to the current process count. Queuing a request, via the `-q` option flag, will require approximately five processes. When the queued transfer is selected for execution approximately five processes will be created.

Messages about missing lock files are a symptom of process quotas being exceeded. For example:

```
/var/spool/DTF/userid/locks/job.12345 not found
```

may appear when a user queues a transfer for execution.

In these cases, use the `dtfjob` administrative command to analyze the contents of the user's queues. Placing jobs in hold status will prevent additional queued jobs from executing. It will be necessary to reinitiate those transfers that failed to get queued because of the problem.

12.5 Isolating DTF/DU Software Problems

MDU and DTF/DU provide interfaces to both UNIX shell users and MVS users. Because of the distribution of the components, the first step in the problem determination process is usually to isolate the problem to either DTF/DU or MDU.

The first goal of problem isolation is to classify the problem as one of the following

- Installation problem
- Usage problem

12.5.1 Installation Problems

Installation problems will typically result in the request being terminated because the SNA session to MDU cannot be established.

Errors establishing sessions with MVS

Requests that fail because a session to MDU cannot be established will terminate the request immediately. However, the reason the session cannot be established may not be quite so obvious. Review the error message reported by DTF/DU. If the message text refers to an SNA problem then contact the MDU installer and have them review the SNA sense codes returned by DTF/DU. The error may be related to a problem with the VTAM definition of the gateway or perhaps a logmode problem. Confirm that the `/etc/ibm_dtf_alias` file on the UNIX system points to the correct gateway, logon mode entry, and VTAM application name.

Sample errors sequences

A user attempts to list an MVS file using `dtfls` and receives:

```
09:46:04 DUN8003E Error detected during function dfci_receive_connect_response
09:46:04 DUN8004E SNA_S_FUNCABORT, access routine function aborted
09:46:04 DUN8005E SNA_S_NOSUCACC, Unknown access name
09:46:04 DUN7508E Error establishing contact with IBM mainframe
```

Notice line 3 of the messages states “Unknown access name”. This error occurred because the Digital SNA gateway is not configured with the access name that was specified in `/etc/ibm_dtf_alias` file. Review the contents of `/etc/ibm_dtf_alias` and update gateway configuration with the correct name. Refer to gateway documentation for procedures on how this is accomplished.

```
9:47:36 DUN8003E Error detected during function dfci_receive_connect_response
09:47:36 DUN8004E SNA_S_FUNCABORT, access routine function aborted
09:47:36 DUN8005E SNA_S_CONREQREJ, connect rejected by IBM host
09:47:36 DUN8006E Sense data: 087D0001
```

09:47:36 DUN7508E Error establishing contact with IBM mainframe

Notice the sense data of 087D0001 and the error messages above it that suggest an SNA error occurred. The IBM VTAM programmer will tell you this means that the application you are attempting to connect to is not defined to MVS. Review the `/etc/ibm_dtf_alias` file to ensure the application name field you entered for MDU agrees with the name the IBM installer defined.

12.5.2 Usage Problems

Usage errors in DTF/DU appear to users as error messages written to `STDERR` and are prefixed with message identifiers (`DUNxxxx`). Issuing `man dtf` provides complete information on DTF/DU error messages and includes corrective action, where possible.

Often, usage problems arise because users are not accustomed to the difference in file systems. For example, mapping stream files to MVS file formats will present record size conflicts if the maximum record size of the MVS file is too small. Sometimes users will use filespec syntax that does not conform to MVS file syntax rules. These types of errors should always produce error messages to the users and they will usually easily identify to the user what the error is. These messages appear with a prefix of “DUN” and are written to `STDERR`. There will be cases where the problem report is simply not clear to the UNIX user. In these cases the DTF/DU administrator and user may have to request advice from the MDU administrator.

A quite common usage error involves the security setup on MVS. UNIX requesters may see the following upon issuing a request to access an MVS file:

09:47:02 DUN7512E File Access, privilege violation (OS denies access).

This may occur because the user did not specify MVS userid and password security information. See the `-U` and `-P` command option flags. If this information was specified then contact the MVS security administrator and ask them to review security logfiles for the cause of the error.

The final portion of this manual lists all the `DUNxxxx` messages that users can receive from the DTF/DU component. `DUN` messages are also available online via `man dtf`.

12.6 Information Gathering

This section discusses methods for gathering problem determination information on the DTF/DU software. These methods range from gathering information on the execution of DTF/DU commands to gathering information on the actual network links used to carry the DTF/DU transfer.

Using the command line option

The `-vv` (very verbose) command line option allows a user to see detailed progress of a UNIX initiated request. All DTF/DU file access commands support this option. In addition, the `dtfdsp` command, used to start the dispatcher session, also supports the `-vv` option.

`-vv` output is logged to `STDERR`. Recorded to the log are various timestamped messages that report the progress of the access to the remote IBM system. Software version numbers, and command line defaults, are also written.

The `-vv` option flag will not cause batch jobs to execute in verbose mode. To cause the batch jobs to operate in verbose mode, use the `dtfjob` administrative command with the `unix` option and assign `-vv` to the environmental variable “`_DTF_USER_FLAGS`”. The generated output will be written to the job’s log file and can be viewed with “`dtfjob log`”. Likewise, to cause IBM initiated jobs to log in verbose mode, use the `dtfjob` administrative command with the `ibm` option and assign `-vv` to the environmental variable “`_DTF_USER_FLAGS`”.

Using the `SNALOG_MASK` facility

It may be necessary to run an additional level of trace if an application protocol error is suspected. The user enables this additional tracing by setting shell variable names to specific values and exporting these variable.

`SNALOG_MASK` is a four byte hexadecimal number with specific bits defined to enable various traces and trace options. For debugging application protocol problems we will ask for the following:

```
export SNALOG_MASK=70F00000
```

If the problem appears related to a lower level protocol, a flag value of `70F000FF` may be requested.

The default location of the trace output is `snalog.dat` in the current directory. Optionally, you can override the location by specifying an additional environmental variable called `SNALOG_FILE`. For example:

```
export SNALOG_FILE=/var/spool/snalog.dat
```

will cause the trace file to be recorded to `/var/spool/snalog.dat`.

12.7 Messages and Codes

This section contains the messages that a Digital UNIX station can receive when performing an access to the IBM system using DTF/DU. Often, messages include variable text strings associated with the particulars of the transfer and are helpful

in determining the nature of the problem. Markers for these strings are included below as <text>.

DUN0001I SHUTDOWN request received: <text>

Cause: A shutdown request has been received during a copy operation. The message contains text received with the shutdown request.

Meaning: The copy operation is terminated.

Corrective Action: None required.

DUN0101E An IBM filename is a required for this command

Cause: An IBM file name was not included on the command line, or an error in processing some other command line parameter prevented the file name from being located.

Meaning: The requested operation has not been performed. All commands require an IBM file name as a parameter.

Corrective Action: Review and correct the command line parameters.

DUN0102E Both a UNIX filename and an IBM filename are required

Cause: Both an IBM file name and a UNIX file name are required parameters for copy commands. Either one of the file names is missing, or an error in processing some other command line parameter prevented filenames from being located.

Meaning: The requested operation has not been performed. All copy commands require an IBM file name as a parameter.

Corrective Action: Review and correct the command line parameters.

DUN0103E Invalid or unsupported <text> on command line

Cause: The character string indicated in the message was found on the command line. It is either an unsupported option or an error in processing some other command line parameter prevented this option from being recognized.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0104E The '-q' (queue) cannot be used with '-' (STDIN/STDOUT)

Cause: The combinations of queued processing and use UNIX STDIN or STDOUT is not supported.

Meaning: The requested operation has not been performed.

Corrective Action: Replace the '-' with a UNIX file name or run the copy without the -q option.

DUN0151E -K <text> is invalid or not supported

Cause: The keyword following the -K flag is not recognized. The error condition is associated with the base keyword, not a leading no or a value following a '!'.
Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0152E -K <text> requires a value (-K keyword:value)

Cause: The keyword following the -K requires a parameter value to follow the keyword. For instance, the sequence '-K blocksize:8000' is valid but the sequence '-K blocksize' would cause this error message.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0153E -K <text> cannot be combined with 'no'

Cause: The keyword following the -K is supported, but not with a prefix of 'no'.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0154E -K <text> : value contains too many characters

Cause: The keyword following the -K is supported, but the value following the : contains too many characters. The number of characters allowed varies by keyword.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0155E -K <text> : value must be a positive integer

Cause: The keyword following the *-K* is supported, but the value following the *:* can not be interpreted as a positive integer.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0156E -K <text> : value is too large

Cause: The keyword following the *-K* is supported, but the value following the *:* exceeds the maximum value allowed for this keyword.

This message most frequently appears when a `BLOCK_SIZE` or `MRS` (maximum record size) value is being specified. The maximum value for `BLOCK_SIZE` is 32,767. The maximum value for `MRS` is 32,759 (`BLOCK_SIZE - 8`).

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0156E -K <text> : value must begin with T or C for TRK or CYL

Cause: In the *-K align : value* parameter, the value field did not start with *T* for track or *C* for cylinder. These are the only supported values for this keyword.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0157W -K <text> is not supported and is ignored by command

Cause: The *-K* keyword parameter is not supported for use with this command. The keyword is valid and supported for some other command.

Meaning: The *-K* keyword parameter and its associated value is ignored and command execution continues.

Corrective Action: No action required.

DUN0158E -K <text> value must be 800, 1600, or 6250

Cause: In the *-K density:value* parameter, the value field was not 800, 1600, 6250. Only these values are supported for this keyword.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0159E -K <text> value must be NL,SL,NSL,SUL,BLP,LTM,AL,AUL

Cause: In the -K label:value parameter, the value field was not one of the supported values.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0160E -K <text> value must be IGNORE,NO,NONE, or a number

Cause: In the -K null:value parameter, the value field was not a supported value. Supported values are NO, NONE, IGNORE, or a number. Numbers can be expressed as octal, decimal, or hexadecimal values that fall in the range of zero through 255 (base 10).

To express a number in octal format, begin the number with a 0. To express a number in decimal format, begin the number with a digit of 1 through 9. To express a number in hexadecimal notation, begin the number with %x or 0x.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0161E -K <text> value must begin with F or V for fixed or variable

Cause: In the -K recfm:value parameter, the value field did not begin with an F or V. IBM mainframe record formats of fixed blocked and variable blocked are supported. Only the first letter following ':' is examined.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0162E -K <text> value must be YES, NO, or NONE

Cause: In the -K catalog:value parameter, the value field was not YES, NO, or NONE. Only these values are supported for the catalog keyword.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0163W -K <text> is not supported in this version of DTF

Cause: The -K keyword is valid but the prefix 'no' is not allowed.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0164E -K <text> table specified cannot be located or read

Cause: The translate table specified cannot be used. Another error message explaining the nature of the error should have preceded this message.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0165E -K <text> prefix 'no' cannot be combined with a :choice

Cause: The -K translate keyword can be preceded with 'no' or followed with a ':' and a table name. The combination of 'no' and a ': table name' is not allowed.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0166E Checkpoint and Append are mutually exclusive options

Cause: Since we cannot recover after an append operation has been interrupted, we do not support checkpoints on these transfers.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0167E Value invalid for -M. Value must be between 1024 and 32767

Cause: Requester specified invalid value for -M.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the command line parameters.

DUN0168W Extraneous parameters ignored after <text>

Cause: Requester specified too many command parameters.

Meaning: The requested operation continues.

Corrective Action: Review and correct the command line parameters.

DUN0169W The -K post: is ignored when the -q option is not specified

Cause: Requester specified -K post: without the -q option.

Meaning: The requested operation continues, but the post processing option is ignored.

Corrective Action: Review and correct the command line parameters.

DUN1001I Create connection to gateway: <text>

Cause: Results from selecting -v or -vv on the command line.

Meaning: This message is issued just before the dtf module attempts to create a connection with the DEC/SNA gateway. The <text> field is replaced with the gateway node name.

Corrective Action: No action needed.

DUN1002I Request BIND from mainframe

Cause: Results from selecting -v or -vv on the command line.

Meaning: The dtf module has successfully connected to the DEC/SNA gateway and is now requesting the gateway to contact the dtf component on the IBM mainframe.

Corrective Action: No action needed.

DUN1003I BIND from mainframe accepted

Cause: Results from selecting -v or -vv on the command line.

Meaning: An SNA BIND request has been received from the IBM mainframe and has been accepted. The dtf module has sent a positive response.

Corrective Action: No action needed.

DUN1004I Start intentional delay of <text> micro seconds

Cause: Results from setting non-zero value as tenth parameter of alias file entry. This delay is required when using some older versions of DEC/SNA gateways.

Meaning: The command delays for the length of time indicated.

Corrective Action: No action needed.

DUN1005I Dispatcher session starting : <text>

Cause: Results from selecting -v or -vv on the command line.

Meaning: This message is issued just before the dtf module attempts to create a connection with the DEC/SNA gateway. The <text> field contains the user-specified server name.

Corrective Action: No action needed.

DUN1006I <text>

Cause: Results from selecting -v or -vv on the command line.

Meaning: This message reflects the progress during a file transfer. The number of records, number of bytes, number of seconds, and approximate transfer rate are displayed.

Corrective Action: No action needed.

DUN1007I Checkpoint being taken

Cause: Results from selecting `-v` or `-vv` on the command line.

Meaning: This message reflects that we are in checkpoint processing.

Corrective Action: No action needed.

DUN1098I Intentional break at record <text>

Cause: Results from selecting `-B` and/or `-N` on the command line. These parameters request a simulated network outage during a file transfer operation.

Meaning: This message displays the record number within the file where the next simulated network outage will occur. The `-B` parameter specifies the break point for the first attempt. The `-N` parameter specifies additional records to be transferred on each re-try.

The `-B` and `-N` capabilities are not supported for use by customers. They are intended for internal regression testing only.

Corrective Action: No action needed.

DUN1100I Input command: <text>

Cause: Results from selecting `-vv` on the command line.

Meaning: The explicitly specified and the default parameter settings for the current command are displayed. Multiple messages with this message code are displayed. The first message contains the name of the command. The next line displays the lower case flags bytes (including `-vv`). One message is generated for each upper case flag/value pair. Finally, IBM and UNIX file names are displayed.

The IBM mainframe password value is not displayed. If a value is available to DTF, the character string `<supplied>` is displayed.

Corrective Action: No action needed.

DUN1101I DAP message received : <text>

Cause: Results from selecting `-vv` on the command line.

Meaning: This message indicates that a DEC ACCESS PROTOCOL (DAP) message has been received. DAP messages are used to exchange control information before and during DTF operations.

The particular type of DAP message is documented at the end of the line. This message type is included as a diagnostic aid for customer service representatives.

Corrective Action: No action needed.

DUN1102I DAP message sent : <text>

Cause: Results from selecting -vv on the command line.

Meaning: This message indicates that a DEC ACCESS PROTOCOL (DAP) message has been transmitted. DAP messages are used to exchange control information before and during dtf operations. The particular type of DAP message is documented at the end of the line. This message type is included as a diagnostic aid for customer service

Corrective Action: No action needed.

DUN1111I IBM Mainframe DTF version : <text>

Cause: Results from selecting -vv on the command line.

Meaning: This message reports the IBM mainframe DTF component's version number contained in the SNA BIND request.

Corrective Action: No action needed.

DUN1112I IBM Mainframe DTF maint.lvl : <text>

Cause: Results from selecting -vv on the command line.

Meaning: This message reports the IBM mainframe DTF component's maintenance level contained in the SNA BIND request.

Corrective Action: No action needed.

DUN1113I IBM Mainframe DTF ACCESS # : <text>

Cause: Results from selecting -vv on the command line.

Meaning: This message reports the IBM mainframe DTF component's ACCESS number for the current operation. This number may be useful, during problem determination, as it is also displayed on the IBM mainframe.

Corrective Action: No action needed.

DUN1121I DEC SNA Gateway node : <text>

Cause: Results from selecting -vv on the command line.

Meaning: This message reports the node name of the DEC/SNA gateway that will be used. This value is derived from the alias file entry associated with the current operation.

Corrective Action: No action needed.

DUN1122I Application ID : <text>

Cause: Results from selecting -vv on the command line.

Meaning: This message reports the VTAM application ID of the mainframe DTF component that will be used. This value is derived from the alias file entry associated with the current operation.

Corrective Action: No action needed.

DUN1123I Access name : <text>

Cause: Results from selecting -vv on the command line.

Meaning: This message reports the DEC/SNA gateway access name that will be used. This value is derived from the alias file entry associated with the current operation.

Corrective Action: No action needed.

DUN1124I Logmode name : <text>

Cause: Results from selecting -vv on the command line.

Meaning: This message reports the SNA logmode name that will be used. This value is derived from the alias file entry associated with the current operation.

Corrective Action: No action needed.

DUN1125I Gateway type : <text>

Cause: Results from selecting -vv on the command line.

Meaning: This message reports the DEC/SNA gateway type indicated in the connect response message.

Corrective Action: No action needed.

DUN1126I Gateway Logical Unit : <text>

Cause: Results from selecting -vv on the command line.

Meaning: This message reports the DEC/SNA gateway logical unit name in the connect response message.

Corrective Action: No action needed.

DUN1127I Data Compression in use

Cause: Results from selecting -vv on the command line.

Meaning: This message indicates that compression is in use during this data transfer.

Corrective Action: No action needed.

DUN2001E Cannot create socket <text>

Cause: A UNIX socket with the specified name cannot be created.

Meaning: The program attempting to create the socket will terminate. Both the IBM dispatcher and file copy operations attempt to create sockets.

Corrective Action: Examine the UNIX errno value reported in the next message to determine the real reason for the failure.

DUN2002E Cannot bind socket <text>

Cause: A UNIX socket with the specified name cannot be bound and therefore cannot be used.

Meaning: The program attempting to bind the socket will terminate. Both the IBM dispatcher and file copy operations attempt to bind sockets.

Corrective Action: Examine the UNIX errno value reported in the next message to determine the real reason for the failure.

DUN2003E UNIX Select failed for socket

Cause: A UNIX select failed while testing for input on the named socket.

Meaning: The program attempting to select will terminate. Both the IBM dispatcher and file copy operations attempt to bind sockets.

Corrective Action: Examine the UNIX errno value reported in the next message to determine the real reason for the failure.

DUN2004E UNIX recvfrom failed for socket

Cause: A UNIX recvfrom failed while attempting to receive data.

Meaning: The program attempting to recvfrom will terminate. Both the IBM dispatcher and file copy operations attempt to bind sockets.

Corrective Action: Examine the UNIX errno value reported in the next message to determine the real reason for the failure.

DUN3001E Error in alias file at line <text>

Cause: There is an error in the alias file at the indicated line number. Another error message should follow this message which explains the error in more detail.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the alias file.

DUN3002E Error description: <text>

Cause: There is an error in the alias file. The text indicates the field in error and the cause of the error. A preceding message should have specified the line number in the file.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the alias file.

DUN3003E Limit of 50 alias entries exceeded

Cause: There are too many entries in the alias file.

Meaning: The requested operation has not been performed.

Corrective Action: Create multiple alias files with fewer than 50 entries per file. Use the -A filename parameter to indicate which alias file a given command should use.

DUN3004E Each alias file line must contain at least 9 values

Cause: There is an error in the alias file. Each line in the alias file must contain 9 or more fields. A previous message should indicate the line number within the alias file. Refer to the sample alias file shipped with the product for a valid format for the alias file.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the alias file.

DUN3005E No alias entries found in alias file

Cause: The end of the alias file was reached before finding any lines which define valid entries.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the alias file.

DUN3006E Could not open alias file named <text>

Cause: The indicated alias file could not be opened. See the next message for additional information.

Meaning: The requested operation has not been performed.

Corrective Action: Review and correct the alias file.

DUN3007E Alias file errno value = <text>

Cause: The alias file named in the preceding message could not be opened. This message documents the UNIX 'errno' value which may provide more information about the nature of the error.

Meaning: The requested operation has not been performed.

Corrective Action: Review the UNIX 'errno' value and correct the alias file access problem.

DUN3008E Alias entry <text> not found

Cause: The entry with the indicated name could not be found in the alias file.

Meaning: The requested operation has not been performed.

Corrective Action: Attempt an interactive DTF command using the `-vv` flag and the same

`-A` and `-I` values. For instance, if the failing command was:

```
dtfui -i -I joe /vmUNIX mf.a.b.c
```

type a command similar to the following:

```
dtfls -vv -I joe mf.a.b.c.
```

Scan the output for 1100I Input command: `-A xx` and verify that the file name following `-A` is the alias file that should contain the alias entry.

Once the alias filename is verified, visually inspect the file for the required entry.

If the problem persists, try using an entry in the sample alias file distributed with the product. Use of these entries normally results in a communications error, but if this error message no longer appears, the problem is in the installation-modified alias file.

DUN4101W Excess parameters ignored - <text>

Cause: Too many parameters were supplied for the requested action. The error message text contains the value of the excess parameters.

Meaning: The requested operation continues, but the excess parameters are ignored.

Corrective Action: None required.

DUN4102E Requested job does not exist - <text>

Cause: The requested job does not exist. The error message text contains the requested job number.

Meaning: The requested operation will not be performed.

Corrective Action: Repeat the request with a valid job number.

DUN4103E Job cannot change states from <text>

Cause: A job is not allowed to change from its current state to the requested state. The error message text contains the current and requested states.

Meaning: The requested operation is not performed.

Corrective Action: None required.

DUN4104E Invalid parameter specified - <text>

Cause: The supplied parameter is invalid or not supported. The error message text contains the value of the invalid parameter.

Meaning: The requested operation is not performed

Corrective Action: Correct the action requested and retry the command.

DUN4105E Cannot change state of IBM initiated request from <text>

Cause: A job is not allowed to change from its current state to the requested state. The error message text contains the current and requested states.

Meaning: The requested operation is not performed.

Corrective Action: None required.

DUN4110I <text> is the location of this log

Cause: Batch (queued) or IBM initiated job has been defined

Meaning: This message documents the job log location. You can use an editor to view the file or issue DTFJOB LOG job number to view the log.

Corrective Action: None required.

DUN4111I <text> job defined

Cause: Batch (queued) or IBM initiated job has been defined

Meaning: The job definition took place at the time indicated in the text of the message. The initial status of the job is USER_HOLD.

Corrective Action: None required.

DUN4112I Status change from <text>

Cause: The status of a batch (queued) or IBM initiated job has changed as indicated in the text portion of the message.

Meaning: The job's status has changed.

Corrective Action: None required.

DUN4113I Job ended with return code = <text>

Cause: A batch (queued) or IBM initiated job has ended. The text portion of the message contains the return code.

Meaning: The job is no longer executing. A preceding message should have documented the state change. This message indicates the job's success or failure.

Corrective Action: Action depends on return code and jobs status as follows:

0 - no action needed / job completed successful

1 - copy operation failed / retry recommended

2 - copy operation failed / no retry allowed

3 - parameter error detected during job execution

* - all other codes indicate non-retryable condition

Examine the log file for additional error messages that may more fully document the cause of any error.

DUN4114I <text>

Cause: The scheduling priority of a batch job has been changed to the value indicated in the text of the message.

Meaning: Batch (or queued) jobs with status of system hold and a current time delay value of 0 are scheduled on a priority basis. The priority of the current job has been changed to the indicated value.

Corrective Action: None required.

DUN4115I <text>

Cause: A batch (queued) or IBM initiated job added the text of this message to the log file.

Meaning: This is an informational message only. See message text.

Corrective Action: None required.

DUN4116W Batch Job Delayed

Cause: An attempt to create a batch job (*dtfui* or *dtfiu -q*) has successfully completed but execution of the job has been delayed.

Meaning: The batch job was not started because some limit on the number of currently executing jobs has been reached. The job's status has been set to system hold. The job be automatically scheduled when the limit of executing jobs is no longer exceeded.

Corrective Action: None required. Recommended action is to wait and allow the job to be scheduled at an appropriate time. The **dtfjob** command with the list option can be used to display all queued jobs. The dtfjob command with the priority option can be used to influence the order in which jobs with status system hold will be attempted.

To force the job to execute regardless of limits on concurrent execution, use the dtfjob command with the execute option. This is NOT recommended.

DUN4117E transfer request rejected - too many transfers in progress

Cause: An IBM initiated transfer requests has been rejected because a limit on the number of concurrent transfers has been reached.

Meaning: The transfer attempt has failed. No user data has been transferred.

Corrective Action: Repeat the transfer request later and/or increase the limit on number of concurrent IBM initiated transfers.

DUN4401E You must be root or super-user to perform this action

Cause: A request to edit a system wide defaults or profile file has been made by a non-root or non-super user account.

Meaning: The requested operation is not performed

Corrective Action: Logon as root or become super-user and repeat the command.

DUN4402E Job number is required for action <text>

Cause: A job number is a required parameter when the indicated action is specified. The error message text contains the value of the action requested.

Meaning: The requested operation is not performed

Corrective Action: Repeat the command with the job number following the action.

DUN4406I <text> is current date and time

Cause: A display of the cron processing log has been requested. The text portion of this message is replaced with the current date and time. This can be compared to the date and time in the cron log file.

Meaning: This message contains the current date and time

Corrective Action: None required.

DUN4407I <text>

Cause: A display of the cron processing log has been requested. This text portion of this message contains one line from the log file.

Meaning: This message contains a copy of data from the cron log.

Corrective Action: None required.

DUN4408W No cron log files exists

Cause: A display of the *cron* processing log has been requested but no log file exists. This indicates that the *dtfcron* process has never executed (or that the log file has been manually deleted).

Meaning: This message indicates that the *dtfcron* log file is not available.

Corrective Action: This log file should only exist when both of the following are true:

1. A batch (or queued) job is executing or in system hold state.
2. UNIX *cron* processing has scheduled the *dtfcron* procedure

If these conditions are not both true, then no action is required. To start, or restart, the *dtfcron* process, do the following:

1. Execute command *dtfboot reset*
2. Wait at least one minute.
3. Repeat *dtfjob cronlog* command

DUN6001E Invalid length - <text>

Cause: During a UNIX to IBM copy operation using FIXED LENGTH records on the IBM mainframe and NewLine delimited records on the UNIX node, a UNIX data file record was encountered that was not the required size. The error message text contains the record number as well as the expected and actual data lengths.

Meaning: The file copy operation is terminated.

Corrective Action: Review the UNIX file used as input to the copy operation. If the data file is correct, then the copy operation cannot

succeed with the selected parameters. Consider using variable length records (-K recfm:variable) on the IBM mainframe file.

DUN6501E UNIX function fprintf failed for checkpoint file

Cause: The return code from UNIX function fprintf indicated an error condition. The error occurred while written a checkpoint/restart record to the checkpoint file. Checkpoint files are normally used during IBM initiated copies and during user initiated 'queued' copies. The checkpoint file is located in the */var/spool/df/user_id/jobs/job_id* directory.

Meaning: The file copy operation is terminated.

Corrective Action: Determine the cause the error by examining the UNIX errno value documented the error message following this message. Correct the error an restart or retry the copy operation.

DUN6502E UNIX function fflush failed for checkpoint file

Cause: The return code from UNIX function fflush indicated an error condition. The error occurred while writing a checkpoint/restart record to the checkpoint file. Checkpoint files are normally used during IBM initiated copies and during user initiated 'queued' copies. The checkpoint file is located in the */var/spool/df/user_id/jobs/job_id* directory.

Meaning: The file copy operation is terminated.

Corrective Action: Determine the cause the error by examining the UNIX errno value documented the error message following this message. Correct the error and restart or retry the copy operation.

DUN6503E UNIX function fclose failed for checkpoint file

Cause: The return code from UNIX function fclose indicated an error condition. The error occurred at the end of a file transfer operation. Checkpoint files are normally used during IBM initiated copies and during user initiated 'queued' copies. The checkpoint file is located in the */var/spool/df/user_id/jobs/job_id* directory.

Meaning: The file copy operation is terminated.

Corrective Action: Determine the cause of the error by examining the UNIX errno value documented in the error message following this message. Correct the error and restart or retry the copy operation.

DUN6504E UNIX function fopen/read mode failed for checkpoint file

Cause: The return code from UNIX function fopen indicated an error condition. The error occurred while opening the checkpoint file in read mode. This is done at the very beginning of a copy operation to determine if the transfer operation should be restarted at a checkpoint. Checkpoint files are normally used during IBM initiated copies and during user initiated 'queued' copies. The checkpoint file is located in the /var/spool/df/user_id/jobs/job_id directory.

Meaning: The file copy operation is terminated.

Corrective Action: Determine the cause of the error by examining the UNIX errno value documented in the error message following this message. Correct the error and restart or retry the copy operation.

DUN6551E Could not open translate table file <text>

Cause: The translate table specified in the -K trans:table parameter can not be opened for input. The translate table file name is contained in the message.

Meaning: The file copy operation is terminated.

Corrective Action: Determine the cause of the error by examining the UNIX errno value documented in the error message following this message. Correct the error and restart or retry the copy operation.

DUN6552E Error in translate table <text>

Cause: The translate table specified in the -K trans:table parameter can be opened, but cannot be processed. The translate table file name is contained in the message. The line number associated with the error is also contained in the error message.

Meaning: The file copy operation is terminated.

Corrective Action: Determine the cause the error by examining reading and examining the translate table file.

DUN7001E UNIX function access failed for file <text>

Cause: The return code from UNIX function access indicated an error condition. This function is used to verify the UNIX user's access rights to the UNIX file.

Meaning: The file copy operation is terminated.

Corrective Action: Determine the cause the error by examining the UNIX errno value documented the error message following this message. Correct the error and restart or retry the copy operation.

DUN7002E UNIX function stat failed for file <text>

Cause: The return code from UNIX function stat indicated an error condition. This function is used to retrieve information about the UNIX file. This function is used to verify that the supplied file is a 'regular' file.

Meaning: The file copy operation is terminated.

Corrective Action: Determine the cause of the error by examining the UNIX errno value documented the error message following this message. Correct the error and restart or retry the copy operation.

DUN7003E UNIX function fopen failed for file <text>

Cause: The return code from UNIX function fopen indicated an error condition. This function is used to open the UNIX file.

Meaning: The file copy operation is terminated.

Corrective Action: Determine the cause of the error by examining the UNIX errno value documented the error message following this message. Correct the error and restart or retry the copy operation.

DUN7004E UNIX file <text> is not a regular file

Cause: The specified UNIX file appears to be a directory, a socket, a pipe, or some other type of 'special' file.

Meaning: The file copy operation is terminated.

Corrective Action: Specify only 'regular' UNIX files -- not directories, sockets, pipes, etc.

DUN7005E UNIX function fsetpos failed after open of file <text>

Cause: The return code from UNIX function fsetpos indicated an error condition. This function is used to position a UNIX file when restarting a file transfer using checkpoint/restart data.

Meaning: The file copy operation is terminated.

Corrective Action: Determine the cause the error by examining the UNIX errno value documented the error message following this message. Correct the error and restart or retry the copy operation.

DUN7006E UNIX function fgetpos failed for file <text>

Cause: The return code from UNIX function fgetpos indicated an error condition. This function is used to retrieve the current position in the UNIX file when recording a checkpoint.

Meaning: The file copy operation is terminated.

Corrective Action: Determine the cause of the error by examining the UNIX errno value documented in the error message following this message. Correct the error and restart or retry the copy operation.

DUN7007E UNIX function read failed for file <text>

Cause: The return code from UNIX function read indicated an error condition. This function is used to read data during a UNIX to IBM transfer when the -i option was specified on the command line.

Meaning: The file copy operation is terminated.

Corrective Action: Determine the cause of the error by examining the UNIX errno value documented in the error message following this message. Correct the error and restart or retry the copy operation.

DUN7008E UNIX function fgets failed for file <text>

Cause: The return code from UNIX function fgets indicated error or end of file condition. A UNIX function ferror returned a code indicating an error.

Meaning: The file copy operation is terminated.

Corrective Action: Determine the cause of the error by examining the UNIX errno value documented in the error message following this message. Correct the error and restart or retry the copy operation.

DUN7009E UNIX function fflush failed for file <text>

Cause: The return code from UNIX function fflush indicated an error condition. This function is used to force any buffered data to be written to an output file to a during a UNIX to IBM transfer when the -i option was specified on the command line.

Meaning: The file copy operation is terminated.

Corrective Action: Determine the cause of the error by examining the UNIX errno value documented in the error message following this message. Correct the error and restart or retry the copy operation.

DUN7010E UNIX function fwrite failed for file <text>

Cause: The return code from UNIX function fwrite indicated an error condition. This function is used to write data during a UNIX to IBM transfer when the -i option was specified on the command line.

Meaning: The file copy operation is terminated.

Corrective Action: Determine the cause of the error by examining the UNIX errno value documented in the error message following this message. Correct the error an restart or retry the copy operation.

DUN7011E UNIX function fclose failed for file <text>

Cause: The return code from UNIX function fclose indicated an error condition. This function is used at the end of file transfer.

This error indicates that the file has NOT been copied in a reliable manner. For IBM to UNIX transfers, there may be data missing at the end of the file. For UNIX to IBM transfers, this error happened before the normal end of transfer message sequence.

Meaning: The file copy operation is terminated. The file on receiving system is probably NOT an accurate copy of the file on the receiving system.

Corrective Action: Determine the cause of the error by examining the UNIX errno value documented in the error message following this message. Correct the error and restart or retry the copy operation.

DUN7012E UNIX function truncate failed for file <text>

Cause: The return code from UNIX function truncate indicated an error condition. This function is used when re-starting an IBM to UNIX file transfer. The file on the UNIX system is truncated at the point where the last successful checkpoint (or at the beginning of the file when no checkpoint was taken before the original failure).

Meaning: The file copy operation is terminated.

Corrective Action: Determine the cause the error by examining the UNIX errno value documented the error message following this message. Correct the error and restart or retry the copy operation.

DUN7013E Not enough space: required / available = <text>

Cause: There is not enough space in the UNIX file system to receive the file. The message contains the required and available space in bytes.

If this is a retry of a recoverable copy, the required space reflects the amount of additional space needed after the partially received file has been truncated to match the checkpoint/restart point.

Meaning: The file copy operation is terminated.

Corrective Action: Use a different UNIX file system that has more space.

DUN7014E UNIX function statfs failed for file <text>

Cause: The return code from UNIX function statfs indicated an error condition. This function is used when receiving into a UNIX regular file to determine the amount of free space in the UNIX file system.

Meaning: The file copy operation is terminated.

Corrective Action: Determine the cause the error by examining the UNIX errno value documented the error message following this message. Correct the error and restart or retry the copy operation.

DUN7501E Attempt to close file failed for file <text>

Cause: An error occurred while attempting to close an IBM mainframe file. These files are closed at the end of successful and unsuccessful file transfer operations. This error message can also happen due to failures during checkpoint processing on a UNIX to IBM transfer.

This message indicates the operation was being attempted when an error condition was detected. Other messages should precede or follow this message which should describe the error condition in more detail.

Meaning: The file copy operation is terminated.

Corrective Action: This message should be preceded or followed by other error messages that indicate the real cause of the problem. Examine those error messages, correct the error condition, and restart or retry the copy operation.

DUN7502E Attempt to re-access file failed for file <text>

Cause: An error occurred while attempting to take a checkpoint during a UNIX to IBM file transfer. This message simply indicates the operation that was being attempted when an error condition was detected. Other messages should precede or follow this message which should describe the error condition in more detail.

Meaning: The file copy operation is terminated.

Corrective Action: This message should be preceded or followed by other error messages that indicate the real cause of the problem. Examine those error messages, correct the error condition, and restart or retry the copy operation.

DUN7503E Attempt to re-access file failed for file <text>

Cause: An error occurred while attempting to take a checkpoint during a UNIX to IBM file transfer. This message simply indicates the operation that was being attempted when an error condition was detected. Other messages should precede or follow this message which should describe the error condition in more detail.

NOTE

Messages DUN7501, DUN7502, DUN7503, and DUN7504 are closely related. Each of these messages can indicate a error condition during checkpoint processing.

Meaning: The file copy operation is terminated.

Corrective Action: This message should be preceded or followed by other error messages that indicate the real cause of the problem. Examine those error messages, correct the error condition, and restart or retry the copy operation.

DUN7504E Continue after checkpoint failed for file <text>

Cause: An error occurred while attempting to continue data transfer after taking a checkpoint during a UNIX to IBM file transfer. This message simply indicates the operation that was being attempted when an error condition was detected. Other messages should precede or follow this message which should describe the error condition in more detail.

NOTE

Messages DUN7501, DUN7502, DUN7503, and DUN7504 are closely related. Each of these messages can indicate a error condition during checkpoint processing.

Meaning: The file copy operation is terminated.

Corrective Action: This message should be preceded or followed by other error messages that indicate the real cause of the problem. Examine those error messages, correct the error condition, and restart or retry the copy operation.

DUN7505E Attempt to send data failed for file <text>

Cause: An error occurred while attempting to send ordinary user data during a UNIX to IBM file transfer. This message simply indicates the operation that was being attempted when an error condition was detected. Other messages should precede or follow this message which should describe the error condition in more detail.

Meaning: The file copy operation is terminated.

Corrective Action: This message should be preceded or followed by other error messages that indicate the real cause of the problem. Examine those error messages, correct error condition, and restart or retry the copy operation.

DUN7506E Signal or Error while polling for IBM msg

Cause: An error was detected during a UNIX to IBM file transfer. Periodically during a UNIX to IBM file transfer, a check for input error conditions is performed. During this transfer, some unexpected event has been detected. Other messages should precede or follow this message which should describe the error condition in more detail.

Meaning: The file copy operation is terminated.

Corrective Action: This message should be preceded or followed by other error messages that indicate the real cause of the problem. Examine those error messages, correct the error condition, and restart or retry the copy operation.

DUN7507E Unexpected message received while sending IBM file <text>

Cause: An error message was received from the IBM mainframe during a UNIX to IBM file transfer. Periodically during a UNIX to IBM file transfer, a check for input error conditions is performed. During this transfer, a message (other than an anticipated checkpoint request) has been received.

Meaning: The file copy operation is terminated.

Corrective Action: This message should be preceded or followed by other error messages that indicate the real cause of the problem. Examine those error messages, correct the error condition, and restart or retry the copy operation.

DUN7508E Error establishing contact with IBM mainframe

Cause: An error was detected while attempting to establish contact with an IBM mainframe or with the mainframe component of DTF.

Meaning: The requested operation is not performed.

Corrective Action: This message should be preceded or followed by other error messages that indicate the real cause of the problem. Examine those error messages, correct the error condition, and retry the operation.

DUN7510E Attempt to receive data failed for file <text>

Cause: An error was detected while attempting to receive data during an IBM to UNIX file transfer operation.

Meaning: The requested operation is not performed.

Corrective Action: This message should be preceded or followed by other error messages that indicate the real cause of the problem. Examine those error messages, correct the error condition, and retry the operation.

DUN7512E <text>

Cause: The mainframe DTF component has detected an error.

Meaning: An error was detected while attempting to access a mainframe file. A status message was returned by the mainframe to report the error.

Corrective Action: Review message text for corrective action. Additional information may appear on the IBM Contact IBM system support personnel for assistance in debugging.

DUN7514E HSM recall failed for IBM file <text>. Specify -K HSMRECALL

Cause: The mainframe DTF component has detected an archived file and has decided to return an error rather than recall the file.

Meaning: The file you are attempting to access has been archived, perhaps because of lack of use. DTF will not HSM recall the file unless explicitly requested to do so.

Corrective Action: Use *-K HSMRECALL* to request DTF to recall the file.

DUN7515E STV error received from IBM mainframe component <text>

Cause: The mainframe DTF component has detected an error.

Meaning: An error was detected while attempting to access a mainframe file. A status message was returned by the mainframe to report the error.

Corrective Action: Review message text for corrective action. Additional information may appear on the IBM system console or log. You may need to contact IBM system support personnel for assistance in debugging.

DUN8001E Mainframe DTF component does not support DTF on DIGITAL UNIX

Cause: The mainframe DTF component does not support or is not enabled for communications with DTF for DIGITAL UNIX.

Meaning: No meaningful communications can take place with this mainframe component.

Corrective Action: If there are multiple copies of DTF running on the mainframe, execute a DTF/DU request (such as dtfls) using *-vv* on the command line. Scan the output for:

- ◆ DUN1122I to verify mainframe application id
- ◆ DUN1111I to verify mainframe DTF/DU version
- ◆ DUN1112I to verify mainframe maint. level

DUN8002E DTF for Digital UNIX rejected SNA BIND request with sense code <text>

Cause: The UNIX component of DTF has rejected an SNA BIND request received from the mainframe. This message documents the 'sense code'

contained in the response that the UNIX component sends to the mainframe.

Meaning: The requested operation is not performed.

Corrective Action: Examine other error messages preceding or following this message. In most cases, one or more other messages will provide more explicit information regarding the real cause of the failure. If the cause of the error is not readily apparent, execute a DTF/DU request (such as dtfls) using -vv on the command line. Examine the output messages to verify that communications is being attempted with an appropriate mainframe component.

If the cause of the error is still not apparent, repeat the corrective action after setting environmental variables SNALOG_MASK=0x7FFFFFFF and SNALOG_FILE to an appropriate UNIX file name.

DUN8003E Error detected during function dfci_<text>

Cause: The indicated SNA communications function completed with an error or exception status code. This message documents the particular dfci layer function call. The function name may be followed with a slash (/) and an additional comment that further defines the purpose of the function call.

Meaning: The requested operation is not performed.

Corrective Action: This message should always be followed by additional messages which document the error condition in more detail.

DUN8004E <text>

Cause: A SNA or SNA gateway related exception condition has been detected. This message documents field IOSB1. The most common codes are supplemented with a short text explanation or description. When such a description is not available, the status codes are displayed in hexadecimal and decimal formats.

Meaning: The requested operation is not performed.

Corrective Action: If the cause of the error condition is not apparent, follow the procedure described under message DUN8002 to generate more detailed information.

DUN8005E <text>

Cause: A SNA or SNA gateway related exception condition has been detected. This message documents field IOSB2. The most common codes are supplemented with a short text explanation or description.

When such a description is not available, the status codes are displayed in hexadecimal and decimal formats.

Meaning: The requested operation is not performed.

Corrective Action: If the cause of the error condition is not apparent, follow the procedure described under message DUN8002 to generate more detailed information.

DUN8006E <text>

Cause: A SNA or SNA gateway related exception condition has been detected. This message documents the SENSE value.

The most common codes are supplemented with a short text explanation or description. When such a description is not available, the status codes are displayed in hexadecimal and decimal formats.

Meaning: The requested operation is not performed.

Corrective Action: If the cause of the error condition is not apparent, follow the procedure described under message DUN8002 to generate more detailed information.

DUN8007E Unexpected SNA Message received : <text>

Cause: A valid but unexpected SNA message has been received. This message is used to indicate that the current operation will be terminated because of the unexpected message. The value following the ':' are the three byte SNA request header and the first byte of the request unit.

Meaning: The requested operation is not performed.

Corrective Action: If the text string (UNBIND) appears in the message, the UNIX component has received a session termination request from the DEC/SNA gateway. Investigate the status of the gateway and/or the SNA network between the gateway and the mainframe. If the text string (UNBIND) does not appear in the message, check on the status of the DEC/SNA gateway, the SNA communications network between the gateway and the DTF mainframe component, and the mainframe component. The most common cause of this messages is a network outage between the DEC/SNA gateway and the IBM mainframe.

If the problem persists, recreate the problem using -vv on the command line of the DTF/DU command, and with a full SNALOG_MASK trace. If convenient, use the DEC/SNA gateway trace facilities as well.

DUN8008E An SNA Exception Request has been received : sense = <text>

Cause: A valid but unexpected SNA message has been received. The received data is an FMD Exception Request. The text field contains the SENSE CODE. This condition generally means that one of the nodes

between the IBM mainframe and the UNIX system has detected an error condition and modified the request into an Exception Request.

Meaning: The requested operation is not performed.

Corrective Action: Examine the sense code value and then contact your IBM mainframe support group for additional information. The most common cause of this messages is a network configuration mismatch between two adjacent nodes in the SNA backbone portion of the network.

If the problem persists, recreate the problem using -vv on the command line of the DTF/DU command, and with a full SNALOG_MASK trace. If convenient, use the DEC/SNA gateway trace facilities as well.

DUN9000E This program can only be executed by root or super user

Cause: The current userid is not root. The IBM dispatcher can only be executed with root or super user authority.

Meaning: The IBM dispatch program will terminate.

Corrective Action: Logon as root or become super user before running this program.

DUN9001E A node name must be specified on the command line

Cause: The command line parameters did not include a node name. The node name on the command line must match the value specified on IBM 3270 DTF SERVER NODE field.

Meaning: The IBM dispatch program will terminate.

Corrective Action: Supply a node name that is unique among all nodes supported by the IBM mainframe component of DTF.

DUN9002E Unable to run as daemon -- fork failed

Cause: The dispatcher module could not convert itself into a daemon using the UNIX fork function. The next message contains the UNIX errno value.

Meaning: The IBM dispatcher program will terminate.

Corrective Action: Investigate the cause of the error based on the errno value.

DUN9003I Dispatcher startup or restart in progress

Cause: The dispatcher module is starting processing.

Meaning: The IBM dispatcher program will execute.

Corrective Action: None required.

DUN9004I <text>

Cause: The dispatcher module started as a result of a DTFDSP command.

Meaning: Use these values to determine what startup parameters are in effect.

Corrective Action: None required.

DUN9005I Dispatcher session with IBM mainframe established

Cause: The dispatcher module has successfully started an SNA session with the IBM mainframe. This message indicates that IBM initiated file transfers should now be possible. This message is displayed only after an SNA communications error disrupted an earlier error message.

Meaning: IBM initiated file transfers should now be possible.

Corrective Action: None required.

DUN9013W Dispatcher session terminated after event : <text>

Cause: An error has been detected while attempting to communicate with the mainframe component of DTF.

Meaning: The IBM dispatcher program will wait 1 minute and attempt to re-connect to the mainframe. Any IBM 3270 users that were waiting for a file transfer to complete will be notified of this outage.

Corrective Action: Wait for at least 1 minute and check to see if the session is re-established. If so, no further action is needed.

Examine preceding messages to determine the cause of the failure. This message will normally be preceded by messages DUN8003 and DUN8004. If the problem cannot be isolated, restart the dispatcher program using -vv on the command line and with SNALOG_MASK set to 0x7FFFFFFF.

NOTE

Ensure that you limit the length of time and/or number of IBM initiated jobs run with this setting. Both the dispatcher program AND all IBM initiated copy operations will perform traces. The size of the copy job traces can become extremely large.

DUN9014W Dispatcher has no in-memory data for terminating job <text>

Cause: An IBM initiated job has completed and notified the dispatcher. The dispatcher is unable to match the completed job number/UNIX userid with an IBM access number/IBM userid waiting for the notification. This message should not appear in normal operations.

The only exception is when the dispatcher program is terminated and restarted while IBM initiated jobs are running.

Meaning: The dispatcher program cannot notify the originating IBM mainframe user that the copy operation has completed.

Corrective Action: If this messages happens as a result of the dispatcher program being restarted while IBM initiated jobs are running, the message can be ignored. The IBM user should have been notified that the server became unavailable when the dispatcher terminated.

Else, stop and then restart the dispatcher program using -vv on the command line. If the error reoccurs, save the dispatcher log file and report the problem.

DUN9015E Validation failure caused by received node name of <text>

Cause: A connect request has been received with a nodename different than the node name where the dispatcher is running. This version of DTF/DU supports the file transfer between the server node and the IBM mainframe only.

Meaning: The operation requested by the IBM mainframe user will not be performed.

Corrective Action: None required. An appropriate error message should have been returned to the IBM user by the mainframe component of DTF/DU.

DUN9016E Validation failure caused by invalid user id of <text>

Cause: A connect request has been received with a UNIX user name that is not available on the UNIX node.

Meaning: The operation requested by the IBM mainframe user will not be performed.

Corrective Action: None required. An appropriate error message should have been returned to the IBM user by the mainframe component of DTF/DU.

DUN9017E Validation failure caused by missing user id

Cause: A connect request has been received with no UNIX user name specified.

Meaning: The operation requested by the IBM mainframe user will not be performed.

Corrective Action: Repeat request with UNIX userid and password supplied. Proxy support is not currently available.

DUN9018E Validation failure caused by received password of <text>

Cause: A connect request has been received with a valid. UNIX userid but with an invalid password.

Meaning: The operation requested by the IBM mainframe user will not be performed.

Corrective Action: None required. An appropriate error message should have been returned to the IBM user by the mainframe component of DTF/DU.

DUN9019E Validation failure caused by no UNIX user password received

Cause: A connect request has been received with a valid. UNIX userid but with a no UNIX password.

Meaning: The operation requested by the IBM mainframe user will not be performed.

Corrective Action: None required. An appropriate error message should have been returned to the IBM user by the mainframe component of DTF/DU.

DUN9020E Could not execute rsh command - check ~<text>/rhost

Cause: The DTF dispatcher needs to execute a rsh but cannot, possibly because the .rhost file cannot be accessed. Some reasons are

- the file does not exist
- file mode is not 600
- no entry in .rhost for `hostname` root

Meaning: The operation requested by the IBM mainframe user will not be performed.

Corrective Action: None required. An appropriate error message should have been returned to the IBM user by the mainframe component of DTF/DU.

DUN9021W Dispatcher discarded command qualifier <text>

Cause: The indicated copy command qualifier has been discarded.

Meaning: Operation requested by the IBM mainframe user will proceed.

Corrective Action: Investigate the discarded qualifier. If it was a valid qualifier, please recreate the problem with -vv on the dispatcher command line and with a full *SNALOG_MASK* trace.

DUN9022I <text>

Cause: The `-vv` option was used on the command line. This message number precedes various text messages generated as a result of using the `-vv` option on the command line.

Meaning: The `-vv` option is performing as designed.

Corrective Action: None required

DUN9023E Dispatcher received command without preceding connect

Cause: A valid copy request has been received but no record of the connect request that should have preceded it can be found.

Meaning: The requested operation is not performed.

Corrective Action: None required. The requesting IBM user should be notified of the condition with an appropriate error message.

DUN9024E Unsupported command type received : <text>

Cause: A command received from the IBM mainframe contained an unsupported value.

Meaning: The requested copy operation will not be performed.

Corrective Action: Restart the dispatcher with `-vv` on command line and with full `SNALOG_MASK` trace enabled. Reproduce the problem. Examine the dispatcher log file and the `SNALOG_MASK` file for related or additional error messages.

DUN9025E <text>

Cause: A shell script invoked by the dispatcher program has not completed successfully. A message indicating the shell script name and other available information is printed.

Meaning: The requested copy operation will not be performed

Corrective Action: If a job was created on the UNIX node, examine the job's log file for additional diagnostic information. [For IBM initiated jobs, the IBM `RETRY` number matches the UNIX `JOB` number].

Restart the dispatcher with `-vv` on the command line and with full `SNALOG_MASK` trace enabled. Reproduce the problem. Examine the dispatcher log file and the `SNALOG_MASK` file for related or additional error messages.

DUN9902E Batch Job could not be created : <text>

Cause: An attempt to create a batch job (`dtfui` or `dtfiu -q`) has failed. Further information may be provided following the `'` in the message.

Meaning: The batch job has either not been created or not created properly and completely.

Note

Do not attempt to release or activate the job. After investigation of the error, the job should be deleted.

Corrective Action: Try a very simple batch job that is not expected to execute. A good choice would be:

```
dtfiu -q -h a.a b.b
```

If this simple batch job works, then the real problem may be with the various parameters associated with the original copy request.

DUN9903E Batch Job status was not set to ready : <text>

Cause: An attempt to activate a batch job (dtfui or dtfiu -q) may have failed. The batch job itself has been defined. Only the activation phase may have failed. If more detailed information is available concerning the cause of the failure, the information will be displayed following the ':' in the message.

Meaning: The batch job has been created properly, but may not have been activated.

Corrective Action: Examine the batch job's log file (dtfjob log JOB_#). There should be messages in the log file that would provide more information on the failure.

DUN9996I FOR TESTING ONLY -- INTENTIONAL BREAKPOINT REACHED

Cause: A network outage has been simulated at the point in the transfer indicated by the -B and -N value on the command line.

Meaning: The copy operation has terminated with a completion code of 1 (retry recommended).

Corrective Action: None required. The -B and -N command line parameter are not documented or intended for customer use. To prevent this error condition, remove both -B and -N from the command line.

DUN9998E UNIX errno <text>

Cause: The UNIX function documented in a previous error message completed with an error or exception indicator. This message documents the content of 'errno' at the time of the failure.

The text description of the errno value is provided when available.

Meaning: The meaning depends on the function that failed and on the errno value.

Corrective Action: Review the description of the failing UNIX function along with the errno value provided.

The most common failures include authorization failures and out of space conditions.

DUN9999E Error Message text not available for message <text>

Cause: Text for the indicated error message is not available in the error messages file.

Meaning: The message text is not available. There is no other adverse effect

Corrective Action: Refer to printed documentation to determine the intended message text, cause, meaning, and corrective action.

13. Sample Installation Dialog

This section is a sample installation dialog for the DTF/DU for Digital UNIX, version 1.0 software.

```
# setld -l /mnt/df100 [Return]
```

DTFDU for Digital UNIX, V1.0

Copyright Digital Equipment Corporation. 1994, 1995. All Rights Reserved. This software is proprietary to and embodies the confidential technology of Digital Equipment Corporation. Possession, use, duplication or dissemination of this software and media is authorized only pursuant to a valid written license from Digital Equipment Corporation or an authorized sublicensor.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in Subparagraph (c)(1)(ii) of DFARS 252.227-7013, or in FAR 52.227-19, or in FAR 52.227-14 Alt. III, as applicable.

The subsets listed below are optional:

There may be more optional subsets than can be presented on a single screen. If this is the case, you can choose subsets screen by screen or all at once on the last screen. All of the choices you make will be collected for your confirmation before any subsets are installed.

- 1) DTF for DIGITAL UNIX - MAN Pages (optional)
- 2) DTF for DIGITAL UNIX - Run Time Components

Or you may choose one of the following options:

- 3) ALL of the above
- 4) CANCEL selections and redisplay menus
- 5) EXIT without installing any subsets

Enter your choices or press RETURN to redisplay menus.

Choices (for example, 1 2 4-6): 3 **[Return]**

You are installing the following optional subsets:

DTF for DIGITAL UNIX - MAN Pages (optional)
DTF for DIGITAL UNIX - Run Time Components

Is this correct? (y/n): y **[Return]**

Checking file system space required to install selected subsets:

File system space checked OK.

DTF for DIGITAL UNIX - Run Time Components

Copying from . (disk)

Verifying

```

DTF for DIGITAL UNIX - MAN Pages (optional)
Copying from . (disk)
Verifying
Configuring "DTF for DIGITAL UNIX - Run Time Components" (DTFRUN100)
Would you like to edit the alias file now? (y/n) [y]: y
* Alias GW Transp Access Appl Logmode PU Xlate Null
* Entry Node Name Name
*****
dtf32 getq tcp tso mvsedev FT819208 * remote ignore
dtf32_d getq decnet tso mvsedev FT819208 * remote ignore
dtf32_l getq local tso mvsedev FT819208 * remote ignore
**
** Please edit this default alias file. It describes the DEC/SNA gateway(s)
** and IBM mainframes.
**
** 1. Any line that starts with a '*' is a comment
** 2. Any line that does not start with a * MUST contain 9 or more values
** 3. The first line that is not a comment is used as the default entry.
**
** Alias: Name used in dtf command to select a line in this file.
** Node: DEC/SNA Gateway node name
** Trans: Transport between this node and DEC/SNA gateway (tcp,decnet,local)
** ACCES: DEC/SNA defined access name used when starting a session
** APPL: IBM mainframe APPLICATION ID (aka ACB NAME)
** LOGMODE: Logmode entry name (defined in mainframe, and in Domain
gateways)
** PU: PU number (used with CT and ST gateways only)
** XLATE: Translation option (local,remote,none)
** NULL NULL record option (none,ignore,number)
"/etc/ibm_dtf_alias" 23 lines, 1258 characters

Would you like to run the IVP now? (y/n) [y]: y
DTFRUN100 IVP PHASE 1 of 5 in progress

```

DTFRUN100 IVP PHASE 2 of 5 in progress
DTFRUN100 IVP PHASE 3 of 5 in progress
DTFRUN100 IVP PHASE 4 of 5 in progress
DTFRUN100 IVP PHASE 5 of 5 in progress
DTFRUN100 IVP COMPLETE - NO ERRORS DETECTED
Configuring "DTF for DIGITAL UNIX - MAN Pages (optional)" (DTFMAN100)
#

14. Files Installed on Your System

This chapter describes the files, soft links, and directories created when you install the DTF/DU software.

14.1 Files Created

The installation procedure creates the following files on your system for the DTFMAN100 subset:

```
/usr/opt/DTF100/man/dtf  
/usr/opt/DTF100/man/dtfdsp  
/usr/opt/DTF100/man/dtfui  
/usr/opt/DTF100/man/dtfjob  
/usr/opt/DTF100/man/dtfls  
/usr/opt/DTF100/man/dtfpr  
/usr/opt/DTF100/man/dtfrm  
/usr/opt/DTF100/man/dtfsb  
/usr/opt/DTF100/man/dtfui
```

The install procedure creates the following soft links for the DTFMAN100 subset:

```
/usr/share/man/man1/df  
/usr/share/man/man1/dtfdsp  
/usr/share/man/man1/dtfui  
/usr/share/man/man1/dtfjob  
/usr/share/man/man1/dtfls  
/usr/share/man/man1/dtfpr  
/usr/share/man/man1/dtfrm  
/usr/share/man/man1/dtfsb  
/usr/share/man/man1/dtfui
```

The installation procedure creates the following files on your system for the DTFRUN100 subset:

- /etc/ibm_dtf_alias
- /etc/ibm_dtf_defaults/defaults
- /etc/ibm_dtf_defaults/prolog_ibm
- /etc/ibm_dtf_defaults/prolog_unix
- /etc/ibm_dtf_defaults/system_defaults
- /etc/ibm_dtf_defaults/system_prolog_ibm
- /etc/ibm_dtf_defaults/system_prolog_unix
- /etc/ibm_dtf_examples/dsp_logs
- /etc/ibm_dtf_examples/example
- /etc/ibm_dtf_examples/inc_example
- /etc/ibm_dtf_examples/iu_ex
- /etc/ibm_dtf_examples/iu_ex_prep
- /etc/ibm_dtf_examples/ls_ex
- /etc/ibm_dtf_examples/pr_ex
- /etc/ibm_dtf_examples/rm_ex
- /etc/ibm_dtf_examples/sb_ex
- /etc/ibm_dtf_examples/sb_ex_prep
- /etc/ibm_dtf_examples/ui_ex
- /etc/ibm_dtf_examples/ui_ex_prep
- /etc/ibm_dtf_translate/sample
- /usr/opt/DTF100/bin/dtf_batch
- /usr/opt/DTF100/bin/dtf_iiia
- /usr/opt/DTF100/bin/dtf_iiid
- /usr/opt/DTF100/bin/dtf_iiij
- /usr/opt/DTF100/bin/dtf_iiin
- /usr/opt/DTF100/bin/dtf_preferences
- /usr/opt/DTF100/bin/dtfboot
- /usr/opt/DTF100/bin/dtfcron
- /usr/opt/DTF100/bin/dtfdsp
- /usr/opt/DTF100/bin/dtfiid

/usr/opt/DTF100/bin/dtfiis
/usr/opt/DTF100/bin/dtfivp
/usr/opt/DTF100/bin/dtfjob
/usr/opt/DTF100/bin/dtfmlu
/usr/opt/DTF100/bin/dtfs
/usr/opt/DTF100/bin/dtfsubs
/usr/opt/DTF100/msg/dap
/usr/opt/DTF100/msg/dun
/usr/opt/DTF100/msg/stv
/usr/opt/DTF100/shlib/libsnadtf0
/usr/opt/DTF100/shlib/libsnadtf1
/var/opt/DTF100/etc/ibm_dtf_alias
/var/opt/DTF100/etc/ibm_dtf_defaults/defaults
/var/opt/DTF100/etc/ibm_dtf_defaults/prolog_ibm
/var/opt/DTF100/etc/ibm_dtf_defaults/prolog_unix
/var/opt/DTF100/etc/ibm_dtf_defaults/system_defaults
/var/opt/DTF100/etc/ibm_dtf_defaults/system_prolog_ibm
/var/opt/DTF100/etc/ibm_dtf_defaults/system_prolog_unix
/var/opt/DTF100/etc/ibm_dtf_examples/dsp_logs
/var/opt/DTF100/etc/ibm_dtf_examples/example
/var/opt/DTF100/etc/ibm_dtf_examples/inc_example
/var/opt/DTF100/etc/ibm_dtf_examples/iu_ex
/var/opt/DTF100/etc/ibm_dtf_examples/iu_ex_prep
/var/opt/DTF100/etc/ibm_dtf_examples/ls_ex
/var/opt/DTF100/etc/ibm_dtf_examples/pr_ex
/var/opt/DTF100/etc/ibm_dtf_examples/rm_ex
/var/opt/DTF100/etc/ibm_dtf_examples/sb_ex
/var/opt/DTF100/etc/ibm_dtf_examples/sb_ex_prep
/var/opt/DTF100/etc/ibm_dtf_examples/ui_ex
/var/opt/DTF100/etc/ibm_dtf_examples/ui_ex_prep
/var/opt/DTF100/etc/ibm_dtf_translate/sample
/var/opt/DTF100/usr_sums

14.2 Softlinks Created

The install procedure creates the following soft links for the DTFRUN100 subset:

```
/sbin/init.d/dtfboot  
/sbin/rc3.d/S57b4_dtfcron  
/usr/bin/dtfboot  
/usr/bin/dtf_batch  
/usr/bin/dtf_ia  
/usr/bin/dtf_iid  
/usr/bin/dtf_iij  
/usr/bin/dtf_iin  
/usr/bin/dtf_preferences  
/usr/bin/dtfcron  
/usr/bin/dtfdsp  
/usr/bin/dtfid  
/usr/bin/dtfiis  
/usr/bin/dtfij  
/usr/bin/dtfiu  
/usr/bin/dtfjob  
/usr/bin/dtfls  
/usr/bin/dtflmlu  
/usr/bin/dtfpr  
/usr/bin/dtfrm  
/usr/bin/dtfsb  
/usr/bin/dtfsubs  
/usr/bin/dtfui  
/usr/bin/dtfuj  
/usr/shlib/libsnadt0.so  
/usr/shlib/libsnadt1.so
```